CHAPTER 1

What Is SQL Server?

SQL Server is a relational database management system (RDBMS) developed by Microsoft. In order to find out exactly what that means, we need to ask three more questions:

- What is SQL?
- What is a server?
- What is an RDBMS?

To answer the first question, SQL is both a programming language and an acronym. It can be pronounced either "sequel" or literally with the letters *SQL*. It stands for Structured Query Language, and it is the language used to talk to databases. So from this, we can infer that SQL Server is actually a database server, which brings us to our second question: what exactly is a server?

A server is a computer that provides some kind of service (or multiple services). For example, a print server processes print requests, a web server hosts web sites, and a database server runs a database management system. This leads to the last question: what is an RDBMS?

An RDBMS is a collection of services that together support the storage and retrieval of relational data from some kind of file store. We'll take a look at the various servers provided by the SQL Server RDBMS a bit later in this chapter. But for now, I've answered our original question, and you have an idea of what SQL Server is.

In this chapter, we'll:

- take a quick SQL Server history tour,
- discover the services provided by SQL Server, and
- look at the types of databases used by SQL Server.

A Quick History of SQL Server

Not everybody likes history, so I'll keep this brief! In the 1980s, Oracle dominated the database market (to a certain extent, it still does). Late in that decade, Microsoft wanted to start competing with Oracle, so they built a database system together with another company, Sybase. This fared well enough to encourage Microsoft to continue development. In the mid-1990s, Microsoft and Sybase parted ways and Microsoft completely redeveloped SQL Server. The first major rewritten version was version 7.0, released in 1998. If you were to look at the internals of version 7.0 today you'd see a lot of similarities with the current version, SQL Server 2014.

Let's have a quick look at the versions released since version 7.0, along with core features added:

- **1998**: Version 7.0. A complete rewrite of the core database system. You are unlikely to see a version 7.0 system running today (although some do exist).
- **2000**: SQL Server 2000 released. Major performance improvements were included, along with basic XML support and an import/export tool called DTS (Data Transformation Services). Lots of systems still use SQL Server 2000 and higher today.
- **2005**: Major XML support is added to SQL Server 2005, along with the ability to run .NET code within the database.
- **2008**: SQL Server 2008 and 2008 R2 came along. The ability to store documents, pictures, videos, and other types of data directly in a structured manner was added, along with new ways to store dates and times.
- **2012**: You guessed it; the new version was called SQL Server 2012. This added greatly improved database availability features such as AlwaysOn and enhanced interaction with Windows Azure.
- **2014**: The version I am talking about, SQL Server 2014, was made available. One big new feature here was in-memory tables.

You don't need to exhaustively know the history of SQL Server, but having some knowledge of the past can inform the present. If you are planning to go into databases as a career, you may come across older versions of SQL Server. It can help to know what those versions can and can't do. Let's see what the current version of SQL Server—2014—can do.

SQL Server Services

A number of services make up SQL Server. Let's take a quick look at them:

- **Database Engine**: This is the core service, and most of the things we look at in this book will be using this service. It is responsible for all aspects of data management. If you want to add some data to a table, create a database, or pull some data from your databases, you'll be interacting with this service.
- **SQL Server Agent**: SQL Server's scheduling tool. You can use Agent to set jobs up to run at different times of the day, week, or month. Jobs are often created to run backups or other regular tasks.
- **SQL Server Browser**: If your SQL Server sits on a network, you probably have other machines connecting to it. Connecting remotely requires the SQL Server Browser service to be up and running. This allows other machines on the network to communicate with SQL Server. However, if SQL Server was installed with the default port and instance name configuration, then SQL Server Browser can be safely disabled. This service is disabled by default, so it must be enabled in order to be used.
- **SQL Server Full Text Search**: Let's say our database has a user record named "Mike McQuillan." I can use the SQL language to directly return this record—*but only if I specify the full name*. What if I don't know the full name? Assume all I know is "Mike." I can still find the record using the standard SQL Server `LIKE` operator. But it won't be a fast lookup.

  Now imagine you were searching for "Mike" using something like Google or Bing. You'd just enter "find everybody called Mike" and voilà! The records would be returned. You can think of the Full Text Search as the Google/Bing part of SQL Server. It can be used to run complex searches on your database; it can even find words that sound similar or have similar meanings.

  Again, we won't look at this service in this book, but remember it exists and it may save you one day!

Business Intelligence Services

SQL Server can be used as a complete Business Intelligence (BI) platform. This platform allows you to import data from databases and other systems, analyze it, and generate reports from that data. The aim of a BI system is to help your organization plan for the future.

The BI platform supplied by SQL Server is vast and won't be covered in this book. It consists of three services, which I introduce here so you have a brief idea of what they are and what they are for:

- **SQL Server Integration Services (SSIS)**: This is SQL Server's ETL service—Extract, Transform, and Load. In other words, this service can import data to or export data from your databases. It can actually import or export to pretty much anything that is machine readable. You can also make changes to the data (Transform) as it goes in or out of the system. SSIS is the answer to the deprecated DTS functionality in SQL Server 2000, since it allows for the creation of import/export utilities.
- **SQL Server Reporting Services (SSRS)**: Have you ever received a bank statement, or perhaps some kind of application form? The chances are these documents are based on reporting templates. Somebody somewhere has created a template of a report, which is then used to generate individual statements, forms, letters, and so on for individual customers. SSRS is the service SQL Server provides to allow you to create these templates.
- **SQL Server Analysis Services (SSAS)**: If your company wants to put some kind of reporting platform together, it could do a lot worse than SSAS. This huge component of SQL Server allows data to be grouped and structured in such a way that reporting becomes easy. You can use SQL Server Management Studio, Excel, .NET programs, or lots of other tools to interact with SSAS. It can perform analyses over time periods, individuals, items, and anything else you can think of. It also supports drilling down into your data via a process called *data mining*.

  Unfortunately, SSAS is just too big a topic to cover in a basics book, even at a simple level. Dedicated books for SSAS exist, and even some of those weighty tomes are far from comprehensive.

  There is a strong chance you'll never need to touch SSAS. But just be aware that it is there and give yourself an idea of what it can do; you never know when it may come in handy.

Editions of SQL Server

There are many different versions of SQL Server, each tailored for a specific need. Here's a quick rundown of what each edition is for:

- **Enterprise**: This is the highest version of SQL Server available. It includes every feature available but is extremely expensive.
- **Business Intelligence**: Aimed specifically at organizations using SQL Server as a BI platform.
- **Standard**: The version of SQL Server most commonly found in production environments. This provides solid data management and BI features, which is usually more than enough for most organizations. It lacks certain high-end features (e.g., in-memory tables) that are found in the Enterprise Edition.
- **Web**: Supports scaling options for SQL Servers supporting web sites.
- **Developer**: This is exactly the same as Enterprise Edition, but intended for developer use only. The rule here is you cannot use this version in production. If you are serious about developing in SQL Server, this is the version to go for. This is great if you are studying for SQL Server certification exams, for example. It's also inexpensive (around $60/£50).
- **Express**: The easiest way to try SQL Server. This is a free version, which has most of SQL Server's core features, but is limited in certain aspects (e.g., the maximum database size allowed is 4GB).

We'll be using SQL Server Express in this book, and we'll look at downloading and installing it in the next chapter.

All versions of SQL Server come as either 32-bit or 64-bit. The 64-bit versions can access more memory (if supported). Which version you want depends on your processor. If you are in doubt, open PC Info (Windows 8) or System in Control Panel (Windows 7). This will tell you whether your processor is 32-bit or 64-bit. The 32-bit version will run on 64-bit systems, so if you are still in doubt, download the 32-bit edition. Unless you have specific requirements, this decision won't hugely affect development machines.

SQL Server Tools and Utilities

SQL Server comes with a number of programs that can be used to manage various aspects of the system. There are also a bunch of utilities to help you perform all sorts of tasks:

- **SQL Server Management Studio (SSMS)**: Learn to love this program—you'll use it pretty much every day. This is the standard interface that 99% of SQL Server DBAs use. SSMS allows you to manage multiple SQL Servers, write and run SQL scripts, and also manage other services such as SSIS and SSAS. This program is amazingly powerful and we'll use it throughout this book.
- **SQL Server Data Tools (SSDT)**: If you've done any .NET programming, you've probably used Microsoft Visual Studio. This is a programming environment for .NET, much as SSMS provides a scripting environment for SQL Server. SSDT is based upon Visual Studio and allows you to create and edit SSIS, SSRS, and SSAS solutions.
- **SQL Server Configuration Manager**: Should you have to manage any of the services involved with SQL Server, you can use this utility. You can start and stop services, as well as manage network configuration.
- **SQL Server Profiler/Extended Events**: If you are having problems with particular queries, you can use one of these tools to analyze the query. They can also identify if locks are occurring within databases, among other things. It is important to note that SQL Server Profiler is being deprecated in a future version of SQL Server, so at some point it won't be around. Extended Events (built into SSMS) should be used instead. We'll look at both of these tools in Chapter 20.
- **SQLCMD**: SQLCMD—short for SQL Command—is a utility that allows users to execute queries and manage various aspects of the server from the command line. It can also be called from SSMS, as we'll see in future chapters.
- **SQL Server Import and Export Wizard**: If you need to run a fairly simple import to or export from a database, you can use this wizard. Alternatively, you can create your own routines using SSIS.
- **SQL Server Installation Center**: Think of this as an installation wizard on steroids. If you need to change any aspect of your SQL Server installation (e.g., add SSIS to a server), you'll use the Installation Center.

- **Reporting Services Configuration Manager**: Once SSRS is installed, you need to configure it. This handy little utility gives you the interface to do just that.
- **Database Engine Tuning Advisor**: If your database is slow, run this utility. It will give you some basic advice on indexes and foreign keys that are missing.

Many of these utilities can make your life easier; the Import And Export Wizard can be extremely useful. Find out what each one does, and then you'll have them in your back pocket should you ever need them. We'll use some of these tools and utilities throughout the book.

Time to Kick Off

I've given you a pretty good grounding in what SQL Server can provide. You now know what the various components of SQL Server are, and which utilities are available to use with SQL Server.

Now it's time to install SQL Server on our computers so we can actually do something with it. It's download time!

PREV
Acknowledgments

NEXT
Chapter 2 : Obtaining and Instal…