Chapter 1

Introduction to SQL Server 2000 and Relational Databases
SQL Server 2000 is a client/server-based relational database management system that runs on Windows 2003; Windows XP; Windows 2000 Professional, Server, and Advanced Server; Windows NT 4; Windows 9x/Millennium; and Windows CE. In this chapter, we'll begin by defining a client/server environment, the types of databases involved, and what they contain. We'll then provide some background material on SQL Server and Windows operating systems, including the new features of SQL Server 2000. In addition, we'll discuss the tasks of the SQL Server developer and those of the SQL Server administrator.

Even though there are no Microsoft SQL Server 2000 exam objectives directly discussed in this chapter, you should still take the time to understand and know the issues and topics presented here. In later chapters, we will build up many of the ideas presented in this chapter.

What Is Client/Server?

Microsoft's SQL Server is a client/server database engine. SQL Server is the server part of the equation.

Client/server describes an application that is split into at least two logical parts: One part runs as a server, and the other part runs as a client. The server side of the application provides security, fault tolerance, improved performance, concurrency control, and reliable backups. The client side provides the user interface and may contain such items as reports, queries, and forms.

In older, non-client/server database systems, the work is not shared between the server and workstation machines. Take a look at Microsoft Access; suppose you have a 10MB database stored on your network server when a client opens the database and runs a query, all of the data needed to complete the query is downloaded to the client and then the query is processed at the client computer (Figure 1.1). In order to accomplish the processing at the client, each client must have copy of the database engine (Jet 4.0 in this case).

In contrast, when a query is run on a client/server system, the server searches the database, processes the data request, and sends back to the client just those rows that match the search conditions (Figure 1.2). This not only saves bandwidth on your network, it (if the server is a powerful enough machine) is often faster than having the workstations do all the work (although even in client/server computing, workstations still do some of the work). Take note that the client computer needs only to have some way to communicate with the database server. It does not need to have a copy of the database engine at each local client.
Types of Databases

A database can generally be thought of as a collection of related data that can be organized in various ways. The database can be designed to store historical data, as in data warehousing.
environment where Online Analytical Processing (OLAP) takes place, or changing data, as in an environment where Online Transaction Processing (OLTP) takes place. In the following sections, we will take a look at the different types of databases.

**Relational Databases vs. Flat-File Databases**

In earlier database products, a database usually consisted of one file per table or index—something like `Payroll.dbf` for an employee table or `Patients.dbf` for a patients table. When all of a table’s information is stored in files and accessed sequentially, the database resembles a spreadsheet and is considered a flat-file database. Many older database applications were flat-file databases, such as Paradox, Btrieve, FoxPro, dBase, Mumps, and many others. Although flat-file databases are relatively easy to set up, those that contain complex data are more difficult to maintain. A relational database is composed of tables that contain related data. The process of organizing data into tables (in a consistent and complete format) is referred to as normalizing the database.

Normalization of relational designs is a complex topic that will not be addressed in this book; it can be found in other books that are devoted to this topic. Before you implement your database design, however, you should start with a fully normalized view of the database.

In a relational database management system (RDBMS), such as SQL Server, although a database is held on one or more operating system files, it is more of a logical concept based on a collection of related objects. A database in SQL Server not only contains the raw data, it also contains the structure of the database, the security mechanisms of the database, and any views or stored procedures related to the database.

A SQL Server database is composed of different types of objects (see Figure 1.3). Table 1.1 lists some of the more common types.

**FIGURE 1.3** Common database objects
### Table 1.1 Common Database Objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>A table is the object that contains rows, columns, and the actual raw data.</td>
</tr>
<tr>
<td>Column</td>
<td>A column is a part of the table—the object that holds the data.</td>
</tr>
<tr>
<td>Row</td>
<td>A row is not a separate object but rather a complete set of columns within a single table. Unlike the rest of the items discussed here, it is a logical construct, not a physically implemented object.</td>
</tr>
<tr>
<td>Datatype</td>
<td>There are various datatypes that can be used in columns, such as character, numeric, and date. A column can hold only data of a single datatype.</td>
</tr>
<tr>
<td>Stored procedure</td>
<td>A stored procedure is a set of Transact-SQL (T-SQL) statements that are combined to perform a single task or set of tasks. This object is like a macro in that SQL code can be written and stored under a name. By invoking the name, you actually run the code.</td>
</tr>
<tr>
<td>Trigger</td>
<td>A trigger is an object that is a special type of stored procedure that activates when data is added to, edited, or deleted from a table. Triggers are used to implement business rules in the database. For example, an automobile dealer’s database might contain a trigger that activates when a car is sold and credits the appropriate salesperson with a bonus depending on the profit from the sale.</td>
</tr>
<tr>
<td>Rule</td>
<td>A rule is an object that is assigned to a column so that data being entered conforms to standards you set. For example, rules can be used to make sure that the state in a person’s address is entered in uppercase letters and that phone numbers contain only numbers. Rules are supported for backwards compatibility. We suggest that you use Constraints instead.</td>
</tr>
<tr>
<td>Default</td>
<td>A default is an object that is assigned to a column so that, if no data is entered, the default value will be used. For example, you might set the state code to the state where most of your customers reside so the default can be accepted for most entries. Defaults (as an object, not a column level constraint) are supported for backwards compatibility. We suggest that you use Constraints instead.</td>
</tr>
<tr>
<td>View</td>
<td>A view is much like a stored query that acts as a table. You don’t need to rewrite the query every time you’d like to run it; you can use the view instead. A view appears very much like a table to most users. A view usually either excludes certain fields from a table or links two or more tables together.</td>
</tr>
<tr>
<td>Index</td>
<td>An index is a collection of stored data or metadata about your data. Using an index makes retrieval of the data much faster.</td>
</tr>
<tr>
<td>Constraint</td>
<td>Constraints are ANSI compliant objects that allow you to specify default values for a column or even rules and lookups for a column. They can also be used to enforce referential integrity.</td>
</tr>
</tbody>
</table>
Data Warehousing vs. Online Transaction Processing

Databases fall into two general categories:

- Stored historical data that can be queried. This is often referred to as a data warehouse or a Decision Support Systems (DSSs). The main focus of data warehousing is the ability to quickly query existing data and perform complex analyses, usually looking for patterns or other relationships that are difficult to locate during the day-to-day operations of a company. This is often referred to as OLAP (Online Analytical Processing).

- Live, continually changing data. This type of database activity is referred to as an Online Transaction Processing (OLTP) environment. In this case, the flexibility to expand and edit the database is the main focus.

Although these types of database activities may appear to be the same at first glance, they are basically opposites. Data warehousing emphasizes reading the historical database with long-running transactions; OLTP emphasizes writing to a live, production database with very quick transactions.

SQL Server makes a great server for both OLTP and data warehousing types of databases, but if your company uses both types, you may want to consider using two SQL Server machines—one for each function—in order to get optimal performance.

Another related (and rapidly growing) use of databases is to put the data into a web page and publish it either internally on an intranet or to the public on the Internet. Again, the same basic types of database activities are used: data warehousing if the data on the web page is for historical reporting purposes or OLTP if the customer can order from the web page.

Some Background on SQL Server

SQL Server 2000 continues a strong tradition of easy-to-use relational database engines that are also very powerful. While earlier versions of SQL Server have allowed many companies to downsize from mainframes to networks based on personal computers, SQL Server 2000 adds many important new functions, including an interface that’s vastly easier to use and many more automated tuning and optimization features. More than two million copies of Microsoft’s SQL Server have been sold since it was first introduced, and it is installed in a significant percentage of the companies that use some kind of database server.

SQL Server is one of the cornerstones of Microsoft’s strategy for its .NET Enterprise Servers family of server applications (Microsoft’s line of client/server support applications). Microsoft’s other .NET software includes Windows Server 2003, Exchange 2000 (the e-mail server), Internet Information Server (IIS, the WWW server), Systems Management Server 2 (the workstation management piece), Internet Security and Acceleration (ISA) server, and Host Integration Server (the next-generation SNA protocol connectivity server) as well as other software that
Some Background on SQL Server

SQL runs on Windows operating systems. Larger enterprise-style applications like Portal Server and Content Management Server and BizTalk server are slated for release.

Every company has data, and almost every program generates even more data. Data in the paperless office needs to be stored somewhere—usually in some kind of data server.

Before you work with SQL Server system administration, you need to understand its native language and history and how it works with Windows operating systems.

History of Structured Query Language (SQL)

During the 1970s, IBM invented a computer language designed specifically for database queries. This language was called SEQUEL, which stands for Structured English Query Language. IBM released SEQUEL into the public domain, where it soon became known as SQL. Over time, the language has been expanded so that it is not just used for queries but can also be used to build databases and manage the security of the database engine. Currently, there are various versions of SQL in use. Microsoft SQL Server uses a version called T-SQL, or Transact-SQL (both stand for Transaction SQL).

T-SQL can be divided into three generic categories of commands:

- Database schema commands, also known as data definition language (DDL), are used to define the database, tables, and other objects that are used for managing the structure of the database.
- Data management commands, also known as data manipulation language (DML), are used much like the original SQL commands in that they manipulate, add to, or delete data in the database.
- Data control language (DCL) is used to assign permissions to objects within the database. This gives you very granular control over the security of your database.

Although T-SQL is the programming language used by SQL Server, in this book our emphasis is on installing, maintaining, and connecting to SQL Server. In some cases, however, using T-SQL is one way to accomplish some of these tasks, such as creating and modifying SQL Server logons. Many of the exercises will show you both graphical and syntax methods of doing things because you may be tested on both.

History of Microsoft’s SQL Server and Windows 2003

Microsoft initially worked with Sybase Corporation on a version of a SQL server system for OS/2. When Microsoft abandoned OS/2 in favor of its new network operating system, Windows NT, it decided to enhance the SQL server engine from Sybase and help modify the code for Windows NT. The resulting product was Microsoft SQL Server 4 for Windows NT, stabilizing at 4.21a.

Over time, Microsoft took over more and more responsibility for the development of SQL Server. By version 7, Microsoft was in complete control of the software, having redesigned
SQL Server from the ground up. Microsoft’s current version of SQL Server (SQL Server 2000) was officially launched in the last quarter of 2000.

<table>
<thead>
<tr>
<th>A Brief History of Windows 2000/2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft started developing Windows NT in 1990 when it became obvious that running a 16-bit version of Windows on top of MS-DOS could not provide the performance and security that businesses needed. Windows NT 4 came out in 1996, and Windows 2000 came out in early 2000.</td>
</tr>
<tr>
<td>Windows 2000 is a 32-bit operating system that has been designed with a familiar user interface (basically an upgraded look of the Windows 9x interface). One way to look at Windows 2000 is that it is much like a mainframe operating system under the hood, but the user interface is a friendly desktop. Windows 2000 has both server and client properties; that is, it can simultaneously function as a network server and as a client on the network. We’ll talk more about client/server systems in the next section.</td>
</tr>
<tr>
<td>Windows 2000 comes in four versions:</td>
</tr>
<tr>
<td>• A Professional (Workstation) version that is primarily used on desktop and laptop computers but can act as a server for up to 10 simultaneous connections at a time.</td>
</tr>
<tr>
<td>• A Server version that is primarily used as a file/application server that can theoretically have thousands of simultaneous users connected to it.</td>
</tr>
<tr>
<td>• An Advanced Server version that contains all of the server functions plus clustering and support for more CPUs.</td>
</tr>
<tr>
<td>• A Datacenter version that is designed to be installed with SQL Server 2000 to provide the ultimate Microsoft database solution. Datacenter is OEM (Original Equipment Manufacturer) only—in other words, you cannot purchase it alone and install it yourself. It is designed to compete with the high-end Unix servers.</td>
</tr>
<tr>
<td>Windows 2000 is the ultimate general-purpose server. You can add functionality to the base server by installing services. SQL Server is implemented as a set of services on either Windows 2000 Professional or Windows 2000 Server and Advanced Server.</td>
</tr>
<tr>
<td>Windows 2003 has enhanced much of the functionality found in Windows 2000 and Windows XP. The .NET Framework, Active Directory, and many other services have now been integrated as base levels of the operating system. This enhances speed, security, and ease of administration.</td>
</tr>
</tbody>
</table>

**SQL Server 2000 Basics**

Although SQL Server 2000 is similar to earlier editions in many respects, Microsoft has continued to modify and differentiate its several versions of SQL Server to meet the differing needs of various users.
SQL Server 2000 not only adds a lower-end version (the Personal version called MSDE (Microsoft Database Engine)), it also adds support on the high end for various clustered Windows 2000 servers. SQL Server 2000 continues the tradition of strong client support by providing backward compatibility with older SQL clients while also adding new, enhanced client features.

Microsoft’s strong commitment to XML and web-based interoperability has also been seen with recent service packs and an optional SQLXML download from Microsoft. SQL Server has fully embraced XML and the Internet and can now support many types of XML/Internet-related interfaces. This includes updategrams and diffgrams. SQL Server can also generate XML datasets in several different formats.

**Versions of SQL Server 2000**

SQL Server comes in five basic versions:

- Windows CE
- Personal
- Developer
- Standard
- Enterprise

**Windows CE Version**

The Windows CE version of SQL Server 2000 is designed for use on Windows CE devices. This version can replicate data to and from Standard and Enterprise versions of SQL Server. The CE version does have limited scope and capabilities, but keep in mind that it is running on a handheld PDA device or cell phone. You don’t need a full-blown server type of application. You need fast and efficient access to small and medium amounts of data. SQL Server CE delivers and can even wirelessly exchange that data with a Standard or Enterprise edition of SQL Server running replication.

**Personal Version**

SQL Server 2000 can be run on the Windows 9x and later operating systems. The Personal version of SQL Server is meant to be used for traveling applications that occasionally link back to the primary network and for ease of use in developing applications. The Personal version installs and operates very much like the Standard and Enterprise versions running on Windows servers. If you purchase and license the Standard or Enterprise version of SQL Server, you can install the Personal version on any of your clients without having to purchase additional licenses.

**Developer Version**

The Developer version is designed to be used by developers on a single computer that no one connects to via the network. The major advantage of the Developer version is that it is much cheaper than the Standard version and it works just fine if all you are doing is developing applications and code on it. (The Developer edition of SQL Server 2000 will run on any 32-bit Microsoft operating system.)
Standard Version
The Standard version of SQL Server 2000 is designed to support workgroups and small departments. It comes with all of the core features of SQL Server 2000 but lacks features that allow it to scale out to a large number of servers. It also has limited support for OLAP activities.

Enterprise Version
The Enterprise version of SQL Server 2000 includes all of the features of SQL Server 2000, including clustering support, log shipping, parallel computing support, enhanced read-aheads, partitioning support, HTTP support, and very large database (VLDB) support.

Clients Supported by SQL Server 2000
SQL Server 2000 supports the following clients directly:
- Windows 9x and Millennium
- Windows NT
- Windows 2000
- Windows 2003

SQL Server 2000 Features
There are many features of SQL Server 2000 that make it a compelling platform for client/server computing. Although a number of these features were present in version 7, they have been enhanced in 2000. These features include the following:
- Support for both the Windows 9x and Windows 2000 and later operating systems
- Identical application programming interface (API) support for both the Windows 9x and Windows 2000 and later operating systems
- Integrated Online Analytical Processing (OLAP) server
- Integrated Simple Network Management Protocol (SNMP) support via Windows 2000 and later
- Integrated user-account support via Windows 2000 and later
- Kerberos support
- Automated task management
- Alert management
- Sophisticated replication support
- Query and index optimization wizards
- Database management wizards
- Full-text search capabilities
- Object Linking and Embedding Database (OLE-DB), Open Database Connectivity (ODBC), SQL Data Management Objects (SQL-DMO), and Database Library (DB-Library) API support
- Dynamic enlarging and shrinking of databases
- Full and differential database backups
- Graphical query tools
- Improved graphical import and export capabilities
- Graphical management tools using Microsoft Management Console (MMC) snap-ins

Although SQL Server 2000 can run on both the Windows 9x (and Millennium) and Windows server platforms, there are some differences between the two types of installations, as shown in Table 1.2.

<table>
<thead>
<tr>
<th>Difference</th>
<th>Windows 9x</th>
<th>Windows Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL engine</td>
<td>Runs as an application</td>
<td>Runs as a service</td>
</tr>
<tr>
<td>Integrated security</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Automated alerts</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum users (recommended)</td>
<td>Five</td>
<td>Limited by hardware</td>
</tr>
<tr>
<td>Performance Monitor</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Tasks of a SQL Server Developer**

A SQL Server developer is responsible for designing, programming, and populating the database. Because the focus of this book is the administration of SQL Server rather than the development
of SQL Server databases, the duties of the developer are not covered in detail. The developer’s responsibilities can be summarized as follows:

- Analyze the business situation to see what type of system is required. Is a new system or an upgrade needed? Is it a small company at one location or a big corporation with multiple locations?
- Design the database, tables, and all objects. In the design process, the developer identifies objects and relationships and how they all fit together as logical entities; the logical entities are then translated into physical tables (normalized). The developer must then plan for the application design, including reports and queries as well as other pieces, such as access to web pages.
- Design the security for SQL Server and for individual databases. Implementing security is covered in Chapter 5.
- Design any data-replication scenarios. Replication, including scenarios and implementation, is covered in Chapter 9.
- Program the database, tables, and all objects. This involves working with T-SQL.
- Program the initial security of the database, possibly planning Windows Active Directory groups to help ease ongoing SQL Server administration.
- Design the user interface, reports, and update screens. This is the front end of the system and will probably have the most impact on the users.
- Test the design, interface, reports, and update screens.
- Populate the database with live data from legacy systems and prepare the database to receive new data.
- Hand the project with appropriate documentation and training over to the administrator.

### Tasks of a SQL Server Administrator

A SQL Server administrator is usually responsible for the day-to-day administration of the database. The administrator takes over where the programmer left off.

**Note**

The tasks of the SQL Server administrator are the focus of this book. Microsoft has two separate SQL tests: one designed for the administrator (the test that is the focus of this book) and one designed for T-SQL programmers (the implementation test).

In many companies, the line between administrator and developer may become quite blurred because the same person may be doing tasks related to both roles.

The administrator’s duties can be summarized as follows:

- Install and configure SQL Server 2000 (see Chapter 2).
Tasks of a SQL Server Administrator

- Plan and create databases (see Chapter 4).
- Back up the databases (see Chapter 6).
- Restore the databases when necessary (see Chapter 6).
- Set up and manage users for SQL Server (see Chapter 5).
- Manage security for new users and existing users (see Chapter 5).
- Import and export data (see Chapter 7).
- Set up and manage tasks, alerts, and operators (see Chapter 8).
- Manage the replication environment (see Chapter 9).
- Tune the SQL Server system for the optimal performance (see Chapter 10).
- Troubleshoot any SQL Server problems (see Chapter 10).

Real World Scenario

A Real-World Microsoft SQL Server and Microsoft .NET Solution

One of the world’s largest medical centers recently took on a project to improve the health care it provides. Those involved not only wanted to improve the quality of the health care, they also wanted to lower the costs. The solution they came up with is an exciting new area of the medical and computer industries called telemedicine.

To begin the process, a consulting firm was hired to refine and implement the first of many health care improvements. The first area they decided to tackle was diabetes and hyperlipidemia (high cholesterol). They proposed, refined, and implemented the following solution: Patients were given a Personal Digital Assistant (PDA) telephone running Microsoft Windows CE (Phone Edition). A custom application written in Windows CE .NET was created to accumulate daily blood test results that the patients perform at home. This includes blood glucose levels as well as lipid levels. Other patient-related information was also taken. The program contains alert values and information for the patients’ use. If a patient inputs a blood glucose that is well outside their normal range and that falls into a danger area, the patient will be alerted on their PDA as they enter their values.

Once the data is entered into the PDA, it is stored to Microsoft SQL Server CE. Routinely, the data is transmitted wirelessly through a virtual private network (VPN) to a host server running Microsoft SQL Server Enterprise. This first SQL Server machine lives outside the hospital’s firewall system and is used primarily as a store and forward database. Once the data is collected at the SQL Server machine outside the firewall, a .NET Web Service is started to move the data to a SQL Server machine inside the firewall. The data is transmitted through the firewall using an XML format. When the data is successfully stored behind the firewall, a response is returned to the SQL Server machine outside the firewall. When the server outside the firewall received the response, it then moves the data just transmitted to its archives. If the data on the inside of the firewall is above/below alert levels, the patient’s primary care provider is notified via SQL Server email.
Much of the patient-collected data is useful to hospital researchers and needs to be incorporated into their existing legacy database system. Several solutions were proposed, including the use of Data Transformation Services (DTS) to move the data. In the end, DTS was abandoned in favor of a solution that was already implemented in other hospital systems.

The final piece is an ASP.NET web interface to the SQL Server data on the inside of the firewall. Using XML, XSLT (Extensible Stylesheet Language Transformations), and SQL Server’s support of XML through the SQLXML service pack, an administrative interface was built. This allowed the doctors and other care providers quick and easy access to patient-gathered data.

This solution should save the hospital money and provide better healthcare. The better care is provided because the PDA application will notify the patients that it is time to perform their tests. The data stored in the SQL Server CE database includes a custom test schedule. The data stored in the database also includes the customizable alert values to inform the patient that their blood glucose is outside the normal range and action needs to be taken (e.g., drink some orange juice). The hospital will save money because patients no longer need to come to the hospital for weekly consults just to drop off their data (formerly taken on a xeroxed sheet of paper). The doctors are freed up and can carry larger caseloads. Patients are monitored remotely through the web interface. If flare-ups occur, it is a simple matter for the care provider to make a phone call and schedule an appointment.

Telemedicine is an incredibly exciting new area of technology, and hospitals and insurers alike will benefit greatly from its use. Microsoft SQL Server (in all of its flavors) will be a cornerstone that great applications will be built upon.

Summary

A client/server database is an application that is divided into logical parts that run on a server and a client. The server side provides security, fault-tolerance, improved performance, concurrency control, and reliable backups. The client side provides the user interface. A relational database is composed of tables that contain related data, and the process of breaking a database into related tables is normalization.

Microsoft’s SQL Server is a client/server-based relational database management system (RDBMS) that uses T-SQL as its dialect of Structured Query Language (SQL), and its ever increasing popularity makes learning SQL Server 2000 a wise career decision. SQL Server developers are responsible for designing and implementing the databases. Designing a good database starts with understanding the client’s requirements for the database. SQL Server administrators are responsible for the day-to-day tasks of maintaining and managing the databases. SQL Server administration involves backing up databases and restoring them when necessary, setting up and managing users, managing database security, managing the replication environment, tuning the database system, and troubleshooting any problems that arise.
Key Terms

Before you take the exam, be certain you are familiar with the following terms:

client/server          normalizing
data warehousing       OLAP (Online Analytical Processing)
database objects       OLTP (Online Transaction Processing)
DSS (decision support system)          page
flat-file database      relational database
Review Questions

1. Which of the following is a database object? (Choose all that apply.)
   A. Table
   B. Index
   C. Rule
   D. Default

2. You are the sales department manager. Your main application runs on SQL Server. Because your department has doubled in size in the last year, you purchased a new computer to serve as your SQL Server computer. The network administrator, SQL administrator, and SQL programmer came in last weekend to move your application to the new server. You want to make sure the database is still getting backed up. Who would you ask first?
   A. The database administrator
   B. The database developer
   C. Database users
   D. The Windows 2000 administrator

3. You are the sales department manager. Your main application runs on SQL Server. You know the SQL programmer who designed and wrote the application. You hire a new user and need to grant them the proper permissions to use your application. Who would you talk to first?
   A. The database administrator
   B. The database developer
   C. The database users
   D. The Windows NT administrator

4. You are the sales department manager. Your main application runs on SQL Server. You know the SQL programmer who designed and wrote the application. You add 25 new users to the application and the server gets much slower. Who would you talk to first about getting the server to run faster?
   A. The database administrator
   B. The database developer
   C. Database users
   D. The Windows NT administrator

5. You have an Excel spreadsheet you want to convert to a SQL Server database. You want to search for any white papers that might help you design a relational database. What term best describes breaking related information into tables?
   A. Fragmentation
   B. Database design
   C. Normalization
   D. Tabulating the data
6. On your workstation, you are running an application that uses a SQL Server database. You are given 256MB of RAM. Queries take a long time to run and you want to speed them up. Where should you install the RAM?
   A. The client.
   B. The server.
   C. Both the client and the server.
   D. Neither; a “middleware” application runs the query.

7. SQL Server is an example of what kind of database system?
   A. Flat-file
   B. 3D
   C. RDBMS
   D. Hierarchical

8. You are going to design a database that will hold your customer and sales data. Customers tend to order more than once from you. You should design your database so that your customers and their sales are held in separate ___________________.
   A. Fields
   B. Files
   C. Reports
   D. Tables

9. You have a sales force with various operating systems installed on their laptops. Your new sales force application requires that SQL Server 2000 Personal be installed on every laptop. You need to budget for new hardware if a laptop can’t handle a new operating system and SQL Server. You won’t need to upgrade the laptops running which operating systems? (Choose all that apply.)
   A. Windows for Workgroups
   B. Windows 9x
   C. Windows 2000 Professional
   D. Windows Millennium

10. You are in a bookstore browsing through programming books. You want to buy one that helps you program SQL Server. Which topic best suits your needs?
    A. DBMS
    B. T-SQL
    C. PL-SQL
    D. QUERY
11. One of your users asks what a view is. Which answer would you give?
   A. Precompiled code that stores data
   B. A stored query that operates like a virtual table
   C. A method for organizing like data
   D. A way of entering default values

12. You have 50 users who telecommute from home and need access to your SQL Server computer. By corporate policy, they can use only Internet Explorer on the corporate network. Which version of SQL Server would you install for those users?
   A. The Personal version on every user’s computer
   B. The Personal version on the server
   C. The Standard version on the server
   D. The Enterprise version on the server

13. Your programmers are getting ready to code an application for SQL Server 2000. You buy all of them the Developer version. Some of the programmers work from home and have older computers. Which operating system will you not need to upgrade to run the Developer version?
   A. Windows 95
   B. Windows 98
   C. Windows Millennium
   D. Windows 2000 Professional

14. You win a copy of the Enterprise version of SQL Server. Which operating system can be on your test computers in order to install it? (Choose all that apply.)
   A. Windows 98
   B. Windows 2000 Professional
   C. Windows 2000 Server
   D. Windows 2000 Advanced Server

15. You read an article about tuning OLTP databases. You are not sure if you should implement any of the suggested changes. What type of database is typical for OLTP?
   A. Quick writes
   B. Quick reads
   C. Long writes
   D. Long reads
16. Your SQL programmer tells you that you have an OLAP-type application. Which of these best describes your database?
   A. Lots of new rows
   B. Lots of updates rows
   C. Lots of deleted rows
   D. Fewer long-running queries

17. You create some new tables and populate them with copies of your production data. You run reports on your new tables but your queries take a lot longer than on the original tables. What should you create to help speed up reads in the database?
   A. Stored procedure
   B. Index
   C. Default
   D. Primary key

18. You have salespeople who are running Windows CE devices. They each need a local copy of the corporate database on their local device. What version of SQL Server would you install on their device?
   A. CE
   B. Personal
   C. Standard
   D. Enterprise

19. You need to obtain copies of SQL Server Personal for all of your salespeople. How can you acquire the Personal version of SQL Server?
   A. It can be purchased separately.
   B. It comes with Office 2000.
   C. It comes with Windows 2000.
   D. Purchase and license SQL Server 2000 (Standard or Enterprise).

20. You are creating a data entry application. You wish to lower the amount of time it takes for users to enter repetitive data. What type of database object would you use?
   A. Index
   B. View
   C. Default
   D. Rule
Answers to Review Questions

1. A, B, C, D. Tables, indexes, rules, and defaults are all database objects. Other database objects include stored procedures, triggers, and views.

2. A. The database administrator is usually in charge of database backups, so they would logically be the first person you would ask about current database backups.

3. A. Although the SQL programmer who wrote the initial database would generally have information about the application, the database administrator is usually in charge of ongoing database security, so they would be the one to talk to about a new user.

4. A. The database administrator is usually in charge of SQL Server optimization. Although it could be a programming or operating system problem as well, the database administrator is the best place to start.

5. C. Normalization is the process of designing relational tables and would be a good keyword for a search on designing relational databases.

6. B. When a query is run in client/server computing, the server executes the query, so upgrading the server will probably give you the best performance increase.

7. C. SQL Server is a relational database management system (RDBMS). In an RDBMS, a database is not necessarily tied to a file; it is more of a logical concept based on a collection of related objects.

8. D. Related data in a relational database is organized into tables, so you would have one table for customers and one for sales. Because relational columns cannot contain multiple values, you would duplicate customer data for every second order, third order, and so on if you only used one table.


10. B. SQL Server 2000 uses Transact-SQL (T-SQL), which is a version of Structured Query Language (SQL). SQL is a language that was designed for database queries and that can be used to build databases and manage the security of the database engine.

11. B. A view is simply a stored query that operates like a virtual table.

12. D. The Enterprise version has HTTP support built into it so you can access your data via IIS and HTTP.


15. A. OLTP stands for Online Transaction Processing, which emphasizes lots of quick writes to the database.
16. D. OLAP stands for Online Analytical Processing, which emphasizes reads over writes. OLAP databases tend to be report oriented, which requires fewer, long-running queries, usually from only a few users.

17. B. Indexes help ensure quick reads of the database. Indexes organize the database based on the specified columns. If a query uses an index, the query time is usually sped up dramatically.

18. A. Only the CE version of SQL Server 2000 will run on Windows CE.

19. D. The Personal version of SQL Server can be installed (for no additional charge) if you have purchased and licensed the Standard or Enterprise version of SQL Server.

20. C. A column default is an optional database object that can be used to enter default data so users won’t have to when they add a new row to the database.