Performance tuning is the one area where the Oracle DBA will probably spend the most of his or her time, whether the DBA is working with a development or a production database. Performance tuning could be proactive as well as reactive. If you’re a DBA helping developers to tune their SQL, you can improve performance by suggesting more efficient queries or table- and index-organization schemes. If you’re a production DBA, you’ll be dealing with user perceptions of a “slow” database, batch jobs taking longer and longer to complete, and so on.

Performance tuning focuses primarily on writing efficient SQL, allocating appropriate computing resources, and analyzing wait events and contention in the system. This chapter focuses on SQL query optimization in Oracle. You’ll learn the important principles that underlie efficient code. You’ll also learn about the initialization parameters you can tune to optimize performance. I present a detailed discussion of the various tools, such as the Explain Plan and SQL Trace utilities, with which you analyze SQL and find ways to improve performance.

Oracle provides several options to aid performance, such as partitioning large tables, using materialized views, storing plan outlines, and many others. This chapter examines how DBAs can use these techniques to aid developers’ efforts to increase the efficiency of their application code.

I begin the chapter with a discussion of how to approach performance tuning. More than the specific performance improvement techniques you use, it’s your approach to performance tuning that determines your success in tuning a recalcitrant application system.

An Approach to Oracle Performance Tuning

Performance tuning is the 800-pound gorilla that is constantly menacing you and that requires every bit of your ingenuity, knowledge, and perseverance to keep out of harm’s way. Your efforts to increase performance or to revive a bogged-down database can have a major impact on your organization, and your results will be monitored and appreciated by users and management.

Unlike several other features of Oracle database management, performance tuning is not a cut-and-dried subject with clear prescriptions and rules for every type of problem you may face. This is one area where your technical knowledge
must be used together with constant experimentation and observation. Practice does make you better, if not perfect, in this field.

Frustrating as it is at times, performance tuning is a rewarding part of the Oracle DBA's tasks. You can automate most of the mundane tasks such as backup, export and import, and data loading—the simple, everyday tasks that can take up so much of your valuable time. Performance tuning is one area that requires a lot of detective work on the part of application programmers and DBAs to see why some process is running slower than expected, or why you can't scale your application to a larger number of users without problems.

**A Systematic Approach to Performance Tuning**

It's important to follow a systematic approach to tuning database performance. It's very common for performance problems to come to the fore only after a large number of users starts working on a new production database. The system seems fine during development and rigorous testing, but it slows down to a crawl when it goes to production. This could be because the application isn't easily scalable for a number of reasons.

The seeds of the future performance potential of your database are planted when you design your database. You need to know the nature of the applications the database is going to support. The more you understand your application, the better you can prepare for it by creating your database with the right configuration parameters. If major mistakes were made during the design stage, and the database is already built, you are left with tuning application code on one hand and the database resources such as memory, CPU, and I/O on the other. Oracle suggests a specific approach to tune your database performance. The steps are as follows:

1. Design the application correctly.
2. Tune the application SQL code.
3. Tune memory.
4. Tune I/O.
5. Tune contention and other issues.

**Reactive Performance Tuning**

Although the preceding performance tuning steps suggests that you can follow the sequence in an orderly fashion, the reality is completely different. Performance tuning is an iterative process, not a sequential process where you start at the top and end up with a fully tuned database as the product. As a DBA, you may be involved in a new project from the outset, when you have just the functional requirements of the project. In this case, you have an opportunity to be involved in the tuning effort from the beginning stages of the application, a phase that is somewhat misleadingly dubbed *proactive tuning* by some. Alternatively, you may come in after the application has already been designed and implemented, and currently is in production. In this case, your performance efforts are categorized as