

## PART 3



# Query Optimizer

*Make the best use of what is in your power, and take the rest as it happens.*

—Epictetus<sup>1</sup>

**E**very single SQL statement sent to the database before being processed by the SQL engine must be turned into an execution plan. In fact, an application specifies only what data must be processed through SQL statements, not how to process it. The aim of the query optimizer is not only to deliver the SQL engine execution plans describing how to process data but also, and most important, to deliver efficient execution plans. Failing to do so may lead to abysmal performance. Precisely for this reason, a book about database performance must deal with the query optimizer.

The aim of this part, however, is not to cover the internal workings of the query optimizer. Other resources, such as Jonathan Lewis's *Cost-Based Oracle* (Apress, 2006), already do an excellent job in this area. Instead, a very pragmatic approach is presented here, aimed at describing the essential features of the query optimizer you have to know. Chapter 4, for example, discusses the statistics used by the query optimizer. Chapter 5 describes the initialization parameters influencing the behavior of the query optimizer and how to set them. Since sooner or later you will be confronted with SQL statements performing badly, Chapter 6 outlines different methods of obtaining execution plans, as well as how to read them and recognize inefficient ones. Finally, Chapter 7 describes several SQL tuning techniques.

---

1. <http://www.quotationspage.com/quote/2525.html>

Before going on to Chapter 4, a very important note: up to Oracle9i, two main optimizers are available, the *rule-based optimizer* (RBO) and the *cost-based optimizer* (CBO). As of Oracle Database 10g, the rule-based optimizer is no longer supported and, therefore, will not be covered here. Throughout this book, when you read the term *query optimizer*, I always mean the cost-based optimizer.