A good database is the bedrock on which you can create a good application. To create a sound database, you need to learn conceptual modeling skills. Database modeling and design are areas in which not all database administrators (DBAs) are uniformly proficient. Other than dealing with tables and the queries that are based on them, many DBAs don’t have a detailed understanding of areas such as normalization, functional dependency, and entity-relationship modeling.

The ability to design a database is particularly useful to DBAs working in smaller organizations where they’ll need to know how to do everything from work with the UNIX file system to resolve networking issues. Even if designing databases isn’t a part of your job description, understanding database design will help you when performance tuning the database.

Because the needs of organizations differ, you can’t take a “one size fits all” approach. This makes database design one of the most interesting and challenging areas available to you when working with databases, particularly large corporate systems. Someone in the organization needs to first model the needs of the organization on a conceptual level and then use this conceptual design to physically design and build the database. Even though it’s not absolutely necessary that you, as a DBA, be an expert in database design, your knowledge as a competent Oracle DBA isn’t complete until you learn at least the rudiments of database modeling and design.

In this chapter, you’ll first learn the conceptual basis of a relational database, which is what an Oracle9i database is. Oracle Corporation refers to its databases as “object-relational,” and you’ll look at this nomenclature issue in more detail toward the end of this chapter. After you explore the basic elements of the relational database life cycle, you’ll learn how to perform conceptual or logical data modeling. The topic of data normalization is very important when dealing with relational databases, and this chapter discusses normalization in detail. Finally, you’ll learn how to translate the logical data model into a design you can physically implement. The chapter concludes with a brief discussion of object-relational databases.
Relational Databases: A Brief Introduction

Oracle9i is a leading example of a relational database management system (RDBMS), although Oracle prefers to call its database an object-relational database management system (ORDBMS). As you'll see toward the end of this chapter, you derive the object-relational model by combining object-oriented design with the traditional relational model. Relational databases have become the pervasive model of organizing data in the last three decades, and they have revolutionized the management of data by companies. The powerful and easy-to-understand relational databases are indeed the mainstay of a vast majority of organizations in today's world economy.

NOTE The very ease with which you can create a database sometimes contributes to poorly designed databases. Thanks to the many RDBMS wizards that walk users step by step through the database creation process, even novices can set up a database. My own general rule of thumb is that if database design isn't your forte, find a person who is good at database design to help you. Good design up front will pay rich dividends later on.

The relational model's domination of the database market is expected to continue into the foreseeable future given the massive investment many large organizations have made in both the databases themselves and the staff required to manage them.

Relational databases are based on the precepts laid down by E. F. Codd in the 1970s, when he was working for IBM Corporation. D. L. Childs presented a similar set-oriented relational model in 1968, but it is Codd's exposition that made relational databases popular. There have been (and there still are) database models that preceded the relational model that are nonrelational in nature—specifically, the hierarchical and the network models. Typically, both the network model and the hierarchical model use actual data links called pointers to process queries issued by users. Both of these models, although powerful as far as performance goes, lead to a very complex database, which has prevented their adoption by most organizations. You can call relational databases second-generation database management systems, with the traditional network and hierarchical flat-file databases being the first-generation database management systems.

The Relational Database Model

A basic feature of relational databases is that all the data is stored in the form of two-dimensional tables. Tables store the data and the metadata (the data about the data). Furthermore, the data has interrelationships among its many components.

NOTE The capability of a relational database to describe itself, through the use of metadata, is one of its most interesting features. Even desktop databases such as Microsoft Access maintain system tables that describe the data and the relationships among the data.