



**Figure 5-21**  
Sample Relationships Showing  
Both Maximum and Minimum  
Cardinalities

one is called, not surprisingly, a **crow's-foot diagram** version. You may learn other versions if you take a database management class.

The crow's-foot notation shows the maximum number of entities that can be involved in a relationship. Accordingly, they are called the relationship's **maximum cardinality**. Common examples of maximum cardinality are 1:N, N:M, and 1:1 (not shown).

Another important question is, "What is the minimum number of entities required in the relationship?" Must an adviser have a student to advise, and must a student have an adviser? Constraints on minimum requirements are called **minimum cardinalities**.

Figure 5-21 presents a third version of this E-R diagram that shows both maximum and minimum cardinalities. The vertical bar on a line means that at least one entity of that type is required. The small oval means that the entity is optional; the relationship *need not* have an entity of that type.

Thus, in Figure 5-21 a department is not required to have a relationship to any adviser, but an adviser is required to belong to a department. Similarly, an adviser is not required to have a relationship to a student, but a student is required to have a relationship to an adviser. Note, also, that the maximum cardinalities in Figure 5-21 have been changed so that both are 1:N.

Is the model in Figure 5-21 a good one? It depends on the policy of the university. Again, only the users know for sure.

## Q6 How Is a Data Model Transformed into a Database Design?

Database design is the process of converting a data model into tables, relationships, and data constraints. The database design team transforms entities into tables and expresses relationships by defining foreign keys. Database design is a complicated subject; as with data modeling, it occupies weeks in a database management class. In this section, however, we will introduce two important database design concepts: normalization and the representation of two kinds of relationships. The first concept is a foundation of database design, and the second will help you understand important design considerations.

### Normalization

**Normalization** is the process of converting a poorly structured table into two or more well-structured tables. A table is such a simple construct that you may wonder how one could possibly be poorly structured. In truth, there are many ways that tables can be malformed—so many, in fact, that researchers have published hundreds of papers on this topic alone.

Consider the *Employee* table in Figure 5-22 (a) on the next page. It lists employee names, hire dates, email addresses, and the name and number of the department in which the employee works. This table seems innocent enough. But consider what happens when the Accounting department changes its name to Accounting and Finance.