1999). UML is the successor to a number of object-oriented analysis and design methods introduced in the 1980s and 1990s. The Object Management Group (OMG) is responsible for the creation and management of UML (available at www.uml.org). UML is currently recognized as the de facto industry standard modeling language for object-oriented software engineering projects. Although we use the UML notation for drawing ER models, we continue to describe the concepts of ER models using traditional database terminology. In Section 27.8 we will provide a fuller discussion on UML. We also include a summary of two alternative diagrammatic notations for ER models in Appendix C.

In the next chapter we discuss the inherent problems associated with representing complex database applications using the basic concepts of the ER model. To overcome these problems, additional "semantic" concepts were added to the original ER model, resulting in the development of the Enhanced Entity–Relationship (EER) model. In Chapter 13 we describe the main concepts associated with the EER model, called specialization/generalization, aggregation, and composition. We also demonstrate how to convert the ER model shown in Figure 12.1 into the EER model shown in Figure 13.8.

Structure of this Chapter In Sections 12.1, 12.2, and 12.3 we introduce the basic concepts of the Entity–Relationship model: entities, relationships, and attributes. In each section we illustrate how the basic ER concepts are represented pictorially in an ER diagram using UML. In Section 12.4 we differentiate between weak and strong entities and in Section 12.5 we discuss how attributes normally associated with entities can be assigned to relationships. In Section 12.6 we describe the structural constraints associated with relationships. Finally, in Section 12.7 we identify potential problems associated with the design of an ER model called connection traps and demonstrate how these problems can be resolved.

The ER diagram shown in Figure 12.1 is an example of one of the possible end-products of ER modeling. This model represents the relationships between data described in the requirements specification for the Branch view of the *DreamHome* case study given in Appendix A. This figure is presented at the start of this chapter to show the reader an example of the type of model that we can build using ER modeling. At this stage, the reader should not be concerned about fully understanding this diagram, as the concepts and notation used in this figure are discussed in detail throughout this chapter.

12.1 Entity Types

Entity type A group of objects with the same properties, which are identified by the enterprise as having an independent existence.

The basic concept of the ER model is the **entity type**, which represents a group of "objects" in the "real world" with the same properties. An entity type has an

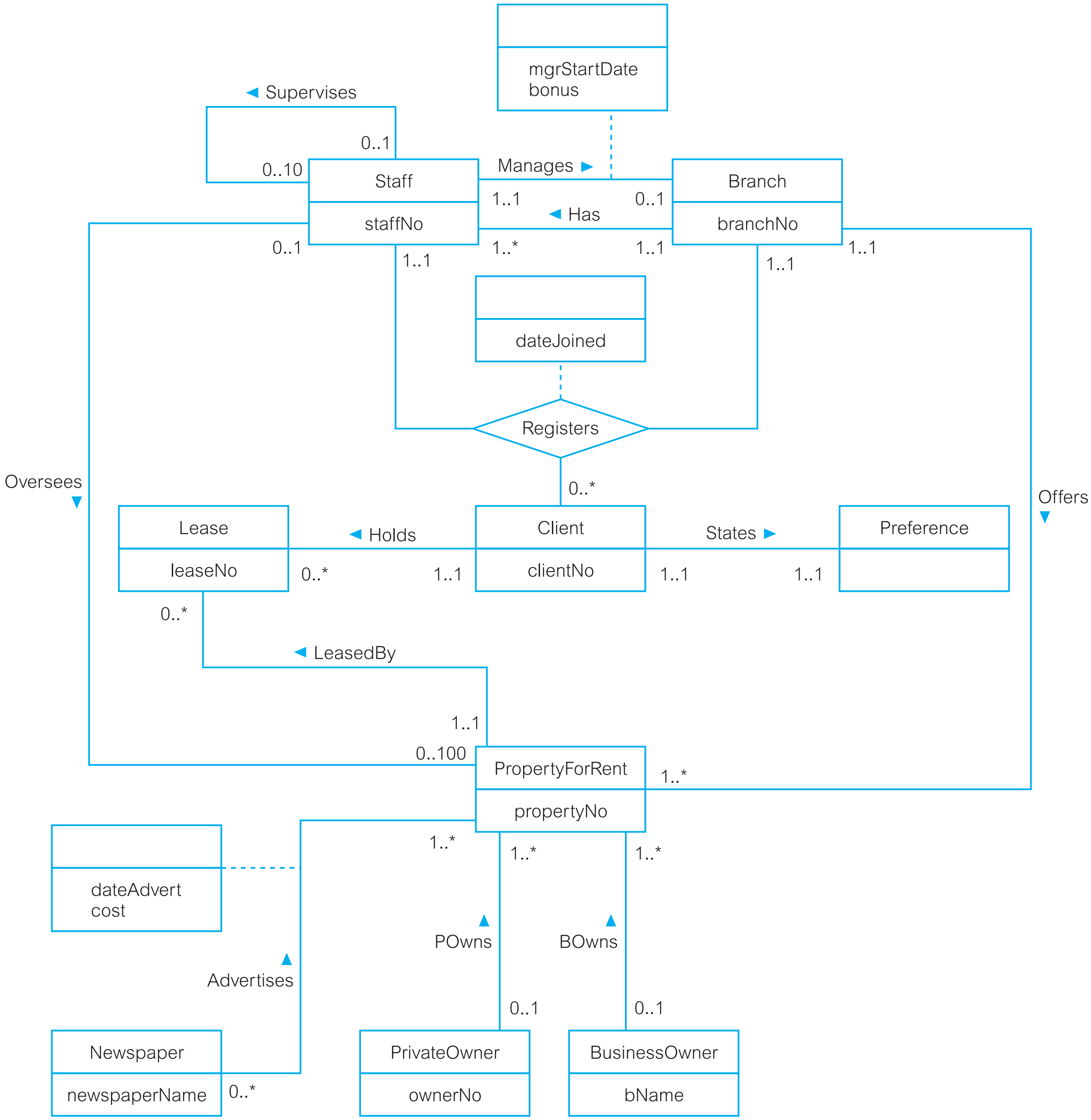

Figure 12.1 An Entity–Relationship (ER) diagram of the Branch view of *DreamHome*.

independent existence and can be objects with a physical (or "real") existence or objects with a conceptual (or "abstract") existence, as listed in Figure 12.2. Note that we are able to give only a working definition of an entity type, as no strict formal definition exists. This means that different designers may identify different entities.

Entity occurrence A uniquely identifiable object of an entity type.