CHAPTER 5

Introduction to the Oracle9i Architecture

In this chapter, you’ll learn about the fundamental structures of the Oracle9i database. I explain in detail the various logical and physical components of the Oracle database and introduce the basic building blocks of the Oracle database: extents, segments, and tablespaces. I also explain the conceptual underpinnings of the Oracle database management system. You’ll learn the difference between an Oracle database and an Oracle instance, and you’ll understand the way Oracle uses memory. You’ll also discover how the Oracle server performs transaction processing, and you’ll cover the concepts behind many of the key Oracle database features.

To understand how the Oracle database works, you need to understand several concepts, including transaction processing, undo management, optimization of SQL queries, and the importance of the data dictionary. This chapter introduces several of these important topics. Later chapters build on the general ideas introduced in this chapter.

Oracle Database Structures

Before you delve deep into the logical and physical structures that make up an Oracle database, I would like to clarify the difference between an Oracle instance and an Oracle database. It is very common for people to use the terms interchangeably, but they refer to different things altogether.

An Oracle database consists of not only your data files, but also many other files. You can see right away that your data will be of no use to you unless you can access it, change it, and add to it. You interact with these files with the help of the operating system. The operating system provides you with processing capabilities and resources, such as memory, to enable you to manipulate the data on the disk drives. When you combine the specific set of processes created by Oracle on the server with the memory allocated to it by the operating system, you get the Oracle instance.

Many times, you’ll hear people remarking that the database is “up,” though what they really mean is that the instance is “up.” The database itself, in the form of the set of physical files it’s composed of, is of no use if the instance is not up and running. The instance performs all the necessary work for the database.

Three major components on a server machine are relevant to the performance of an Oracle database: the central processing unit (CPU), the disk storage system, and random access memory (RAM). (The network that connects users and the server is also important, but that’s not a part of the server, per se.)
Although an Oracle database's performance depends on all three operating system components, most of the time you'll be tuning the database performance by tuning the memory and disk storage components. There isn't a whole lot to tune in terms of the CPU. You make your initial decisions based on your organization's needs and the available finances, and purchase the server with the requisite number of processors (24 Intel 1.05 GHz processors, for example). Once you buy the CPU, you're pretty much done as far as tweaking and adjusting it, unless you realize there's a CPU bottleneck and you upgrade the number and/or the speed of your processors. The other two components, disk space and memory, are a different ballgame altogether. You will often face the need to tune and adjust several configuration parameters related to these two components.

You can approach the structures of the Oracle databases based on the distinction between physical and logical structures. You don't take all the data belonging to the tables of an Oracle database and just put it on disk somewhere on the operating system storage system. Oracle uses a sophisticated logical view of the internal database structures that help in storing and managing data properly on the physical data files. This logical defining of Oracle's database structure has another fundamental motive behind it. By organizing space into logical structures and assigning these logical entities to users of the database, Oracle databases achieve the logical separation of users (owners of the database objects, such as tables) of the database from the physical manifestations of the database in terms of data files and so forth. The next section covers how Oracle logically partitions its databases and maps these logical parts to the physical components of the database.

**The Logical Structures**

Oracle9i uses a set of logical database storage structures to use disk space, whether the database uses operating system files or "raw" database files. These logical structures, which primarily include tablespaces, segments, extents, and blocks, control the allocation and use of the "physical" space allocated to the Oracle database. Note that Oracle database objects such as tables, indexes, views, sequences, and others are also logical entities. These logical objects make up the relational design of the database, which I discuss in detail later on.

You can look at the logical composition of Oracle databases from either a top-down viewpoint or a bottom-up viewpoint. Let's use the bottom-up approach by first looking at the smallest logical components of an Oracle database and progressively move up to the largest entities. Before you begin learning about the logical components, remember that Oracle database objects such as tables, indexes, and packaged SQL code are actually logical entities. Taken together, a set of related database logical objects is called a *schema*. Dividing a database's objects among various schemas promotes ease of management and a higher level of security.

The Oracle *data block* is the foundation of the database storage hierarchy. The Oracle data block is the basis of all database storage in an Oracle database. Two or more contiguous Oracle data blocks form an *extent*. A set of extents that you allocate to a table or an index (or some other object) is termed a *segment*. A *tablespace* is a set of one or more data files, and it usually consists of related segments. The following sections explore each of these logical database structures in detail.