once an attribute is replaced by a relationship, the attribute itself should be removed from the entity type to avoid duplication and redundancy.

- Similarly, an attribute that exists in several entity types may be elevated or promoted to an independent entity type. For example, suppose that each of several entity types in a UNIVERSITY database, such as STUDENT, INSTRUCTOR, and COURSE, has an attribute Department in the initial design; the designer may then choose to create an entity type DEPARTMENT with a single attribute Dept_name and relate it to the three entity types (STUDENT, INSTRUCTOR, and COURSE) via appropriate relationships. Other attributes/relationships of DEPARTMENT may be discovered later.

- An inverse refinement to the previous case may be applied—for example, if an entity type DEPARTMENT exists in the initial design with a single attribute Dept_name and is related to only one other entity type, STUDENT. In this case, DEPARTMENT may be reduced or demoted to an attribute of STUDENT.

- Section 3.9 discusses choices concerning the degree of a relationship. In Chapter 4, we discuss other refinements concerning specialization/generalization.

3.7.4 Alternative Notations for ER Diagrams

There are many alternative diagrammatic notations for displaying ER diagrams. Appendix A gives some of the more popular notations. In Section 3.8, we introduce the Unified Modeling Language (UML) notation for class diagrams, which has been proposed as a standard for conceptual object modeling.

In this section, we describe one alternative ER notation for specifying structural constraints on relationships, which replaces the cardinality ratio (1:1, 1:N, M:N) and single/double-line notation for participation constraints. This notation involves associating a pair of integer numbers (min, max) with each *participation* of an entity type E in a relationship type R, where $0 \leq min \leq max$ and $max \geq 1$. The numbers mean that for each entity e in E, e must participate in at least *min* and at most *max* relationship instances in R *at any point in time*. In this method, $min = 0$ implies partial participation, whereas $min > 0$ implies total participation.

Figure 3.15 displays the COMPANY database schema using the (min, max) notation.[14] Usually, one uses either the cardinality ratio/single-line/double-line notation *or* the (min, max) notation. The (min, max) notation is more precise, and we can use it to specify some structural constraints for relationship types of *higher degree*. However, it is not sufficient for specifying some key constraints on higher-degree relationships, as discussed in Section 3.9.

Figure 3.15 also displays all the role names for the COMPANY database schema.

---

[14]In some notations, particularly those used in object modeling methodologies such as UML, the (min, max) is placed on the *opposite sides* to the ones we have shown. For example, for the WORKS_FOR relationship in Figure 3.15, the (1,1) would be on the DEPARTMENT side, and the (4,N) would be on the EMPLOYEE side. Here we used the original notation from Abrial (1974).
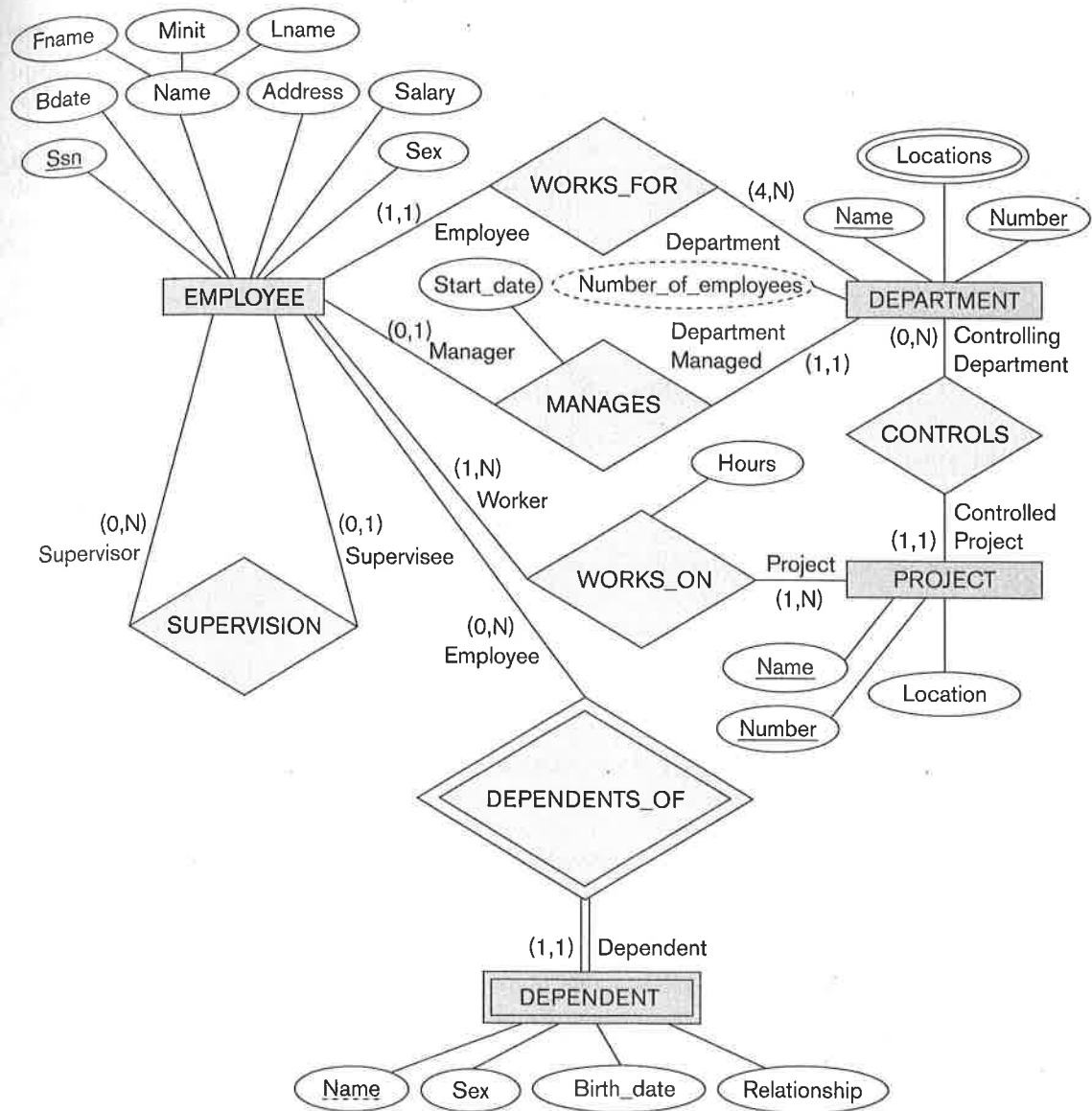
Figure 3.15
ER diagrams for the company schema, with structural constraints specified using
(min, max) notation and role names.

3.8 Example of Other Notation: UML Class Diagrams

The UML methodology is being used extensively in software design and has many
types of diagrams for various software design purposes. We only briefly present the
basics of **UML class diagrams** here and compare them with ER diagrams. In some