Chapter 1

The Joy of SQL Server 2005 Programming

In This Chapter

- ▶ Choosing the right SQL Server edition for your programming goals
- ▶ Gathering and manipulating data
- ▶ Enforcing business rules for your database
- ▶ Ensuring SQL Server security
- ▶ When Transact-SQL isn't enough

SQL Server 2005 builds on the existing strengths of SQL Server 2000 to help you build applications that retrieve and manipulate data to suit your business needs. SQL Server 2005 continues to support Transact-SQL (T-SQL) as the primary language for the manipulation of relational data but has also added new functionality to allow you to work better with XML and to use .NET languages in your applications.

SQL Server 2005 allows you to flexibly create powerful applications based on its relational tables. Traditionally, you had to use Transact-SQL for those applications, and often that remains the programming language of choice.

When you create applications based on SQL Server 2005, you need to consider the goals of your programming before you do any coding. In this chapter, I discuss how you can define your programming goals and gather and manipulate data in SQL Server 2005. A well-designed SQL Server 2005 application enforces the business rules that your company has defined. If your data is automatically checked for conformity to those rules, you can have increased confidence that the application supports your business objectives.

One of the tasks that many applications carry out frequently is retrieval of data from SQL Server 2005. This process is based on the Transact-SQL `SELECT` statement. In Chapter 4, I show you the basics of using the `SELECT` statement. In Chapter 5, you discover how to use joins that select data from multiple SQL Server tables to create a result set for a query.

Security of your data is enormously important. You want to make sure that authorized users have access to the data they need and ensure that unauthorized people do not have access to your data. I discuss security in detail in Part IV.

At times, the conventional approach of using Transact-SQL with relational data isn't enough. Sometimes you want to use a .NET programming language to carry out calculations that Transact-SQL isn't well-suited for. I introduce you to using the Common Language Runtime in SQL Server 2005 in Chapter 21 and show you some techniques you can use in Visual Studio 2005 in Chapter 22. You may also want to store XML (Extensible Markup Language) data, which SQL Server 2005 supports; see Chapter 20 for details.

Deciding Which Version of SQL Server 2005 to Use

When you program with SQL Server 2005, you need to decide early on what you want your application to do. This book can't tell you what the functionality of your application ought to be; you need to decide who will use the application you create and what they need to be able to do with it. Your goals can determine which edition of SQL Server 2005 you need to buy. They can also influence which version of the Windows operating system you need: Not all editions of SQL Server 2005 run on all current Windows operating systems.

TIP

If you need detailed installation information about SQL Server 2005, check out SQL Server 2005 Books Online at `http://msdn2.microsoft.com/en-us/library/ms130214.aspx`. Detailed information about installing SQL Server 2005, including which editions to install on which operating-system versions, is also available at `http://msdn2.microsoft.com/en-us/library/ms143516.aspx` and related pages.

Here is a brief rundown of the various SQL Server 2005 editions:

- **Express:** If you want to simply teach yourself the basics of Transact-SQL and explore the basics of how you can use Visual Basic.NET or Visual C# in SQL Server 2005, you can use the Express Edition of SQL Server 2005.

  SQL Server 2005 Express Edition has some features that are not included in other editions of SQL Server 2005. In this book, I don't cover features that are present only in Express Edition.

  For production use, you can use SQL Server 2005 Express Edition if your application will run adequately on the limited specification and functionality of Express Edition. Check the Features Comparison Web page at `www.microsoft.com/sql/prodinfo/features/compare-features.mspx` to see if Express Edition meets your needs.

- **Developer:** The Developer Edition of SQL Server 2005 allows you to explore all the features of any edition of SQL Server 2005. The Developer Edition, which is modestly priced, is technically the same as the Enterprise Edition except that it is not licensed for production use. You can work through the example techniques shown in this book using the Developer Edition.

  Installing the Developer Edition on Windows XP Professional is a cost-effective way to learn SQL Server 2005 programming. It enables you to avoid the much larger licensing costs of Workgroup, Standard, and Enterprise Editions, while allowing you to carry out any SQL Server 2005 programming task.

- **Workgroup, Standard, and Enterprise:** If Express Edition doesn't meet your production needs, you have a choice of Workgroup, Standard, and Enterprise Editions. Again, a detailed feature-by-feature comparison is available at `www.microsoft.com/sql/prodinfo/features/compare-features.mspx`.

Gathering and Manipulating Data

In most SQL Server–based applications, you use a custom interface for data entry. Because you can create such an input application only after you know how to use Transact-SQL and have some understanding of creating applications with Visual Studio 2005, I often use SQL Server Management Studio's functionality to input data in the chapters of this book.

When it comes to manipulating data in practice, you will use custom applications. However, SQL Server Management Studio is a good teaching tool to help you find out more about the individual parts of Transact-SQL. I have written the chapters in this book in such a way that you can follow the examples by simply reading the text and looking at the figures. However, you learn much more if you open SQL Server Management Studio and run each example. By typing in the Transact-SQL or other code yourself, you are forced to pay much more careful attention to the exact syntax of each command. There is no substitute for actually coding.

Enforcing Business Rules for Your Database

SQL Server 2005 provides several ways for you to enforce the rules that you use when running your business. Constraints provide one way of enforcing some classes of business rules. A *constraint,* as the name suggests, constrains the values that can be inserted into a column. If, for example, you are running a club that allows members of 18 or more, you might constrain an age or date-of-birth column to reflect that rule. I show you how to use constraints in Chapter 10.

Another approach to enforcing business rules is the use of triggers. Typically, these are Data Modification Language (DML) triggers. A DML trigger fires in response to some specified event in a SQL Server database. For example, if somebody changes the data in a particular table, the trigger you have defined may automatically audit who made the changes and when. Having this trigger provides an audit trail that tells you who did what to your database. You find out how to use triggers in Chapter 13.

Ensuring SQL Server Security

Keeping your SQL Server data secure is hugely important. In a worst-case scenario, unauthorized access to your data could cripple your business if stored data is maliciously damaged or competitors are allowed access to confidential information.

SQL Server 2005 security is based on logins (at the SQL Server instance level) and users (at the database level). The permissions you grant or deny to a specified login or user can be applied in a very granular way. I introduce you to logins and users in Chapter 17.

In addition, schemas group database objects in ways that are convenient to allow change of ownership; for example, when an employee leaves the company.

One security concern for Web-facing database applications is *SQL injection.* A malicious user can shape the data entered in a Web form so that SQL Server treats it like Transact-SQL code. One way of minimizing that risk is to use stored procedures to process data entered into Web forms — and treat the data entered by a user as parameters to such stored procedures. If a malicious user attempts to enter malicious input, it likely won't take the form required of a stored procedure parameter, and an error will result. The bonus is that the malicious code isn't executed in a way that may damage your data or compromise its future security. I show you how to create stored procedures in Chapter 12.

When Transact-SQL Isn't Enough

Transact-SQL is an immensely powerful and flexible language for data retrieval and manipulation. But in some situations, you may want to do things with your data that traditional Transact-SQL isn't suited to doing.

In SQL Server 2000, if you wanted to carry out complex calculations, you quite possibly used extended stored procedures that had potential reliability and security concerns. In SQL Server 2005, you can use Visual Basic.NET or Visual C# to create software modules that carry out complex calculations (or any other suitable task) in ways where SQL Server security is more specifically controlled.

SQL Server 2005 provides new functionality that allows you to store XML data directly in a column in a SQL Server 2005 table by using the new `xml` data type. This functionality complements the existing XML-related functionality where you could break XML into relational data for storage and manipulate retrieved relational data into an XML form.