There are a number of tasks in application development. We saw in Chapter 6 to Chapter 9 how to design and build an application. One of the aspects of application design is the performance one expects out of the application. In fact, it is common to find that once an application has been built, it runs slower than the designers wanted or handles fewer transactions per second than they required. An application that takes an excessive amount of time to perform requested actions can cause user dissatisfaction at best and be completely unusable at worst.

Applications can be made to run significantly faster by performance tuning, which consists of finding and eliminating bottlenecks and adding appropriate hardware such as memory or disks. There are many things an application developer can do to tune the application, and there are things that a database-system administrator can do to speed up processing for an application.

Benchmarks are standardized sets of tasks that help to characterize the performance of database systems. They are useful to get a rough idea of the hardware and software requirements of an application, even before the application is built.

Applications must be tested as they are being developed. Testing requires generation of database states and test inputs, and verifying that the outputs match the expected outputs. We discuss issues in application testing. Legacy systems are application systems that are outdated and usually based on older-generation technology. However, they are often at the core of organizations and run mission-critical applications. We outline issues in interfacing with and issues in migrating away from legacy systems, replacing them with more modern systems.

Standards are very important for application development, especially in the age of the internet, since applications need to communicate with each other to perform useful tasks. A variety of standards have been proposed that affect database-application development, which we outline in this chapter. Organizations often store information about users in directory systems. Applications often use such directory systems to authenticate users and to get basic information about users, such as user categories (e.g.,
student, instructor, and so on). We briefly describe the architecture of directory systems.

Bibliographical Notes

The classic text on queueing theory is [Kleinrock (1975)].

An early proposal for a database-system benchmark (the Wisconsin benchmark) was made by [Bitton et al. (1983)]. The TPC-A, -B, and -C benchmarks are described in [Gray (1991)]. An online version of all the TPC benchmark descriptions, as well as benchmark results, is available on the World Wide Web at the URL www.tpc.org; the site also contains up-to-date information about new benchmark proposals. The OO1 benchmark for OODBs is described in [Cattell and Skeen (1992)]; the OO7 benchmark is described in [Carey et al. (1993)].


Information about ODBC, OLE-DB, ADO, and ADO.NET can be found on the Web site www.microsoft.com/data and in a number of books on the subject that can be found through www.amazon.com. ACM Sigmod Record, which is published quarterly, has a regular section on standards in databases.

A wealth of information on XML-based standards and tools is available online on the Web site www.w3c.org. Information about RosettaNet can be found on the Web at www.rosettanet.org.

Business process re-engineering is covered by [Cook (1996)]. [Umar (1997)] covers re-engineering and issues in dealing with legacy systems.

Bibliography


Credits

The photo of the sailboats in the beginning of the chapter is due to ©Pavel Nesvadba/Shutterstock.