DEFINING SUBTYPE DISCRIMINATORS

Given a supertype/subtype relationship, consider the problem of inserting a new instance of a supertype. Into which of the subtypes 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. 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.

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. Notice that the following constraints have been added to this 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.

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.

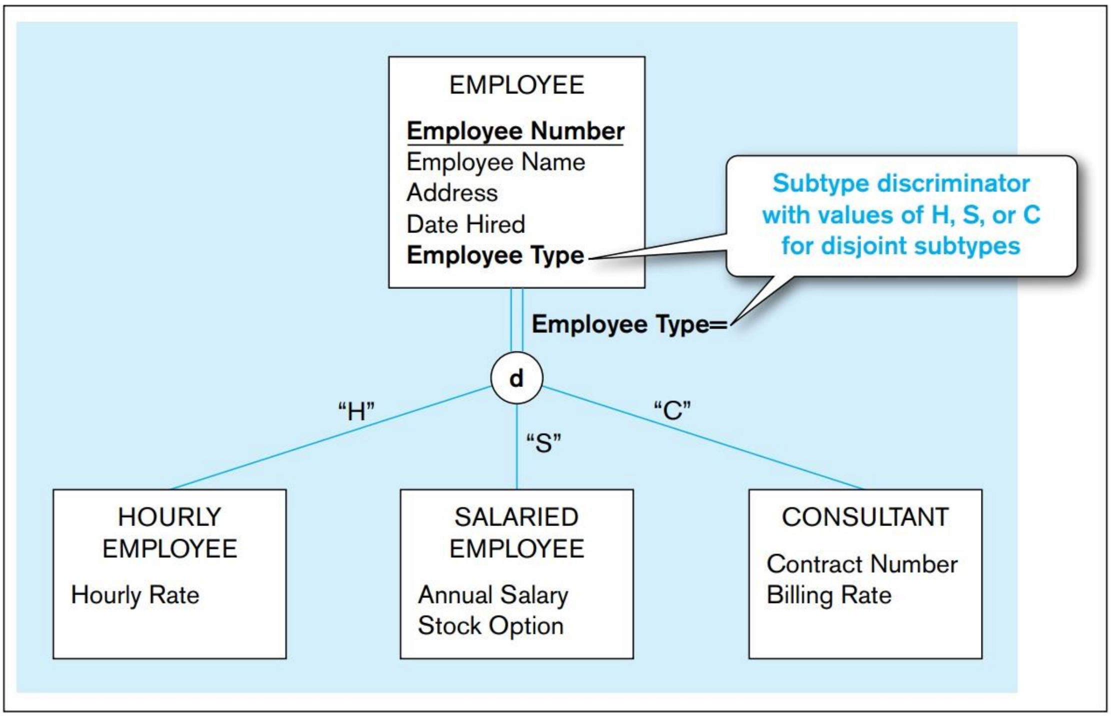


FIGURE 3-8 Introducing a subtype discriminator (disjoint rule)

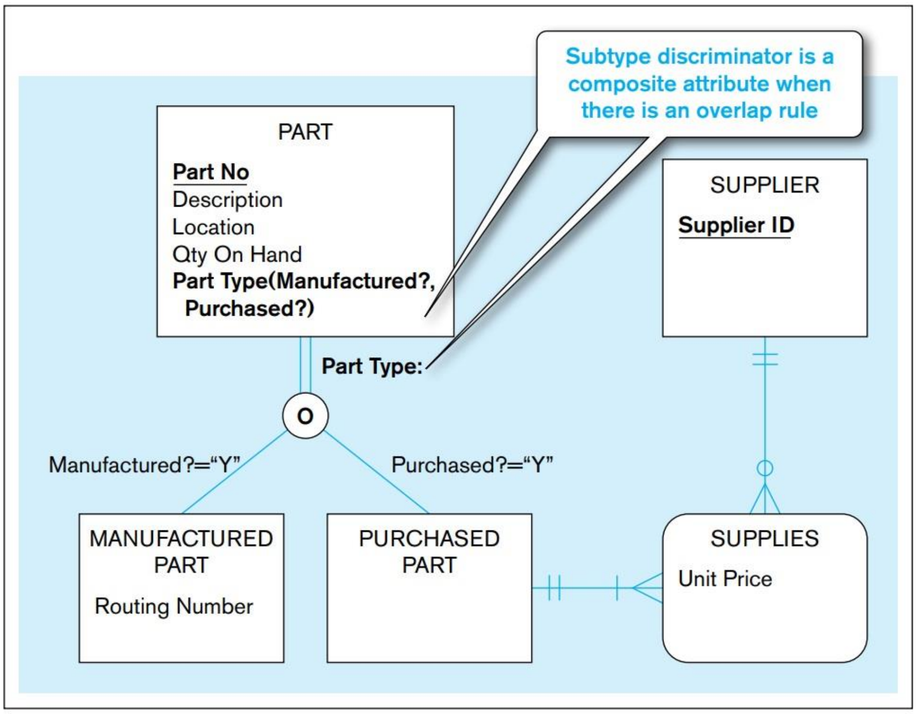
Overlapping Subtypes

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 3-9 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.

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 examples). This example includes most of the concepts and notation we have used in this chapter to this point.

An Example of a Supertype/Subtype Hierarchy

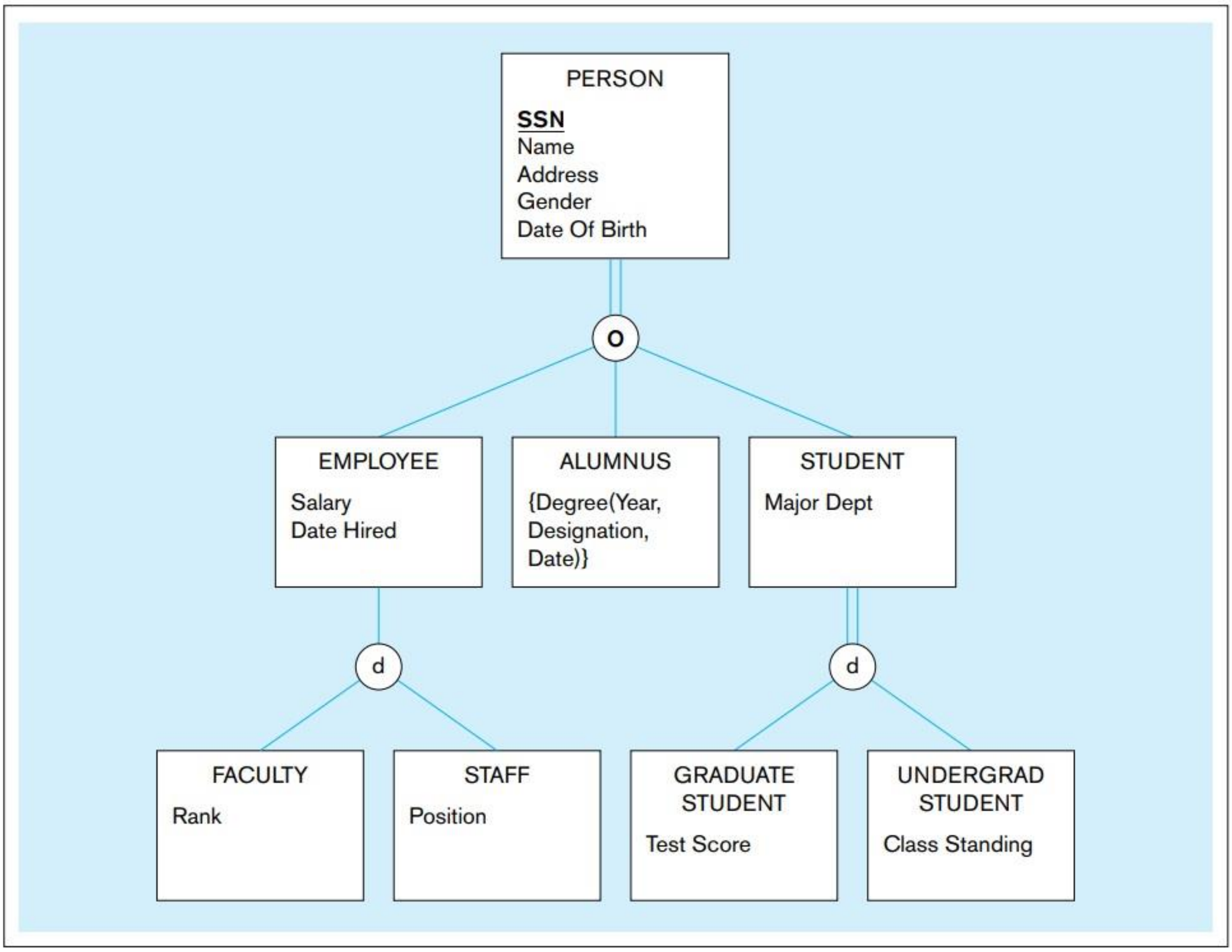
Suppose that you are asked to model the human resources in a university. You might proceed as follows: Starting at the top of a hierarchy, model the most general entity type first. In this case, the most general entity type is PERSON. List and associate all attributes of PERSON. The attributes shown 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).

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 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.

FIGURE 3-10 Example of supertype/subtype hierarchy



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.

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).