

CertifyExpress



Exam 70-228

Installing, Configuring, and Administering Microsoft SQL Server 2000 Enterprise Edition

Study Guide, written by Yu Chak Tin Michael

Abstract

This ExamNotes Study Guide intends to provide you with information to prepare for the Microsoft SQL Server 2000 70-228 Exam.

Before you start

This study guide provides you with information on the many different aspects of "SQL Server 2000". Before you proceed with this subject, please read through the study material for the following and make sure you are 100% comfortable with the SQL Server architecture:

- o SQL Server 7 Admin 70-028
- o SQL Server 7 Database Design 70-029

The reason you need to be familiar with SQL 7 is that the new version has many things similar to the version 7. Especially for the database design concepts, they are about the same.

Version 7 and Version 2000 have too many things in common. In fact, experience on 70-028 and 70-029 will be extremely helpful. This study guide will have its focus on the new features of SQL Server 2000.

Do NOT rely solely on this study notes for the exam. By all means read more than one book on the subject and make sure you understand the material well enough so that you could be ready for the questions. There is no quick way to succeed for this topic. Ideally you must work things out and gain experience before even trying to sign up for the exam.

Your Study Track for SQL Server 2000 Administration

1, Make sure you are comfortable with the concept of Relational Database System. SQL Server is a relational database system. Also know about SQL Structured Query Language, which is the language you will use to control many of the database operations.

For tutorials on RDBMS concepts as well as the SQL language, please visit the following sites:

http://www.spnc.demon.co.uk/ora_sql/sqlmain.htm

<http://w3.one.net/~jhoffman/sqltut.htm>

<http://freeskills.com>

http://www.ourfriends.net/pages/rdbms_b.htm

2, Make sure you know the basic concept of the Client Server Computing Model. The client presents and manipulates data on the desktop computer, while the server runs at the back end to store and retrieve protected data. SQL Server mostly works in a multi tier set up, but not as the front end though.

To gain a better understanding of Client Server Model, please visit:

http://webopedia.internet.com/TERM/c/client_server_architecture.html

<http://www.uth.tmc.edu/~atonnese/clinserv.html>

<http://www.faqs.org/faqs/client-server-faq/>

3, As of this writing many of the SQL Server 2000 resources are still in BETA testing stage. Due to the lack of SQL 2000 material in the market, you may want to start with those of SQL 7 first. Version 7 and Version 2000 have too many things in common. This study guide will have its focus on the new features of version 2000 of SQL Server

The scenario types of questions in the Transcender SQL 7 Practice tests (www.transcender.com) are extremely helpful. For SQL Server Admin, pay attention to questions related to Index, Replication, Database Structure, Profile Trace, DTS and Utilities (BCP, DBCC...etc). You will see similar stuff in the 2000 test.

Additional Suggested Readings from bookstore:

[Microsoft Sql Server 2000 Administrator's Companion](#) by Marcilina Garcia(Editor), et al (Hardcover - October 2000)

[MCSE : SQL Server 7 Administration: Training Guide : Exam : 70-028 \(MCSE](#)
by Christoph Wille, et al. Textbook Binding (May 1999)

[Deploying Microsoft Sql Server 7.0 : Notes from the Field](#)
by Microsoft Corporation(Editor). Paperback (July 1999)

[Inside Microsoft SQL Server 7.0](#)
by Ron Soukup, Kalen Delaney. Hardcover (May 15, 1999)

4, Know the permissions and security measures in Windows NT and Win2K. Here is a list of recommended readings for this topic:

[Microsoft Windows 2000 Security Handbook](#) by Jeff Schmidt, et al (Paperback)

[Configuring Windows 2000 Server Security](#)

by Thomas W. Shinder(Contributor), et al. Paperback (November 1999)

[NT 4 Network Security](#)

by Matthew Strebe, et al. Paperback (March 15, 1999)

5, Learn the new features offered by SQL Server 2000. Please visit the official SQL Server site for a list of new features:

<http://www.microsoft.com/sql/>

The best way to learn is to actually play with SQL Server 2000. You may obtain an evaluation edition from

<http://www.microsoft.com/sql/productinfo/evaluate.htm>

You should also download the set of reference materials for SQL Server 2000 from the following location:

<http://www.microsoft.com/sql/productinfo/evaluate.htm#Evaluation%20Materials>

6, Optionally, you may want to know XML, as SQL Server 2000 has integrated into it the support of XML. Short for Extensible Markup Language, XML is a specification developed by the W3C. Being the pared-down version of SGML, it is designed especially to allow designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

The relational database engine of SQL Server 2000 can return data as XML documents. Also, we can use XML to insert, update, and delete values in the database.

To learn more about XML, try the following sites:

<http://webdevelopment.developersnetwork.com/xml/>

<http://www.w3.org/XML/>

<http://www.csclub.uwaterloo.ca/u/relander/XML/>

Specifications

Operating system and software	Windows NT Server 4.0 Enterprise Edition with Service Pack 5 (SP5) or later Windows NT Server 4.0 with SP5 or later Windows NT Workstation 4.0 with SP5 or later Windows 2000 Internet Explorer 5.0 or later
Upgrade from SQL Server 6.5	On the same computer, you must have applied SQL Server 6.5 Service Pack 5 (SP5) or later. On a different computer, you must have applied SQL Server 6.5 Service Pack 3 (SP3) or later.
Upgrade from SQL Server 7.0	Any Service Pack level is ok.
Network protocols	Named Pipes at the default pipe of <u>\\.\pipe\sql\query</u> .

Now watch for the RAM and Hard drive space requirements:

Memory:

- o Enterprise Edition: 64 MB; 128 MB recommended.
- o Standard Edition: 64 MB.
- o Developer Edition: 64 MB.
- o Personal Edition: 64 MB for Windows 2000; 32 MB for other operating systems.
- o Desktop Engine: 64 MB for Windows 2000; 32 MB for other operating systems.

Hard-Disk Space:

- o Enterprise, Standard, Evaluation, Developer and Personal Editions require 95–270 MB for server; 250 MB for typical installation.
- o 50 MB for minimum installation of Analysis Services; 130 MB for typical installation.
- o 80 MB for Microsoft English Query.
- o Desktop Engine requires 44 MB of available hard-disk space.

Regarding Networking Support, for Microsoft Windows 95, Windows 98, Windows Me, Windows NT 4.0, and Windows 2000, since they already have built-in network software, nothing special need to be done. However, additional network software is required if you are using Banyan VINES or AppleTalk ADSP. Also note that Novell NetWare IPX/SPX client support is provided by the NWLink protocol of Windows-based networking. For non-MS clients, UNIX, Apple Macintosh, and OS/2 require Open Database Connectivity client software from a third-party vendor.

Editions

Keep in mind that SQL Server 2000 Personal Edition is offered for desktop and mobile use, and so it does not contain the full functionality of Standard Edition.

For detailed comparison between the different editions of SQL Server 2000, please visit:

<http://www.microsoft.com/sql/productinfo/sqlcompdata.htm>

Collations

With SQL Server 7.0, during install you must specify a default code page and sort order. All databases will then be locked into that particular code page and sort order. In SQL Server 2000, collation acts as a collection of the sort order for Unicode data types, the sort order for non-Unicode character data types, and the code page that is used to store non-Unicode character data types. You can specify collations at the database or column level, in addition to the default collation specified during installation.

Installation

You may freshly install SQL Server 2000, or to upgrade from SQL Server 6.5 or SQL Server 7.0 to SQL Server 2000. In any case, the computer must meet the hardware and software requirements mentioned above.

When doing an upgrade, you overwrite an installation of SQL Server 7.0 if SQL Server 7.0 is detected as an existing installation when you run Setup. All data stored in SQL Server 7.0 databases is preserved. However, SQL Server 7.0 profiler traces and registered servers are not upgraded even when the SQL Server 7.0 tools are upgraded to SQL Server 2000. Also, information models that were installed with Microsoft Repository 2.0 are not upgraded too.

If later you want to undo the upgrade, the only way to restore the SQL Server 7.0 is to first uninstall SQL Server 2000, perform a complete reinstall of SQL Server 7.0 files, and then restore the backed-up SQL Server 7.0 databases.

After the upgrade is done, it is recommended that you perform the following to enhance the performance of SQL Server 2000:

- Repopulate Full-Text Catalogs, as the upgrade process has marked your databases as full-text disabled. This process can be very time-consuming, that is why it is not automatically done during set up.
- Update Statistics, as using SQL Server 7.0 statistics with SQL Server 2000 may result in poor query performance. You do the update via the sp_updatestats stored procedure.

You can convert data from SQL Server 6.5 to the formats for SQL Server 2000 using the SQL Server Upgrade Wizard. To run the Upgrade Wizard, you must have a default instance of SQL Server 2000 installed on your computer first. The wizard will upgrade:

- all databases
- all catalog data, objects, and user data
- replication settings
- SQL Executive settings
- most of the SQL Server 6.5 configuration options

Note that the Upgrade Wizard does not support consolidation of databases from multiple SQL Server 6.5 installations, meaning you must consolidate all of the SQL Server 6.5 databases onto one server first by yourself. Also, the Upgrade Wizard does not remove SQL Server 6.5 from your computer, meaning there will be two separate installations of SQL Server exist afterwards.

During the upgrade, errors will be noted in the output logs of the SQL Server Upgrade Wizard, except for the intentional differences in objects causing any conflict.

The transfer of objects and data by the SQL Server Upgrade Wizard is very reliable, as there are verification measures to:

- o Validate successful object data transfer
- o Exhaustive data integrity verification

The order of upgrade is roughly the same for direct pipeline and tape drive upgrade, except that for tape drive upgrade, data is exported to the tape drive after shutting down SQL Server 6.5 and before starting SQL Server 2000, while with direct pipeline, the export and import steps are done simultaneously.

When you plan to install a SQL Server 2000 failover cluster, you must ensure that the cluster operating system is installed properly to support failover clustering. You should also consider whether the SQL Server tools, features, and components you run are supported with failover clustering.

When you are upgrading to a SQL Server 2000 failover cluster, you need to use the Cluster Wizard in SQL Server 6.5 or SQL Server 7.0 to uncluster any existing SQL Server 6.5 or SQL Server 7.0 clustered instances first before upgrading. Also, SQL Server 6.5 or SQL Server 7.0 failover clusters cannot exist on the same computer as a SQL Server 2000 failover cluster, and that you cannot run the Cluster Wizard in SQL Server 6.5 or SQL Server 7.0 after SQL Server 2000 has been installed. For SQL Server 2000, you must use a domain account for the services that must be an administrator on all computers in the cluster.

Copy Database Wizard and DTS

With the Copy Database Wizard, you can move or copy a database and associated meta data from SQL Server 7.0 to SQL Server 2000 without the need to shut down any servers in the process. We refer this to online database upgrade, meaning there is no downtime for servers ever during the process. Of course, this can also be used for non-upgrade copying or moving operations. Also, the copy/move process can support local or remote options. Clustered environment is supported, however, for the upgrade feature, SQL Server 6.5 database is not supported.

Under the hood the Database Copy Wizard is based on detach and attach functionality that allows user databases to be moved or copied from a source to a destination server. It uses Data Transformation Services DTS package to perform the actual move or copy operation. You may schedule the package to run at anytime you prefer.

At the start of a database move or copy process, you should ensure that one administrator is having exclusive use of all files to prevent any changes to the file set during the process. Actually, you need 2 connections to copy database files: sysadmin privileges on all the SQL Servers and administrator privileges on the server and the network. To prevent any chance of data corruption, the databases being copied must be in read-only condition. Also keep in mind that the Copy Database Wizard cannot be used when the database has identical name on both the source and the destination servers, meaning you must manually resolve any potential naming conflicts in advance.

With the DTS Import/Export Wizard, you can quickly and easily move data between heterogeneous data sources in ad-hoc data transfer or database alteration. You can also build packages to be stored in SQL Server Meta Data Services for regular automated use. The DTS Designer is a graphical tool that allows you to import, export, and transform heterogeneous data between one or more databases and SQL Server. You should use DTS Designer when you want your package workflow to contain sophisticated logic, or when you have an existing package that requires editing.

Security and Auditing

Regarding security, you should know the difference between a trusted connection and an untrusted connection. An untrusted user is one who attempts a remote login to the local server. In a network connecting the client and the server, if it is a closed one supported by components with authentication, the connection in between is trusted.

In Windows 2000, SQL Server 2000 uses Kerberos to support mutual authentication between the client and the server. It has the ability to pass security credentials of a client between computers to allow work on a remote server to proceed using the credentials of the impersonated client.

Users trying to connect to SQL Server must identify themselves with a specific login ID. To integrate NT security, the Sp_grantlogin stored procedure authorizes a Windows network account to be used as a SQL Server login account. This effectively allows for connecting to SQL Server using Windows Authentication. Once logged on, the users can only see the tables and views that they are authorized to see, and can only execute procedures and functions they are authorized to execute.

The syspermissions table contains information about permissions granted and denied to users, groups, and roles in the database. This table is stored in each database. For a user to gain access to an object, he/she must be the object owner, or be granted access to the object, or be in the appropriate database role membership.

With SQL Server 2000, an audit trail of activity that has occurred on SQL Server can be kept. This is referred to as C2 auditing capability. In SQL Server 2000, when enabled, auditing is always on and is not in the context of any particular user, as it must be running prior to user logon. Every modification of an audit is itself an auditable action, however, since NT's Service Control Manager does not notify SQL Server who started a service, you should consider to audit service control actions in Windows NT as well. Keep in mind that only members of the sysadmin fixed server role are able to control and modify auditing.

You set up C2 auditing and log file rollover with the C2 audit trace option, which is available as an advanced configuration setting. Here is a list of auditable events:

- o End User Activity
- o DBA Activity
- o Security Events
- o Utility Events

Each audit event record contains at least the start time of the event, the name of the user who caused the event to occur, SID of the user who caused the event to occur, the Event Class and Subclass, success or failure of the event, server name of the SQL Server, the request client computer name, the name of the application, and the corresponding server process id of the user's SQL Server connection.

When auditing is enabled, the trace files roll over automatically and create a new file, then close the old file handle when each reaches 200 megabytes in size. When no more room is available to write audit logs in the \mssql\data directory, SQL Server will shut down. If there is an unexpected system crash, the maximum number of audit records that could be lost is only 128 kilobytes worth of data.

To open a trace file, you can:

- o Use SQL Profiler to import the trace files into a SQL Server table
- o Use FiletoTable.exe to load the trace file to a database table and then analyze the data through Transact-SQL commands.

Replication

In SQL Server 2000, the 2 major types of replication are:

Transactional Replication:

- o provides loose consistency between a publisher and subscriber
- o suitable for application that demands not just identical data at different sites, but the necessity to mirror each and every data update, addition or deletion

- allows you to reliably track every change to your publication in close to real-time

Merge Replication:

- allows many or all subscribers as well as the publisher to make updates to replicated data
- allows subscribers to modify or add data while on the road, then connect to the network to merge their modified data with the original copy
- suitable for environment where subscribers are frequently disconnected from their network

To be ready for replication, some setup need to be done in the server. To set up your server as a publisher and distributor, you need to:

- Start the "Configure Publishing and Distribution" Wizard in Enterprise Manager.
- Configure your server to be its' own Distributor.
- Configure SQL Server Agent to start automatically.

Performance

SQL Server Query Analyzer is a GUI tool tightly integrated with the Index Tuning Wizard and SQL Server Profiler that enables you to write queries, execute multiple queries simultaneously, view results, analyze the query plan, and receive assistance to improve the query performance. Database Maintenance Plan Wizard helps setting up the core maintenance tasks necessary to ensure your database performs well, that it is regularly backed up and is checked for inconsistencies.

New in SQL Server 2000 is indexed view. In SQL Server 2000, a view that has a unique clustered index is referred to as an indexed view. You may actually create a unique clustered index as well as non-clustered indexes on a view to improve data access performance on complex queries. However, keep in mind that not all queries will benefit from indexed views. Most importantly, if the indexed views are not used, there is no benefit.

Benefits of indexed view can be realized in the following situations:

- Joins and aggregations of large tables
- Repeated patterns of queries
- Repeated aggregations on the same or overlapping sets of columns

- Repeated joins of the same tables on the same keys

The query optimizer uses indexed views only for queries with nontrivial cost in order to avoid the high cost of trying to match various indexed views during the query optimization. Precisely, indexed views are rarely used in queries with a cost of less than 1. In general, you should consider implementing indexed views for the following types of applications:

- Decision support workloads
- Data marts
- Online analytical processing stores and sources
- Data mining workloads

You should consider using indexed view in schema that involves any subset of tables referenced in the query, any subset of the conditions in the query for that subset of tables, grouping columns or aggregate functions.

SQL Server query optimizer automatically determines when an indexed view can be used for a given query execution, meaning you do not need to reference the indexed view directly in the query for the optimizer to use it in the query execution plan. In fact, Query optimizer considers many conditions to determine how an indexed view cover a query, such as: Tables in the query FROM clause must be a superset of the tables in the indexed view FROM clause; Query's join conditions must be a superset of the join conditions in the view; Aggregate columns in the query must be a subset of the aggregate columns in the view; All expressions in the query select list must be derivable from the view select list, or from the tables not included in the view definition; The query search condition predicates must be a superset of the search condition predicates in the view definition; All columns in the query search condition predicates that belong to tables in the view definition must appear in at least one of the following: same predicate in the view definition, GROUP BY list, view select list (if there is no GROUP BY)...etc.

With the NOEXPAND option you can force the query optimizer to treat the view like an ordinary table with a clustered index. Or you can explicitly exclude indexed views from consideration by using the EXPAND VIEWS option with the query.

For the best result, indexes on tables and indexed views should be designed concurrently to avoid redundant recommendations that incur high storage and maintenance overhead. Also, when designing the indexed view, follow these guidelines if possible:

- Design indexed views that can be used by multiple queries.
- Keep the index compact by using the fewest number of columns and bytes as possible.

- o Estimate the size of the resulting indexed view in advance, as a large size may not provide any significant performance gains at all.
- o Use multiple smaller indexed views when possible.

You may use Index Tuning Wizard for creating indexed view, as it will make recommendation for you. For maintenance, SQL Server automatically maintains indexed views similar to any other index. However, if the indexed view references several tables, updating any of them may require updating the indexed view, which is expensive relatively. You need to take this into account before considering the use of indexed view, if the view will reference multiple sources.

Datatypes, Functions, and Triggers

SQL Server 2000 introduces the 64-bit integer (bigint), variant (sql_variant), and table data types. The bigint data type is an integer that supports data from -2⁶³ (-9,223,372,036,854,775,808) through 2⁶³-1 (9,223,372,036,854,775,807) with a storage size of 8 bytes. Note that functions will return bigint only if the parameter expression is a bigint data type, and that SQL Server will not automatically promote other integer data types to bigint. Sql_variant data type stores values of various supported data types except text, ntext, image, timestamp, and sql_ variant. It operates like the variant data type in Visual Basic in that it allows a single column, parameter, or variable to store data values of different data types. The new table data type is a local variable for temporarily storing a rowset. It can be used in place of temporary tables stored in the tempdb database. Keep in mind that table data type is managed in memory only, it provides performance benefits over other disks based alternatives.

User-defined functions are similar to stored procedures. They accept zero or more input parameters, and can return a scalar data type such as int, decimal, varchar, or sql_variant as well as the table data type. However, you must specify the RETURNS value and to terminate with the RETURN statement.

The new trigger type supported is the INSTEAD OF trigger, which has the ability to be specified on views in place of the triggering action, and the AFTER trigger, which is now the default trigger executed after the statement that triggered it completes for table (and only table).

