As you look at Figure 2-12,understand that any particular data model represents a specific situation ,not a generalization. For example,consider the Manages relationship in Figure 2-12a. In some organizations,it may be possible for one employee to be managed by many other employees (e.g. in a matrix organization). It is important when you develop an E-R model that you understand the business rules of the particular organization you are modeling.

**Unary relationship** a unary relationship is a relationship between the instances of a single entity type.(Unary relationships are also called recursive relationships.) Three examples are shown in Figure 2-12a.In the first example,Is Married To is shown as a one-to-one relationship between instances of the PERSON entity type.Because this is a one-to-one relationship,this notation indicates that only the current marriage ,if one exists,needs to be kept about a person. What would change if we needed to retain the history of marriages for each person? See Problem and Exercise 12 for other business rules and their effect on the Is Married To relationship representation. In the second example,Manages is shown as a one-to-many relationship between instances of the EMPLOYEE entity type. Using this relationship,we could identify,for example,the employees who report to a particular manager.The third example is one case of using a unary relationship to represent a sequence,cycle,or priority list.In this example,sports teams are related by their standing in their league(the Stands After relationship ).(Note:In this examples,we ignore whether these are mandatory-or optional-cardinality relationships or whether the same entity instance can repeat in the same relationship instance;we will introduce mandatory and optional cardinality in a later section of this chapter.)

Figure 2-13 shows an example of another unary relationship,called a bill-of-materials structure.Many manufactured products are made of assemblies,which in turn are composed of subassemblies and parts, and so on. As shown in Figure 2-13a,we can represent this structure as a many-to-many unary relationship. In this figure,the entity type ITEM is used to represent all types of components,and we use Has Components for the name of the relationship type that associates lower-level items with higher-level items.

Two occurrences of this bill-of-materials structure are shown in Figure2-13b.Each of these diagrams shows the immediate components of each item as well as the quantities of that component. For example,item TX100 consists of item BR450(quantity 2) and item DX500(quantity 1). You can easily verify that the associations are in fact many-to-many. Several of the items have more than one component type(e.g. item MX300 has three immediate component types :HX100,TX100,and WX2400.Also ,some of the components are used in several higher-level assemblies. For example,item WX240 is used in both item MX300 and item WX340,even at different levels of the bill-of-materials. The many-to-many relationship guarantees that ,for example,the same subassembly structure of WX240(not shown) is used each time item WX240 goes into making some other item.

The presence of the attribute Quantity on the relationship suggests that the analyst consider converting the relationship Has Components to an associative entity. Figure 2-13c shows the entity type BOM STRUCTURE,which forms an associative between instances of the ITEM entity type.A second attribute (named Effective Date)has been added to BOM STRUCTURE to record the date when this component was first used in the related assembly.Effective dates are often needed when a history of values is required. Other data model structures can be used for unary relationships involving such hierarchies;we show some of these other structures in Chapter9.

TERNARY RELATIONSHIP A ternary relationship is a simultaneous relationship among the instances of three entity types. A typical business situation that leads to a ternary relationship is shown in Figure 2-12c. In this example,vendors can supply various parts to warehouses.The relationship Supplies is used to record the specific parts that are supplied by a given vendor to a particular warehouse.Thus,there are three entity types:VENDOR,PART,and WAREHOUSE.There are two attributes on the relationship Supplies:Shipping Mode and Unit Cost.For example,one instance of Supplies might record the fact that vendor X can ship part C to warehouse Y,that the shipping mode is next-day air,and that the cost is ＄5 per unit.

Don’t be confused:A ternary relationship is not the same as three binary relationships.For example,Unit Cost is an attribute of the Supplies relationship in Figure 2-12c.Unit Cost cannot be properly associated with any one of the three possible binary relationships among the three entity types,such as that between PART and WAREHOUSE.

Thus ,for example,if we were told that vendor X can ship part C for a unit cost of ＄8 ,those data would be incomplete because they would not indicate o which warehouse the parts would be shipped .

As usual, the presence of an attribute on the relationship Supplies in Figure 2-12c suggests converting the relationship to an associative entity type .Figure 2-14 shows an alternative (and preferable)representation of the ternary relationship shown in figure 2-12c.In Figure 2-14 ,the(associative)entity type SUPPLY SCHEDULE is used to replace the Supplies relationship from Figure 2-12.Clearly,the entity type SUPPLY SCHEDULE is of independent interest to users.However,notice that an identifier has not yet been assigned to SUPPLY SCHEDULE.This is acceptable.If no identifier is assigned to an associative entity during E-R modeling,an identifier(or key)will be assigned during logical modeling (discussed in Capter4).This will be a composite identifier whose components will consist of the identifier for each of the participating entity types(in the example,PART,VENDOR,and WAREHOUSE).Can you think of other attributes that might be associated with SUPPLY SCHEDULE.

As noted earlier,we do not label the lines from SUPPLY SCHEDULE to the three entities.This is because these lines do not represent binary relationships.To keep the same meaning as the ternary relationship of Figure 2-12c,we cannot break the Supplies relationship into three binary relationships,as we already mentioned.

So,here is a guideline to follow: Convert all ternary(or higher) relationships to associative entities,as in this example.Song et al.(1995)show that participation constraints(described in a following section on cardinality constraints)cannot be accurately represented for a ternary relationship,given the notation with attributes on the relationship line.However,by converting to an associative entity,the constraints can be accurately represented .Also,many E-R diagram drawing tools,including most CASE tools,cannot represent ternary(or higher order)relationship.So,although not semantically accurate,you must use these tools to represent the ternary relationship with an associative entity an three binary relationships,which have a mandatory association with each of the three related entity types.

**Attribute s or Entity?**

Sometimes you will wonder if you should represent data as an attribute or an entity;this is a common dilemma.Figure 2-15 includes three examples of situations when an attribute could be represented via an entity type. We use this texts E-R notation in the left column and the notation from Microsoft Visio in the right column;it is important various that you learn how to read ERDs in several notations because you will encounter various styles in different publications an organizations.In Figure 2-15a,the potentially multiple prerequisites of a course(shown as a multivalude attribute in the Attribute cell)are also courses(and a course may be a prerequisite for many other courses).Thus,prerequisite could be viewed as a bill –of materials structure