CHAPTER 1

■ ■ ■

SQL Server 2005 Overview and Installation

Welcome to *Beginning SQL Server 2005 for Developers*. As you are reading this book, I assume that you are interested in learning how to create solutions with Microsoft SQL Server 2005, but have no prior knowledge of SQL Server 2005. You may well have had exposure to other databases such as MySQL, Oracle, or Microsoft Access, but SQL Server uses different interfaces and has a different way of working compared to much of the competition. The aim of this book is to bring you quickly up to a level at which you are developing competently with SQL Server 2005. This book is specifically dedicated to beginners, and to those who at this stage wish to use only SQL Server 2005. You may find this book useful for understanding the basics of other databases in the marketplace, especially when working with T-SQL. Many databases use an ANSI-standard SQL, and so moving from SQL Server to Oracle, Sybase, etc., after reading this book will be a great deal easier.

This chapter covers the following topics:

- Why SQL Server 2005?

- How do I know if my hardware meets the requirements?

- Can I just confirm that I have the right operating system?

- What can I do with SQL Server 2005?

We will also then look at installing our chosen edition—this section of the chapter covers the following:

- Installing SQL Server 2005 on a Windows XP platform

- Options not installed by default

- Where to install SQL Server physically

- Multiple installations on one computer

- How SQL Server runs on a machine

- How security is implemented

- Logon IDs for SQL Server, especially the sa (system administrator) logon

Why SQL Server 2005?

The following discussion is my point of view, and although it no doubt differs from that of others, the basis of the discussion holds true. SQL Server faces competition from other databases, not only from other Microsoft products such as Microsoft Access and Microsoft Visual FoxPro, but also from competitors like Oracle, Sybase, DB2, and Informix, to name a few.

Microsoft Access is found on a very large number of PCs. The fact that it is packaged with some editions of Office and has been around for a number of years in different versions of Office has helped make this database ubiquitous; however, a great number of people actually do use the software. Unfortunately, it does have its limitations when it comes to scalability, speed, and flexibility, but for many small, in-house systems, these areas of concern are not an issue as such systems do not require major database functionality.

Now we come to the serious competition: Oracle and Sybase. Oracle is seen as perhaps the market leader in the database community, and has an extremely large user base. There is no denying it is a great product to work with, if somewhat more complex to install and administer than SQL Server; it fits well with large companies that require large solutions. There are many parts to Oracle, which make it a powerful tool, including scalability and performance. It also provides flexibility in that you can add on tools as you need them, making Oracle more accommodating in that area than SQL Server. For example, SQL Server 2005 forces you to install the .NET Framework on your server whether you use the new .NET functionality or not. However, Oracle isn't as user friendly from a developer's point of view in areas like its ad hoc SQL Query tool and its XML and web technology tools, as well as in how you build up a complete database solution; other drawbacks include its cost and the complexity involved in installing and running it effectively. However, you will find that it is used extensively by web search engines, although SQL Server could work just as effectively. With the new functionality in SQL Server 2005, Oracle will be under pressure to expand its existing functionality to meet this challenge. SQL Server has always been a one-purchase solution, such that (providing you buy the correct version) tools that allow you to analyze your data or to copy data from one data source such as Excel into SQL Server will all be "in the box." With Oracle, on the other hand, for every additional feature you want, you have to purchase more options.

Then there is Sybase. Yes, it is very much like SQL Server with one major exception: it has no GUI front end. Sybase Adaptive Server Anywhere, which is mainly used for small installations, does have a front end, but the top-of-the-range Sybase does not. To purists, there is no need for one, as GUI front ends are for those who don't know how to code in the first place—well, that's their argument, of course, but why use 60+ keystrokes when a point, click, and drag is all that is required?

Sybase is also mainly found on Unix, although there is a Windows 2000 version around. You can get to Sybase on a Unix machine via a Windows 2000/XP machine using tools to connect to it, but you still need to use code purely to build your database solution. It is very fast and very robust, and it is only rebooted about once, maybe twice, a year. Another thing about Sybase is that it isn't as command-and-feature rich as SQL Server. SQL Server has a more powerful programming language and functionality that is more powerful than Sybase.

Each database has its own SQL syntax, although they all will have the same basic SQL syntax, known as the ANSI-92 standard. This means that the syntax for retrieving data, and so on, is the same from one database to another. However, each database has its own special syntax to maintain it, and trying to use a feature from this SQL syntax in one database may not work, or work differently, in another.