Exam
Manual

LearnSmart

Microsoft (70-432) SQL Server 2008 Implementation and Maintenance

LearnSmart's SQL Server 2008 Implementation and Maintenance exam manual is designed to meet the needs of IT professionals who are responsible for database administration. In this manual, candidates who wish to become MCTS certified will become familiar with an array of topics, including:

- Installing and Configuring SQL Server 2008
- Maintaining SQL Server Instances
- Managing SQL Server Security
- Maintaining a SQL Server Database
- And more!

Give yourself the competitive edge necessary to further your career as an IT professional and purchase this exam manual today!

SQL Server 2008 Implementation and Maintenance (70-432) LearnSmart Exam Manual

Product ID: 012335
Production Date: July 13, 2011

Warning and Disclaimer

Every effort has been made to make this document as complete and as accurate as possible, but no warranty or fitness is implied. The publisher and authors assume no responsibility for errors or omissions. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this document.

Volume, Corporate, and Educational Sales

Favorable discounts are offered on all products when ordered in quantity. For more information, please contact us directly:

1-800-418-6789
solutions@learnsmartsystems.com

International Contact Information

International: +1 (813) 769-0920

United Kingdom: (0) 20 8816 8036

Table of Contents

Abstract

This LearnSmart exam manual will help you prepare for the SQL Server 2008 administration as tested in Microsoft Exam 70-432. This manual provides practical examples, test tips, and guidance on Microsoft best practices, which pervade the questions in Exam 70-432.

What to Know

Students should study all facets of installing SQL Server, installing SQL Server instances, and adjusting SQL installations via SQL Server Management Studio, SQL Powershell, and SQL Command Line. Students should study SQL query syntax and understand the T-SQL commands related to installation, security, and management of SQL Server.

Tips

It is important to ensure that you have hands-on experience with SQL Server 2008 and can attempt the syntax of queries demonstrated in this manual and the configuration examples. The availability of so many virtual server/PC type products today and Microsoft's 180-day free trial of SQL Server 2008 should allow plenty of chances to work with the product prior to taking the exam.

Domain 1 – Installing and Configuring SQL Server 2008

The first objective when preparing for Microsoft Exam 70-432 is installing Microsoft SQL Server 2008 and obtaining a strong understanding of its various options. As a database administrator knowing how to install Microsoft SQL Server 2008 accurately in production scenarios will give you the knowledge to answer questions based on this section of Microsoft Exam 70-432.

Hardware and Software Requirements for Installing Microsoft SQL Server 2008

Microsoft SQL Server 2008 is available in multiple versions. For test preparation purposes we will focus on the Enterprise 2008 x64 and Standard x32 versions. Each version of Microsoft SQL Server 2008 has unique features but for test purposes the installation questions apply to all versions except for Microsoft SQL Server 2008 Express. This section will detail processor types for Microsoft SQL Server, hardware requirements, and software requirements. Installation also covers data file locations, default paths for Microsoft SQL Server, and security during installation.

Microsoft SQL Server 2008 comes in multiple processor versions, and the following definitions are applicable for understanding the processor versions for installation.

Processor and System Architecture Type	Nickname or Common Reference	Example Operating System	Microsoft SQL Versions for this Architecture
x86	32-bit, a 32-bit processor (e.g., a Pentium IV or Pentium III processor)	Microsoft Windows Enterprise 2003 R2 x86 Microsoft Windows 2008 R2 Enterprise x86	Microsoft SQL Server 2008 Enterprise 32-bit Microsoft SQL Server 2008 Standard 32-bit
x64	64-bit, a 64-bit processor (e.g., a AMD Opteron, or Intel Xeon with EM64T support)	Microsoft Windows Server 2003 Enterprise R2 x64 Microsoft Windows 7 Ultimate x64	Microsoft SQL Server 2008 Enterprise (64-bit) x64 Microsoft SQL Server 2008 Standard (64-bit) x64
Itanium	IA64, a 64-bit custom processor built by Intel, which requires a custom operating system and versions of SQL Server. Normally used for high-end installations	Microsoft Windows 2008 Server Enterprise R2 Itanium IA 64 Microsoft Windows 2003 R2 64-bit Itanium Datacenter	Microsoft SQL Server 2008 Enterprise (64-bit) IA64

Table 1: SQL Server and Supported Architectures

Test Tip: Always ensure a match between the Microsoft SQL Server 2008 version and the server processor architecture. You are asked to install a 64-bit version of Microsoft SQL Server on a Pentium IV. Is this configuration possible?

The required SQL Server 2008 Hardware is listed in the table below. The ability to quickly recall these parameters is an important part of preparing to install SQL Server and serving as a SQL Server Administrator. The software requirements are addressed and handled by the SQL Server installation wizard.

Required RAM	Required Processors	Required Disk Space
All versions of SQL Server 2008 require a minimum of 512 MB of RAM. However, SQL Server 2008 will not run very well on 512 MB of RAM. One exception is that SQL Server 2008 Express only requires 256 MB of RAM. 2GB of RAM is recommended.	32-bit versions of SQL Server require a 32-bit processor with 1 GHz or higher of CPU. 64-bit versions of SQL Server require a 64-bit processor with 1.4 GHz or higher.	SQL Server has different disk size options based on the options selected, but for testing purposes the amount of disk space required for the database engine, replication, and full text search is 280 MB. Other SQL Server options will increase the disk space. Analysis Services requires 90 MB of space. Integration Services requires 120 MB of space. Reporting Services requires 120 MB of space. SQL Books Online requires 240 MB of space. SQL Server manager and other client components require 850 MB of space.

Table 2: SQL Server Hardware Requirements

Installing SQL Server 2008

Installing Microsoft SQL Server 2008 is accomplished via the **setup.exe** file of the source binaries.

Installation will run a check for all required hardware and software and stores all events related to the setup in the following location on the server or workstation hard drive: **%programfiles%\Microsoft SQL Server\100\Setup Bootstrap\Log**. Each file name is named in the format Summary_engine_based_ YYYYMMDD_HHMMss.txt. This is the year, month, date, hour, minute, and second of the setup. Once installation is complete all of the setup activity is logged in the summary.txt file located in the above directory. For example, if you are troubleshooting why a certain SQL service or feature is not available (which the exam will question you on), the first place to start is the summary.txt file.

Test Tip: For any failures during Microsoft SQL Server setup, the summary text file is the place to start when troubleshooting. This is as easy as opening the file and conducting a search for the keyword "error." If the summary.txt is not listed as an option for troubleshooting the SQL Server installation, then check for Application Event Log entries, and don't forget the basics for verification – run the SQL Server Management Studio and also check if the binaries are installed.

This overview of a SQL Server installation will cover some of the required concepts for the exam and illustrate best practices, which Microsoft covers in the exam.

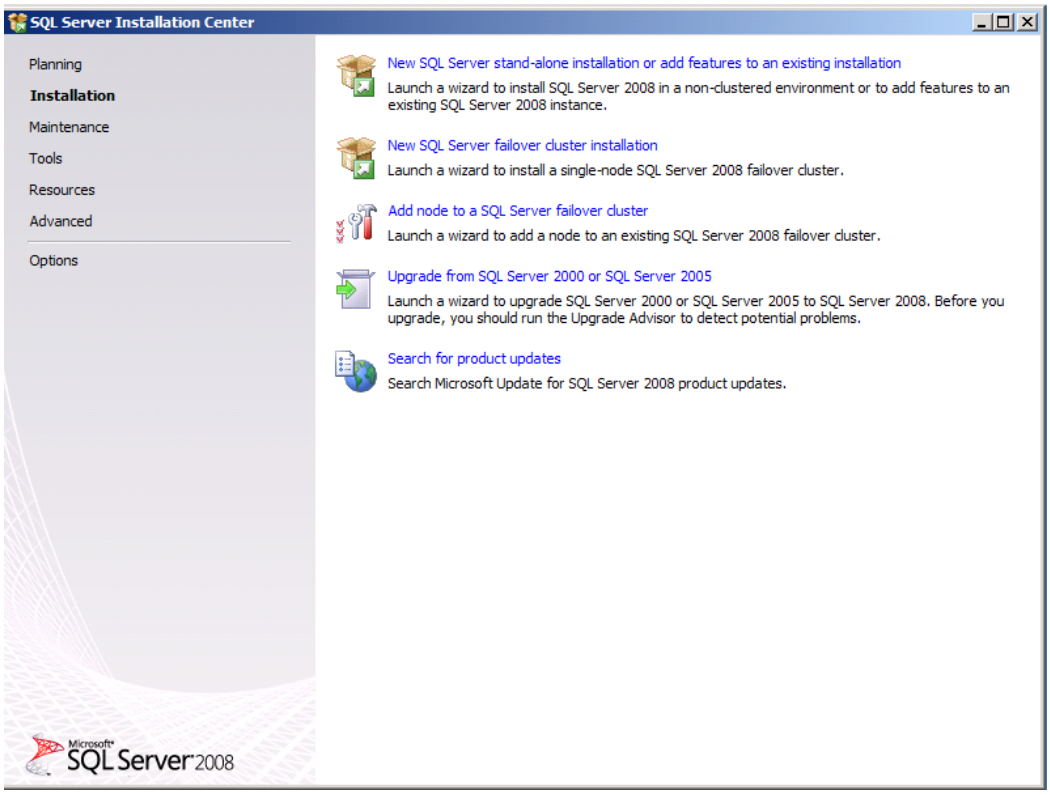


Figure 1: SQL Server Install Options

Figure 1 shows the options for the type of install. We will cover SQL Server failover clustering in Domain 8 in this Exam Manual. The first option will install a new default instance SQL Server, or add a new named instance of SQL Server. The upgrade option is to upgrade SQL Server 2000 or SQL Server 2005 to SQL Server 2008.

SQL Server Instances

One very important exam concept is SQL Server instances. SQL Server can be installed on the same physical hardware with multiple instances that are completely independent of each other. For example, you can install a default instance of SQL Server 2008 named "Server Name." On the same server you can install a new instance of SQL Server 2008 named "Server Name\Test01." This second instance of SQL Server 2008 is completely independent of the first instance for database locations, service accounts, and features. The only items shared are the core SQL Server binaries installed on the server. The second instance will browse for applications as a different SQL Server. A .NET application employing connection strings could connect to the default SQL Server for its core application databases and the SQL Server instance for its Reporting Server needs. A quick illustration is shown below.

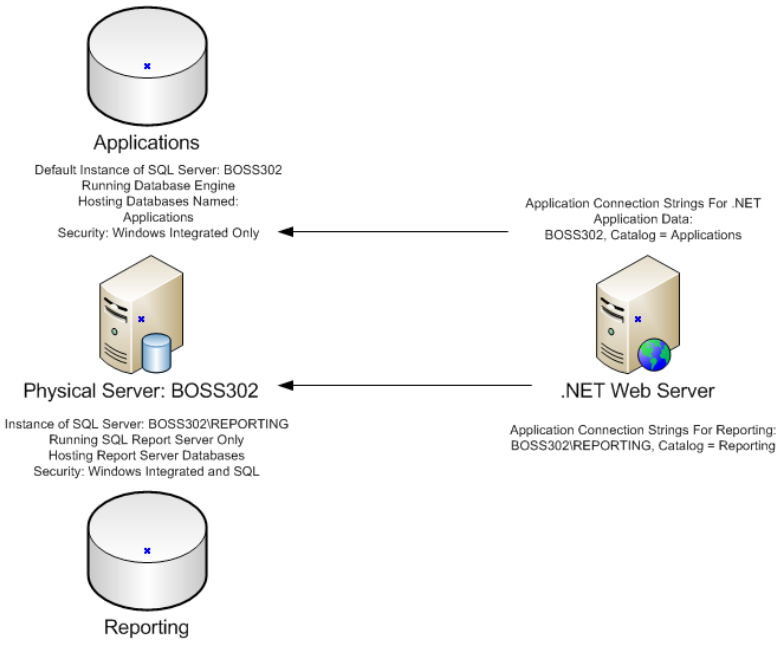


Figure 2: Practical Examples of SQL Server Instances in Use

Test Tip: SQL Server versions can support 8 instances, and SQL Server Enterprise can support 50. SQL Server instances provide an array of capabilities and are a valid answer to some exam questions. For example, many of the exam questions focus on upgrading from SQL 2005 to SQL 2008. In some cases the questions on the exam will have answers that use the least amount of hardware investment. Anytime that phrase is used you should use an SQL instance to answer the question. For example, if you are upgrading to SQL 2008 from SQL 2005 and require testing and other certification of existing databases but only have one server what should you do? You should install a named instance of SQL 2008 on the same server running SQL 2005. SQL 2005 and SQL 2008 can run on the same hard drives; however, since SQL 2005 is already running on the server you must use a named instance because a default instance (the SQL 2005 installation) is already running. SQL 2005 and SQL 2008 use different directories for the installation so they will not affect each other. This is a rapid way to use an instance to solve an upgrade issue with constrained hardware and allow detaching of databases running SQL 2005 and reattaching of the same databases in SQL 2008.

Selecting SQL Server Features

The SQL Server installation allows you to select multiple products. These products are described above in Table 1 in the disk space section. When installing SQL Server you are faced with this array of functionality and must understand what each item provides to perform well on the exam.

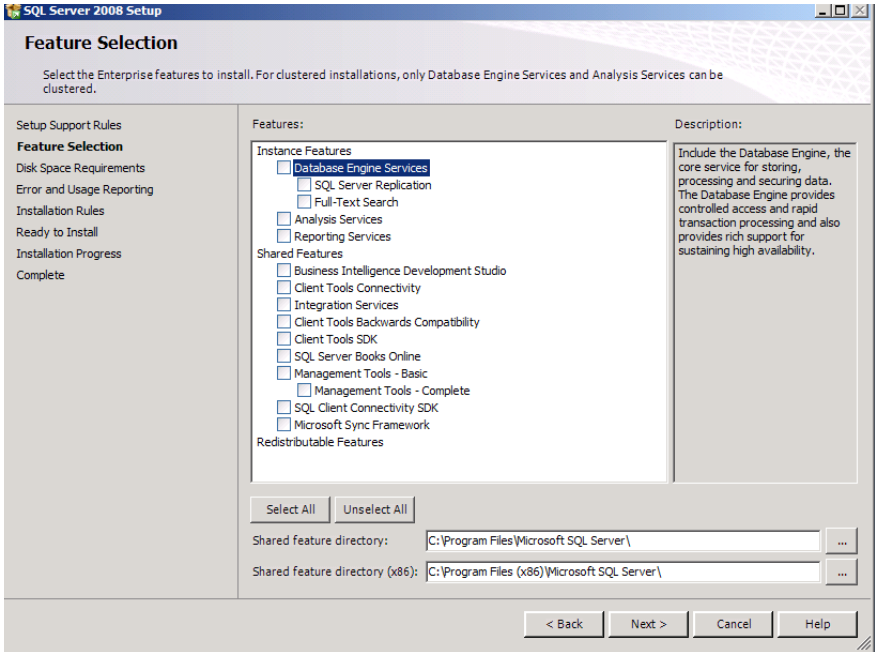


Figure 3: SQL Server features during install

The SQL Server Database Engine is the heart of the product and provides the location for storing and processing data securely. SQL Server Replication provides for fault tolerance support by allowing the mirroring and replication of SQL Server databases to other SQL Servers. Full-Text Search provides enhancements in the searching of text and content on SQL Servers. Analysis Services is the solution enabler for analytical processing of data. Reporting Services provides an out of the box reporting solution for the creation and management of data reports presented via the web. The SQL Server shared features provide all of the other tools that ensure SQL Server functionality. The important features to understand for the exam are the management tools, which provide the SQL Server Management Studio.

Test Tip: In normal production scenarios where fault tolerance and high availability is a consideration Microsoft does not recommend installing all SQL Server options on one server. This is also a performance consideration.

An SQL Server installation requires the selection of services needed for the business, and the selection of a service will prompt other configuration data as the installation wizard for SQL Server proceeds. If Reporting Services is selected, a service account and indication of the file locations for the reporting databases, transaction logs, and binaries will be required.

Test Tip: If users cannot connect to a properly installed SQL Server after installation and all security is correct and the SQL Server is running, make sure the SQL Server Browser is disabled by default. By enabling this SQL Service through the SQL Server Configuration manager, clients can connect to the SQL instance without using port numbers or named pipe names.

SQL Server Security Considerations During Install

The selection of SQL Server service accounts is required during installation. Service accounts are domain accounts or local server accounts with required administrative privileges to run the SQL Server services. Local computer accounts are also available for running the SQL Server services. Local system accounts are used only by system operations and are not available for an interactive login. The types of local system accounts available depend on if the SQL Server is a member of an Active Directory domain or a stand-alone. A stand-alone SQL Server is an installed SQL Server that is in a workgroup versus an Active Directory domain. System accounts do not require passwords as they are embedded system resources that require physical or RDP server access to configure. The recommendation (which will help on the test) is to create service accounts in advance of the SQL Server and make these accounts members of the Administrative group on the SQL Server. You can then use these accounts to run SQL Server services and ensure you have an audit trail of which accounts are assigned to what services in an Active Directory domain. An example service account name is domain name_svc_dbserver.

Test Tip: Never configure SQL Server services to use the Network Service built-in account in an Active Directory configuration. This account has rights to traverse domain servers and is considered a violation of the security best practice.

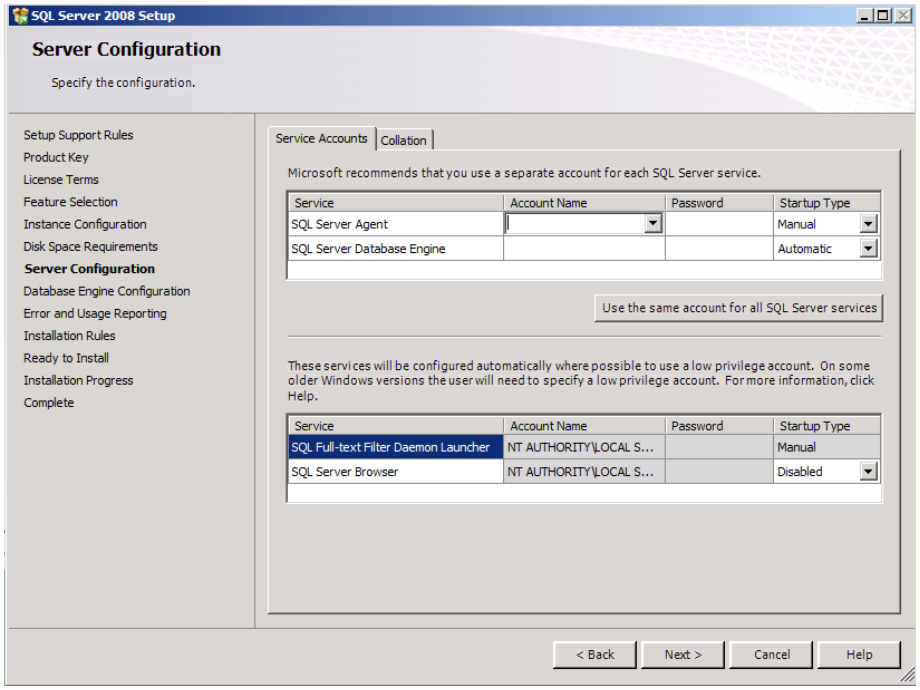


Figure 4: SQL Server Installation Dialogue for Selection of Service Accounts

Test Tip: The SQL Server Agent is disabled by default. It will not start automatically. We will address this in SQL Server Configuration later in this section.

Account provisioning is a critical area during SQL Server installation. SQL Server provides two authentication modes: Windows authentication mode and Mixed Mode authentication, which allows SQL Server to maintain accounts for SQL Server access. Today's best practice for security is not to configure any SQL Server in Mixed Mode, thus reducing the attack surface against SQL Server by not allowing authentication by the SQL Server itself. In some cases a legacy application or support consideration will require accounts in SQL Server. For the exam, keep in mind that using another instance of SQL Server to accommodate different security needs is the way to go. Account provisioning is the stage of the installation when you will decide who will administer the SQL Server. In SQL Server 2008 a decision must be made to add an administrator to SQL Server for management once the product is installed. This data should be prepared in advance of the installation. For example, a group in the Active Directory may contain designated SQL Server administrators or a local group when the Active Directory is not in use.

Test Tip: Make sure you completely understand the ramifications of using Mixed Mode SQL Security! In most cases – don't do it!

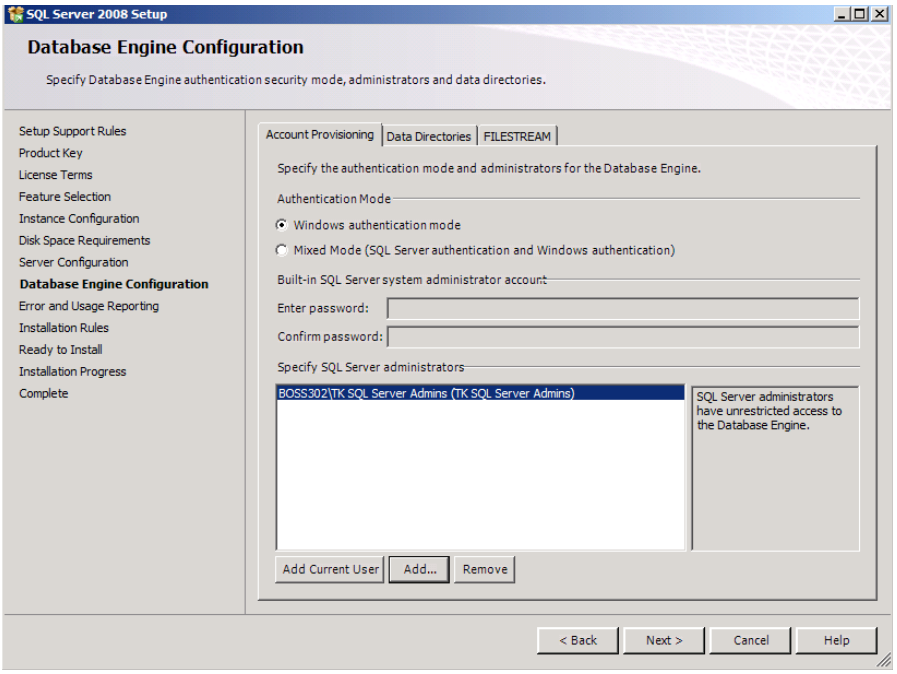


Figure 5: SQL Server Account Provisioning and Authentication Mode Choices

FILESTREAM Configuration

An important new feature in SQL Server 2008 is FILESTREAM support. FILESTREAM is a feature that supports the storage of large images and other BLOB type data over 1 MB. An example is scanned images, or JPEGs of receipts. This SQL option enhances fast read access for stored objects. This feature can be configured during installation or by using the SQL Server Configuration Manager. FILESTREAM enables SQL Server to quickly access objects stored in a server file system versus a database table. The data type is varbinary (max) when items are stored in file server storage. The configuration requires the name of a Windows share.

Test Tip: The following T-SQL statement enables the FILESTREAM attribute capability after the installation of SQL Server. The option 0 disables FILESTREAM, 1 enables FILESTREAM for T-SQL access, and 2 allows FILESTREAM access for Win32 applications.

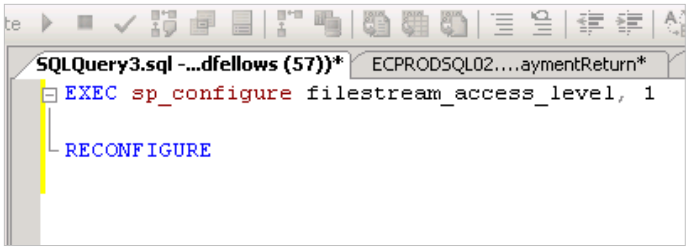


Figure 6: T-SQL Statement to complete FILESTREAM post-installation configuration

Data Directories in SQL Server Installation

The location of SQL databases is determined during this phase of installation. Best practices dictate that Temp DB and the Temp DB log run on LUNs or drive spindles separate from other databases, and in a RAID 1 configuration. User databases should exist on different LUNs rather than user database log files, and should also be on different drive spindles. User databases should run on RAID 1+0 volumes, and user database log files on RAID 1 volumes. Most disk locations should be configured and ready to use prior to installation.

In common SAN configurations each drive letter will present to the SQL installation as the location to achieve best practices. For example, the T (T for Transaction Logs!) drive the server has access to is a good location for transaction logs, configured as a RAID 1 volume. The D drive (D for Data!) is a good location for data, configured as a RAID 1+0 volume. This allows quick adjustment in the installation to the data locations and ensures that all databases added will end up configured with the right drive location.

In the high availability section of this manual these considerations control the success of cluster installations covered in the last section. This section of the installation allows configuration of the backup directory. This directory becomes important in the setup of management jobs if it is already configured. Best practice is for the backup location to exist on a protected SAN and to ensure the LUNs are not used in any SQL read or write operations with the stored databases.

Test Tip: Although a RAID 5 disk configuration does provide fault tolerance, the function of writing multiple bits across the array is a write inhibiter to SQL Server performance in high volume or high transaction per second configurations.

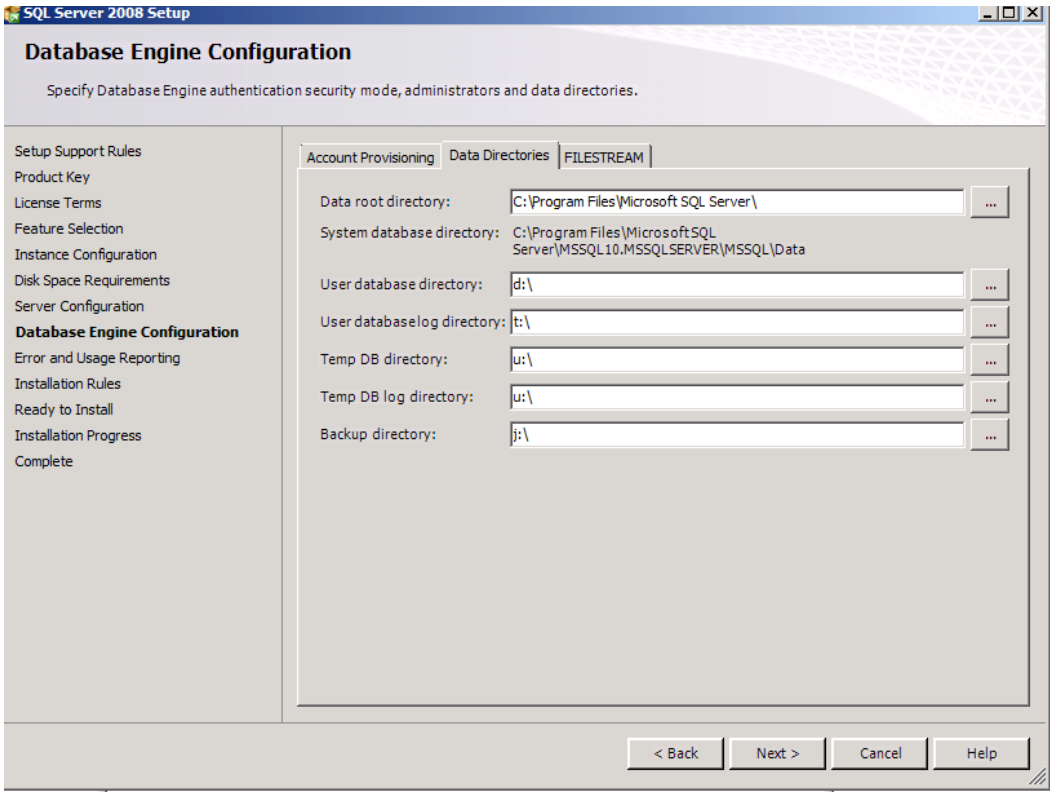


Figure 7: SQL Server Installations Data Directories

Command Line Setup

SQL Server allows the adjustment of SQL Server features via the command line after installation of an instance is complete. The exam has questions related to setup capability at the command line. The setup.exe file located with the SQL Server source binaries provides the executable to use when making command line adjustments.

Here is an example of a command line install of SQL Report Server for an SQL instance called BOSS302 in the MUSTANG domain.

```
Setup.exe /q/ACTION = Install/FEATURES=AS /INSTANCENAME =MSSQLSERVER/
RSSVCACCOUNT = "MUSTANG\RPTSvc" /RSSVCPASSWORD = "testme01_"
```

This will install SQL Server Reporting Services to the default SQL Server instance, with the assignment of a Reporting Services startup domain account of RPTSvc and a password.

The best way to prepare for the command line capabilities of an SQL Server installation is to do it hands-on with scripts. A quick reference for this capability is to open a command prompt, navigate to the SQL Server 2008 install source, and type setup.exe /?.

Test Tip: The setup.exe command with switches will always use the FEATURES=AS switch. This is important because on the exam the questions will insert the word FEATURES=IS, which is a quick clue that the answer is wrong. Also note how the target upgrade SQL Server instance is listed. If it is default, then no instance specification is required. Read the question very closely and you can identify which answers can be eliminated easily.

Implementing Database Mail

The SQL Server email feature is important to many database management operations. SQL Server can email with errors, as a result of SQL application activity, and for SQL Agent job notifications. SQL database mail is not enabled by default. To configure database email open the SQL Server Management Studio and expand the Management section. Right-click on database mail and select configure. Clicking the new profile option will enable database mail for the SQL Server instance in which you are connected. Assign a profile name and add an SMTP account. The SMTP account data will require information on your SMTP relay in use or Microsoft Exchange environment. The next option is to configure the profile to be public, which means it can be accessed by all users of any database, or private which requires access to use it. Select Send Test E-Mail to test the configuration, and View Database Mail Log if the test is unsuccessful to troubleshoot.

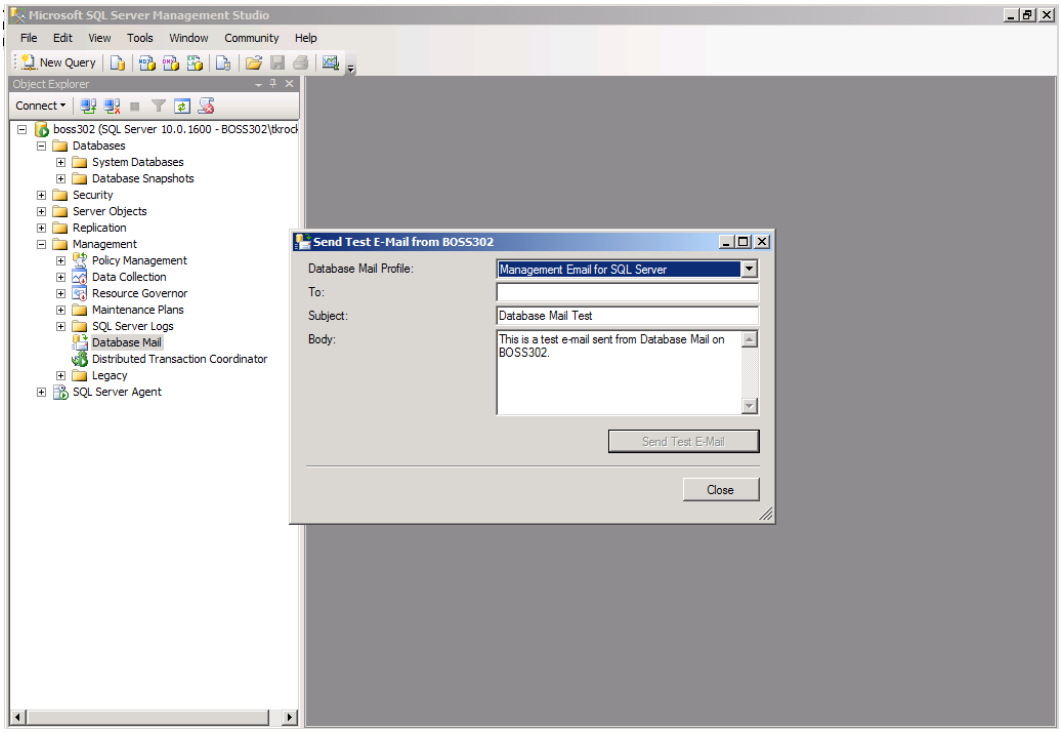


Figure 8: Send Test E-Mail for SQL Database Mail

Test Tip: The stored procedure *sp_send_dbmail* operates from within MSDB for system wide database email operations, so for security purposes ensure each database user that will send database email is a member of the *DatabaseMailUserRole*.

SQL Server Configuration Manager

Once SQL Server is installed the SQL Server Configuration Manager is the tool to use to configure your SQL instances for service accounts, network ports, and other data such as hiding SQL instances or disabling SQL protocols. The primary SQL network protocols are TCP/IP and Named Pipes. Best practices dictate the use of TCP/IP and a selected SQL port. SQL Server defaults to port 1433, but this is easily changed to support higher levels of security by not allowing SQL to communicate on the default port. Encryption of connections is a configuration option here and allows the addition of certificates used for encryption.

Test Tip: Exam questions that require reconfiguration of any network parameters, SQL instance hiding, and SQL listening protocols should always alert you to look for an answer where the SQL Configuration Manager is used.

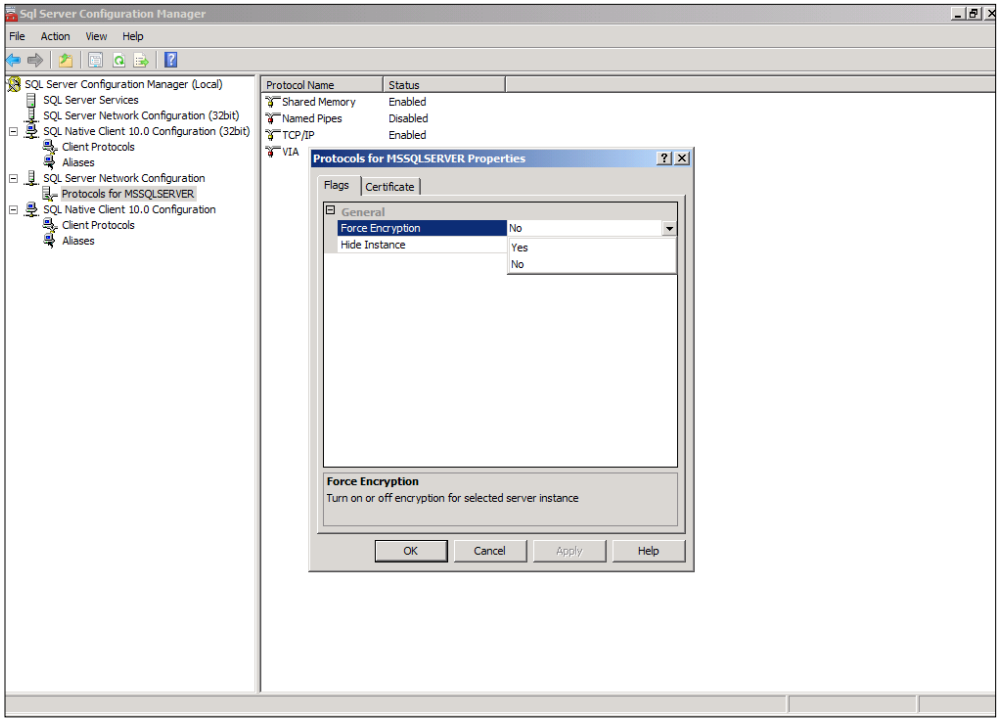


Figure 9: SQL Configuration Manager Showing Protocol Properties for an Instance

Domain 2: Maintaining SQL Server Instances

Understanding System Stored Procedure: sp_configure

SQL Server provides several tools to manage installations once installed. The system stored procedure sp_configure, the SQL Server Management Studio, SQL Powershell and the SQL command line.

The exam tends to focus on the use of the T-SQL storage procedure sp_configure. In some instances the manual will point out when the SQL Management Studio can be used over sp_configure, but this is only in cases where the change is quicker in SQL Management Studio. The stored procedure **sp_configure** is a powerful tool which provides many of the configuration capabilities for adjusting SQL server installation and activating or changing features. This review will focus on common uses of sp_configure. This is a system or built in storage procedure that supports SQL tasks across all domains of the exam including management, security, and performance tuning.

Test Tip: Know what the syntax options are for sp_configure and how to use it to adjust features on SQL Server instances.

Manage SQL Server Agent Jobs

The heart of managing SQL Server instances is configuring jobs which are managed by the SQL Server Agent. The exam covers, in detail, numerous fundamentals of using the SQL Server Agent. SQL Server Agent is the subsystem of SQL Server which runs jobs on a schedule, conducts the appropriate logging of job activity, fires alerts, conducts database commands, and uses required security contexts for the jobs.

All SQL Server Agent configured jobs are stored in the system database msdb. For exam purposes remember that the fastest route to backup configured SQL Server Agent Jobs is to backup system database msdb.

SQL Server Agent jobs are created in SQL Management Studio by right-clicking the SQL Server Agent jobs tab and following the configuration dialogue flow. The Agent is running and ready for jobs if the green arrow is present. The job will ask for a name, owner, and a category of database activity (e.g., database maintenance, data collector, or Full-Text). The steps to the job are inserted next. These steps contain the SQL commands and syntax needed to complete the required activity. The steps can consist of external code, operating system support, or Power Shell scripts. Each step allows specification of the database target and credentials to use for the step. Steps in a SQL Agent Job can access different databases as required and use different credentials. The Schedule for the jobs is the next configuration step, and it supports complete date and time selection, recurring, one time, or the option of taking advantage of idle CPU time. An important part of the selection process is to enable the schedule. If the schedule is not enabled the job will not run. The SQL Server Agent jobs support alerts that can write to the SQL Server event log or a WMI event. The alert dialogue allows configuration of severity levels and custom error number. In addition to the alert the job configuration supports responses with options to the alert, which include notifying SQL operators or running defined SQL jobs. The Agent will send alerts via email, pager, or a **net send** command. In most cases net send is outdated. The Agent will provide notifications of job activity and results via email, page, net send, Windows Application event log, or provide for deletion of the job. Conditions are set for each notification method, for example, when the job fails to send an email. Lastly, the target section is configured. The Agent jobs can target the local server or other defined linked SQL Servers.

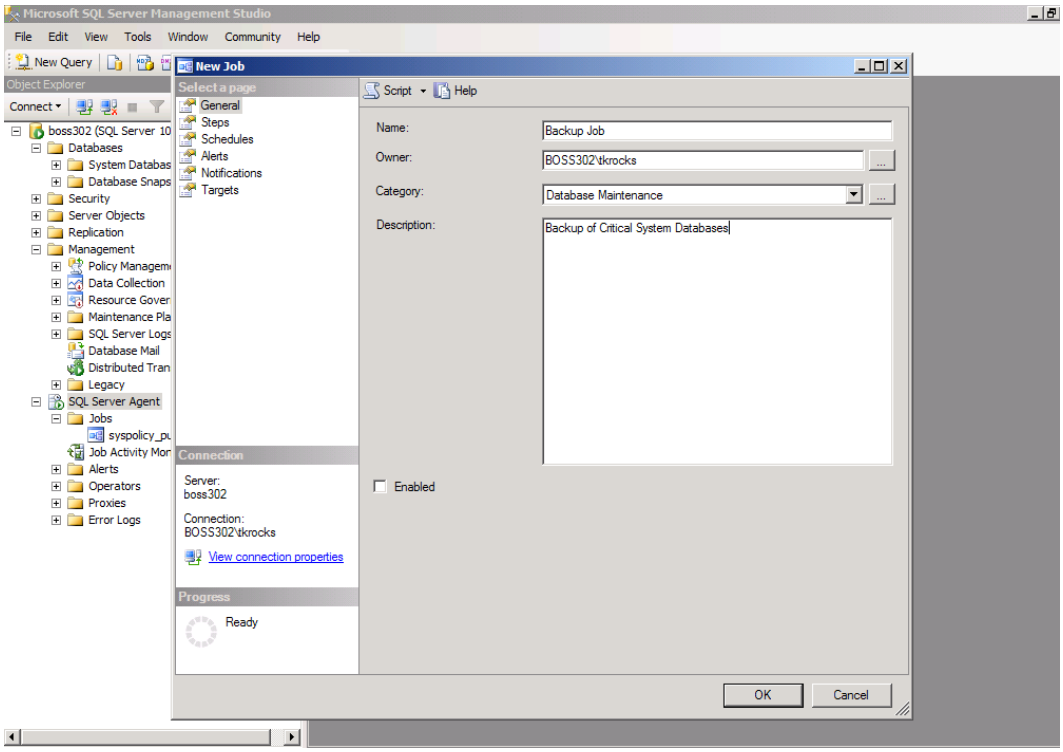


Figure 10: SQL Server Agent Job Creation Dialogue

Test Tip: When managing SQL Server Agent jobs, all activity is logged to the file SQLAgent.out file in the SQL Server log directory, with the full path \program files\Microsoft SQL Server\mssql10mssqlserver\mssql\log. This file is independent of any alerting or logging setup for the Windows Application event log, email, or WMI logging.

Scheduling of SQL Agent jobs is a required skill for the exam. SQL Server Agent jobs can have multiple schedules. For example, the job named 4.6Power 1 and 4.6Power 2 could be scheduled to run the 1st of the month, 15th of the month, and 30th of the month. When requirements change always determine if the job schedule must change as well.

Test Tip: Some Agent jobs are dangerous, i.e., they are configured to delete data such as older records or other data. In some cases the job may be dependent on preparatory activity for tables. As you read the exam questions, when you spot an SQL Agent job that may delete or remove data and its use is as needed, make sure you keep the job disabled.

Manage SQL Server Agent Alerts

Once SQL Server Agent jobs are configured the alerts that the jobs create are integral to SQL Server management activities. Critical jobs for backups, data transformation, data removal, and export of data (all of which support some business activity) must have alerts to ensure their success or failure is monitored. In some compliance situations such as PCI Visa certification, this alerting is required and proof must be available that it exists.

The SQL Server Agent can alert via email, SQL event log, or WMI. WMI or Windows Management Instrumentation is an alert that is parsed to a system that is running WMI based agents, such as Microsoft System Center. These alerts can use the rich programming capability of the WMI model. These alerts can also use any Windows performance object contained in the performance monitoring counter sets. If you have a case where the alert should monitor SQL Server disk space, then the alert should use the Windows performance counter element LogicalDisk with a configuration of % Free Space. The solution of combining SQL Server Agent performance condition alerts with the underlying Windows Server operating system performance object is very powerful.

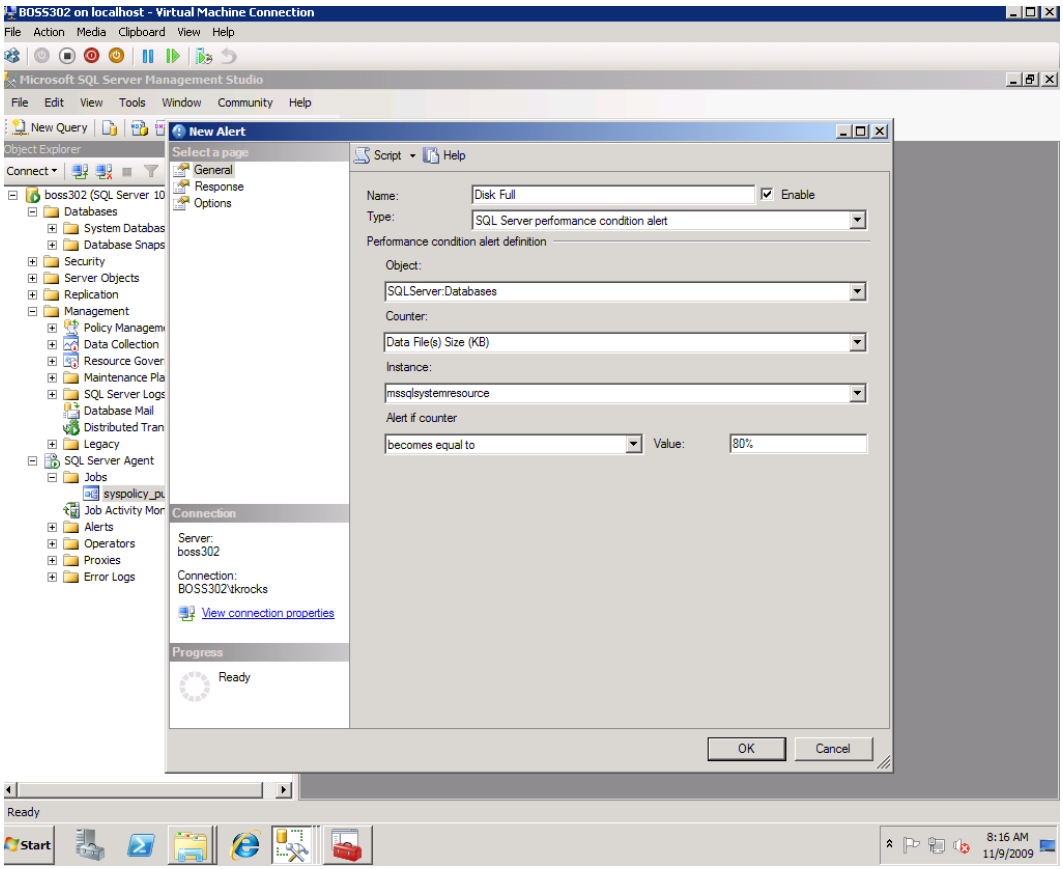


Figure 11: SQL Server Agent Alert for Database Disk Size > 80%

SQL Server Agent alerts work in conjunction with other SQL Server notification features. For example, if you have configured stored procedures running, ensure that they use the option to RAISEERROR WITH LOG, and this will work with the configured SQL Server Agent alert.

The ubiquitous existence of email enabled cell phones today underscores email notification as a viable and effective SQLS Server alert method. Emails configured as alerts can notify for both performance and event alerts. This configuration also builds on the database mail settings covered in the Domain 1 of this manual.

Test Tip: SQL Server Agent logging uses the Windows Application Event log to create SQL Server Agent alerts. If the Windows Application event log is using the setting to not grow and requires a manual intervention to clear it, then SQL Server Agent alerts will fail if the Application event log is full.

Manage SQL Server Agent Operators

SQL Server Agent operators are the credentials that receive alerts on jobs. The operator is defined with the SQL Server Agent job management section. The Operators folder supports the creation of operators that are available for job notification. The availability of an operator is a key part of the operator options set and allows for designation of when operators may be on shift, or to support helpdesk operations or DBA groups. The setup of an operator includes methods to notify the operator, which are via email, pager, and net send address. Again, **net send** is a method that will send a message to a logged in computer so this method is not as reliable as email. The operator view also contains a notification list that stores all alerts by type sent to the defined operator. The operator objects created for notification are manageable, thus if someone goes on vacation or is not available that notification operator object can be disabled, or deleted in the case of an operator change.

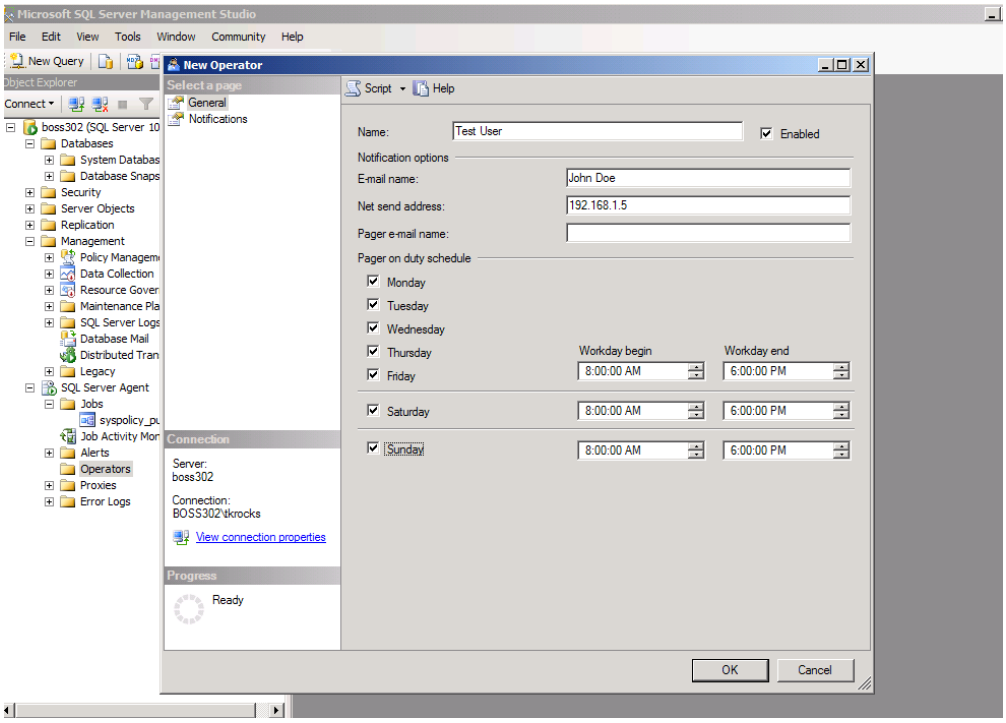


Figure 12: SQL Agent Operator Setup

Test Tip: In a scenario where one operator should receive all notifications or the exam question asks to notify an operator if another operator is not available, make sure you understand the solution is to configure the fail-safe operator in the SQL Server Agent core properties.

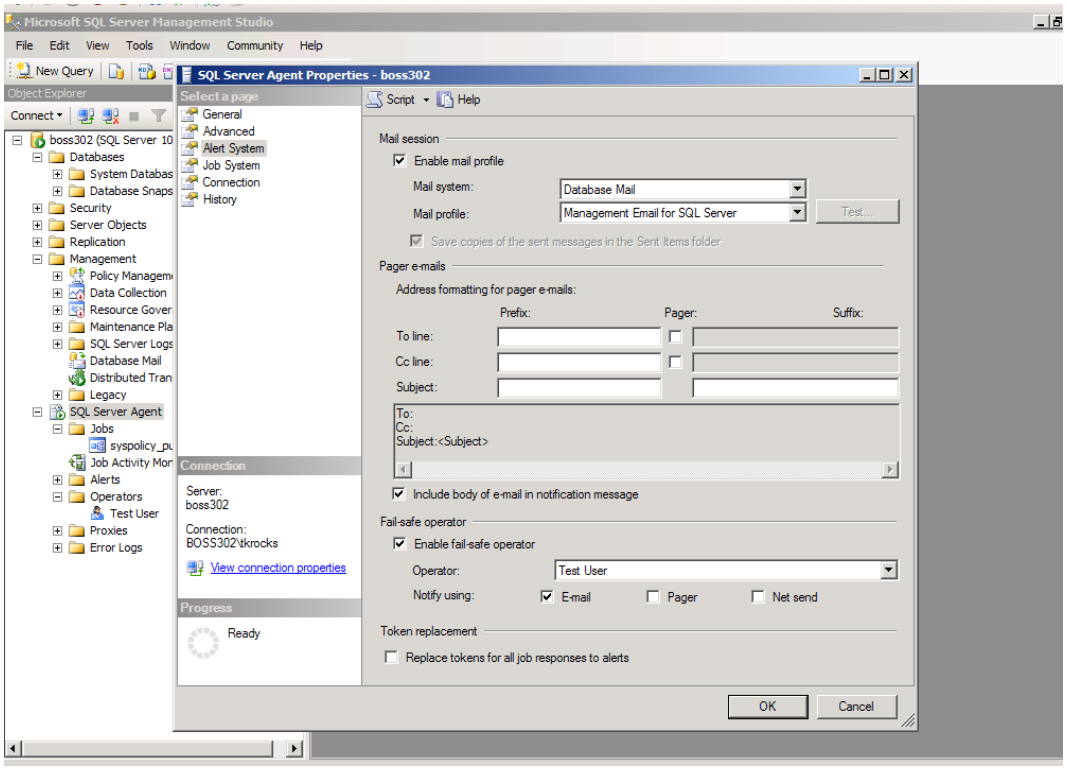


Figure 13: SQL Server Agent Fail-Safe Operator

In cases where database mail is used to provide email services to SQL Server Agent always remember that if any new database email profiles are created and used for SQL Server Agent notification, the SQL Server Agent service requires a restart in order to use the new profile. SQL Server Agent restarts do not affect the uptime of the SQL Server database engine.

The SQL Server Declarative Management Framework (DMF)

SQL Server 2008 includes the ability to create policies that work toward best management practices. The collection of tools within the DMF focus on the policy capabilities of SQL Server to support ease of management from one to many SQL Server instances. The DMF targets surface features. For example, protocol types in use, servers, logins, databases, tables, and users. The policies enforce security and database settings.

Policy management is achieved via the Policy Management node under the Management area in SQL Server Management Studio. The policy section supports Microsoft's DMF goals for SQL Server, which is to allow policy control that mirrors the ease and capability of Active Directory's Group Policy features but is targeted toward SQL Server security, settings, and databases. Policies are created based on need and the creation of conditions the policy enforces. The conditions are comprised of SQL Server functions and settings, for example, @RecoveryModel type, and actual calculation functions for determining conditions, for example, addition, subtraction, and other math functions.

Once a condition is selected in building the expression, the options for that field are available. In this example, setting a database policy for Recovery Model, when selecting the field @RecoveryModel, the options of Full, Simple, and Bulk-Logged are available. Conditions can be combined to form multiple element policies. The policy is applicable against all databases or select databases. Policy configuration can be restricted based on the formulation of policy facets. A facet can determine exclusions for policies or other considerations.

Policy Management contains an evaluation feature that allows evaluation of polices prior to production implementation. Facets of a database are the feature settings, and the SQL Server policies set and manage facets of a database. Policies can evaluate the conditions they are going to employ and different modes exist to include:

- **On schedule**, which is running a policy on a schedule.
- **On demand**, which is running a policy manually.
- **On change: log**, which will log compliance with the policy as it would have done in an evaluation mode.
- **On change: prevent**, which will stop the policy from executing if a policy evaluation fails.

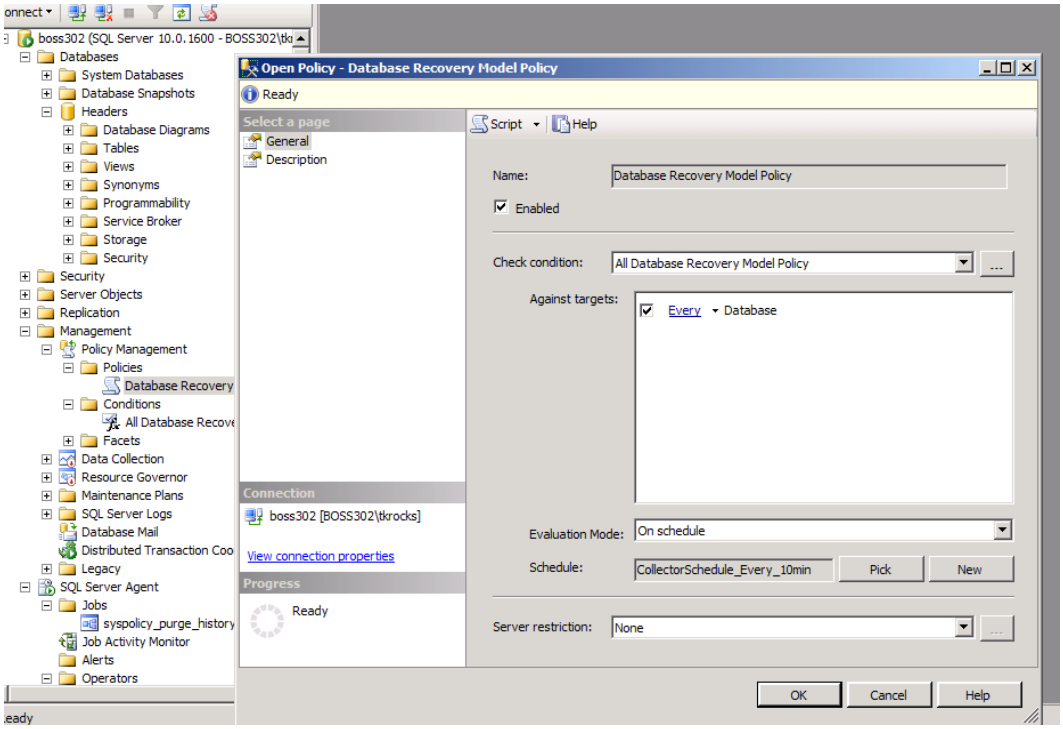


Figure 14: SQL Server Policy for Database Recovery Model

Test Tip: Any DMF policy created that is left in the on demand evaluation mode will not enforce the policy. The policy must be set to on schedule to ensure it is executing. For example, if a policy for auditing does not allow SQLMail enabling but SQLMail is running the reasons can be that the policy is set to on demand. When policies execute on schedule, they use the security role of sysadmin, which ensures that a proper security context is in use to execute the policy. SQL Administrators can log on and evaluate a policy manually, but this will use the logged on user's security context and it may not have all required permissions to run the policy successfully.

Backing Up SQL Server 2008

The backup and protection of SQL Server data is one of the most important SQL Server administrator tasks. Companies that lose critical data do not survive for long. SQL Server 2008 provides built in back capability. The exam has multiple questions concerning the setup, administration, and restore operations for SQL Server Backup. Any SQL Server database or database log is easily backed up by right-clicking on the database name and selecting Tasks\Backup. However, this type of ad hoc backup is not reliable nor the proper method to backup SQL Server production databases. SQL Backups require configuration so that they are scheduled, automatic, use defined backup sets, and log their results.
SQL Server provides for several types of backups as follows:

- **Full Backup** – Complete backup of a database file. This type of backup only represents the database at the time of backup.
- **Differential Backup** – A backup that only adds data changed since the previous full or differential backup.
- **Transaction Log Full Backup** – Complete backup of a transaction log. Transaction log backups can be Full recovery model or bulk-logged recovery model. The full recovery model allows recovery back to a point in time by restoring backed up transaction logs. The bulk-logged recovery model is for temporary backups and operations during a large bulk operation.
- **Partial Backup** – A backup that only addresses full data in the primary filegroup and any specified filegroups.
- **Differential Partial Backup** – A backup that only contains the data extents modified since the most recent partial backup.

Establishing a SQL backup is best handled through the DMF's Maintenance Plan Wizard.
This wizard is launched through the Maintenance Plans node under Management in the SQL Server Management Studio. This wizard allows the setup of multiple schedules when using full and differential backups, and for different times for a transaction log backup. You can select the Backup Database (Full) option or others as required. You can also select other common tasks, such as Shrink Database. The wizard allows configuration of the back location, which can be disk or tape. Normally, backups on disk are then backed up to tape by another product, such as VERITAS Backup Exec. Compression is also an option to ensure that data storage is optimized. For full backups the option exists to append to the designated backup location, or overwrite. In most cases you will append data for the backup so that a point in time is selectable. The result of the backup job can be emailed or written to a file.

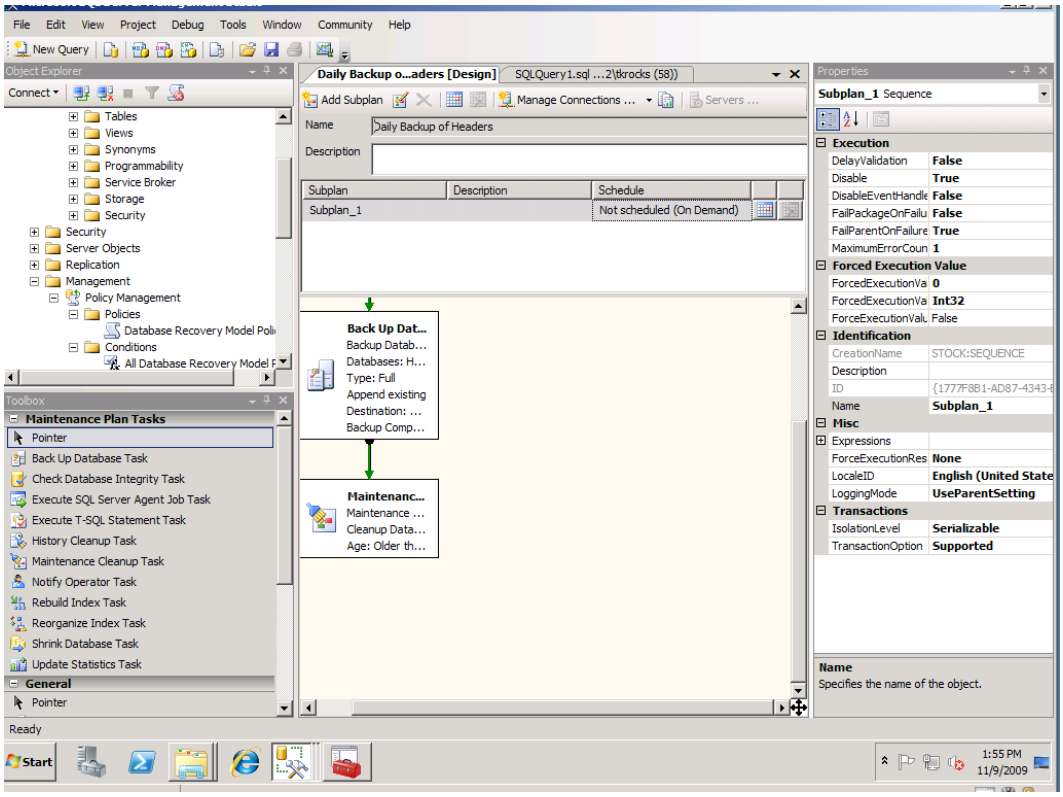


Figure 15: SQL Server Backup Plan

Test Tip: A standard workable SQL Backup schedule is as follows. We are protecting a database called ponypowersalesDB. This database is backed up with the following schedule and type of backup:

- 2300 PM Nightly MTWTFSS: Full Backup
- 1000 AM Daily MTWTFSS: Differential Backup
- 1400 PM Daily MTWTFSS: Differential Backup
- Every 30 Minutes of Each Day, MTWTFSS: Transaction Log Backup

The exam will challenge you to conduct a database restore for a database problem with ponypowersalesDB. Pay close attention to the order of the backups and what time the exam indicates the problem has occurred. In this example, a failure occurs at 3:00 PM or 1500 hours. The restore order is (1) the full database backup from 2300; (2) the 1400 differential backup; and (3) each transaction log backup before 3:00 PM. This will restore the database as quickly as possible.

In some recovery cases, the restore will be for a database already restored. The database recovery model Full Recovery allows you to restore a database using the WITH RECOVERY option. The Full Recovery model allows the use of transaction log backups, while the Simple recovery does not. This will allow you to restore a backed up database, which places the database in standby mode and allows the full backup with recovery option described above.

The COPY_ONLY backup option is also covered on the exam and allows backing up data from a database or log file without affecting the backup sequence. If you recovered a file within a database from a previous backup and want to ensure playback of the transaction log backups for true point in time recovery then restore whatever part of the database file or filegroup is required then backup the transaction log with the COPY_ONLY option and restore it WITH NORECOVERY and then restore it again WITH RECOVERY. Using WITH_RECOVERY as the last option in the restore will bring the file back online. This is known as a tail-log transaction log backup. When the failure of the database in question is reported or known, the quick backup of the transaction log attempts to recover transactions between the failure and the transaction log backup. In many cases this will ensure you are the SQL Administrator hero of the day because normally no data is lost, especially if best practices are followed and the transaction logs are on different disks than the database files.

Another option, which the exam will also cover, is the SQL 2008 backup option called READ_WRITE_FILEGROUPS. In database design read-only databases are usually filegroups with the database structure. Backups are quicker because read-only data is skipped.

Domain 3 - Managing SQL Server Security

SQL Server security questions account for 15% of the exam. This is an area requiring fluency and accuracy. The exam will expect understanding of SQL Server logins, server roles, the management of users, SQL Server instance permissions, database permissions, schema permissions, object permissions, auditing, and data encryption.

SQL Server Logins and Server Roles

SQL Server logins consist of Active Director or domain login accounts, local computer login accounts when a SQL Server may not be a member of a domain, and the SQL Server account. SQL Server accounts are logins which are maintained by SQL Server and don't belong to the local computer SAM or an Active Directory domain. The type of account that SQL Server will allow for a login is determined by the SQL Server authentication mode. The SQL Server authentication mode is set during install or adjusted at any time in the Server Properties section of the Server node or with an sp_configure change.
The two authentication modes are Windows Authentication, which will use Active Directory or local server accounts, and SQL Server and Windows Authentication, which only allows the use of SQL Server accounts. The latter authentication is commonly referred to as "mixed mode." Best practice dictates that only Windows Authentication mode be used, which reduces server attacks that attempt to compromise SQL Server based accounts. The default SQL Server security mixed mode login is called "sa."

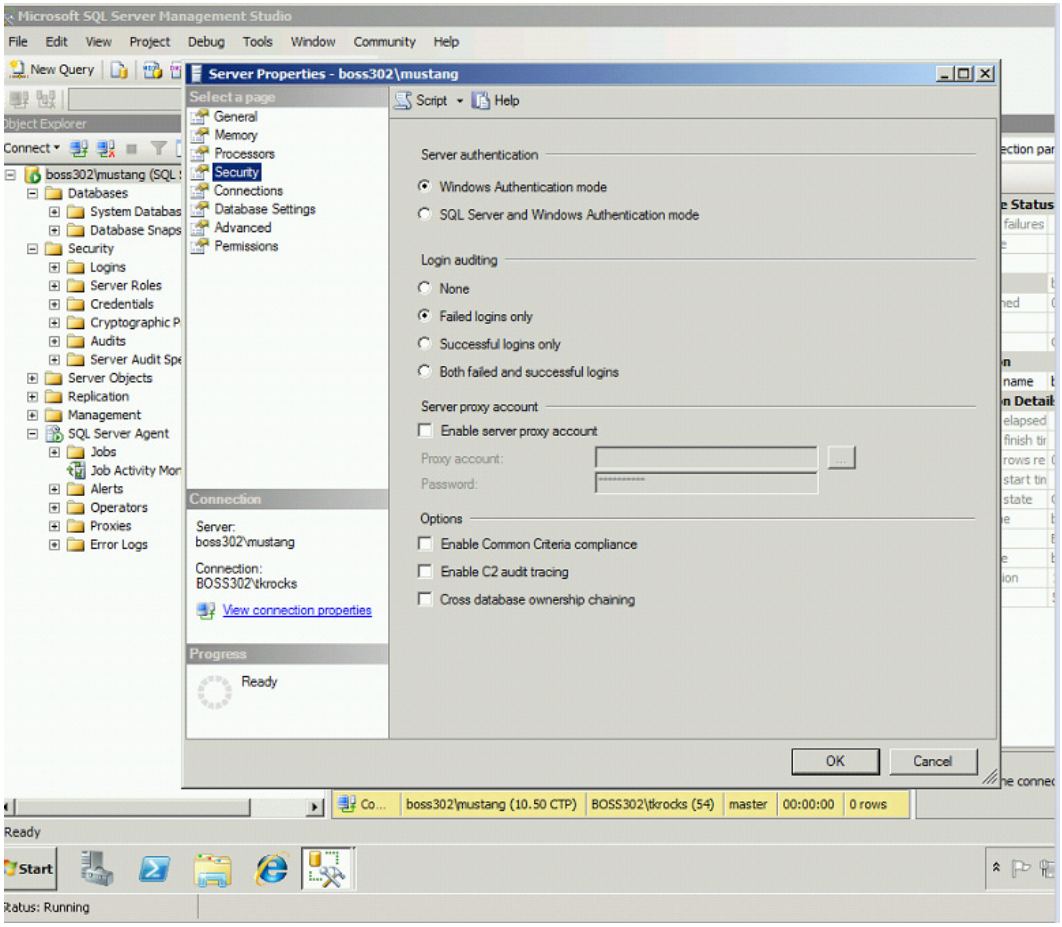


Figure 16: Server Properties for Server Authentication Modes

Logins are created through the Security Node of SQL Server Management Studio. Right-click on the Logins sub node and select new login. The new login is given a name, but in the case of Windows authentication this name should correspond to the Active Directory or local server account name. SQL Server Login creation allows a search of the Active Directory or the local server accounts. No password is assigned for a Windows authentication login as it will use the password maintained via the Active Directory. A SQL Server login requires a password and options as to whether to use password policy, password expiration, and a change on first login. Additionally, SQL Server logins can be certificate based for application support, mapped to asymmetric keys, or mapped to credentials. A default database for the login is the last portion of the new login dialogue. Logins can be mapped to the databases they are supporting or should have access to.

Test Tip: The following syntax will create users in query editor. For a Windows user, add the command FROM WINDOWS and put the user name in the context of DOMAIN\USERNAME. To force a password change for a SQL login, add the MUST_CHANGE command to the CREATE LOGIN syntax.

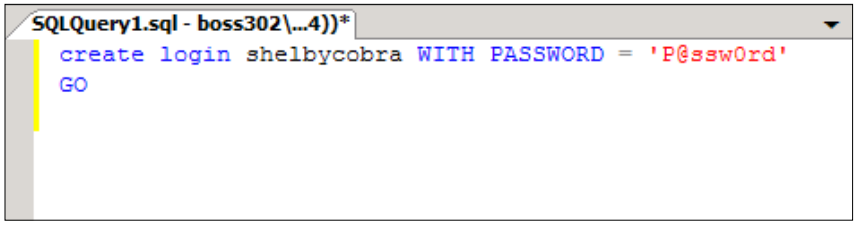


Figure 17: Create Login Command

Password Policies

SQL Server password policies provide the same robust security enforcement as Windows 2003 and Windows 2008 Active Directory complex passwords enforcement. The parameters include enforcing password complexity, password expiration, and then enforcing the policy. The policy applies ONLY to SQL based logins, not Windows Active Directory based accounts.

Test Tip: Use the ALERT LOGIN T-SQL command with CHECK_POLICY to ON to enforce the password policy.

SQL Server-Level Roles and Database-Level Roles

SQL Server security contains several fixed or default server roles. By placing SQL logins into the available server fixed roles, the login receives the privileges in which the role is configured. Fixed server roles apply server wide not just to individual databases. SQL Server also contains multiple predefined database roles that are applicable to databases individually versus the SQL Server range of a fixed role. An administrator can create custom database roles, but the predefined ones answer most SQL security issues.

Fixed Roles	Definition
SysAdmin	Any member can perform any action on the SQL Server
ServerAdmin	Any member can set configuration options on the server
SetupAdmin	Any member can manage linked server and SQL Server startup options and tasks
Securityadmin	Any member can manage server security
ProcessAdmin	Any member can manage kill processes running on SQL Server
DBCreator	Any member can create, alter, drop and restore databases
DiskAdmin	Any member can manage SQL Server disk files
BulkAdmin	Any member can run the bulk insert command
Public	Default role for every SQL Server login

Table 3: SQL Server Fixed Security Roles

Predefined Database Roles	Definition
Db_owner	Members have full access
Db_accessadmin	Members can manage Windows groups and SQL logins
Db_datareader	Members can read all data
Db_datawriter	Members can add, delete, or modify data in tables
Db_ddladmin	Members can run any data definition language (ddl) commands
Db_securityadmin	Members can modify role membership and manage permissions
Db_backupoperator	Members can back up the database
Db_denydatawriter	Members cannot add, modify, or delete any data in user tables
Db_denydatareader	Members cannot read any data in the user tables

Table 4: Predefined Security Roles

Test Tip: Exam questions in this area will focus on the least amount of privilege. If the question indicates that backup capability is required and the ability to kill processes, then assign only those two roles; do not push up the stack to SysAdmin or ServerAdmin.

In the case of custom database roles, the role should be added to the respective function of predefined database roles. If you create a database role called Shelby_Backups and want the members to backup databases, then add the Shelby_Backups role as a member to the db_backupoperator role on the database in question.

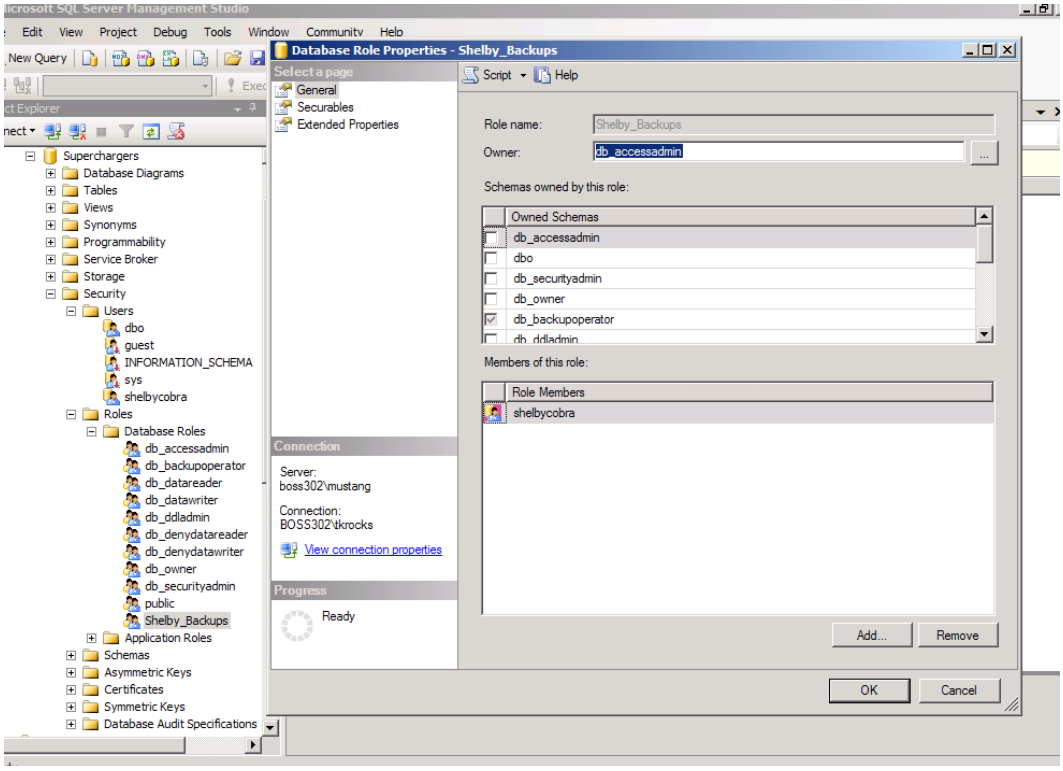


Figure 18: SQL Server Custom Role with Members

User mapping is also important to complete the setup of database roles. When setting up a user administers have the option of mapping the user to a database and selecting a role. Again, best practice is to use least amount of permissions to maintain a secure audit posture. Logins not mapped to a database have no access.

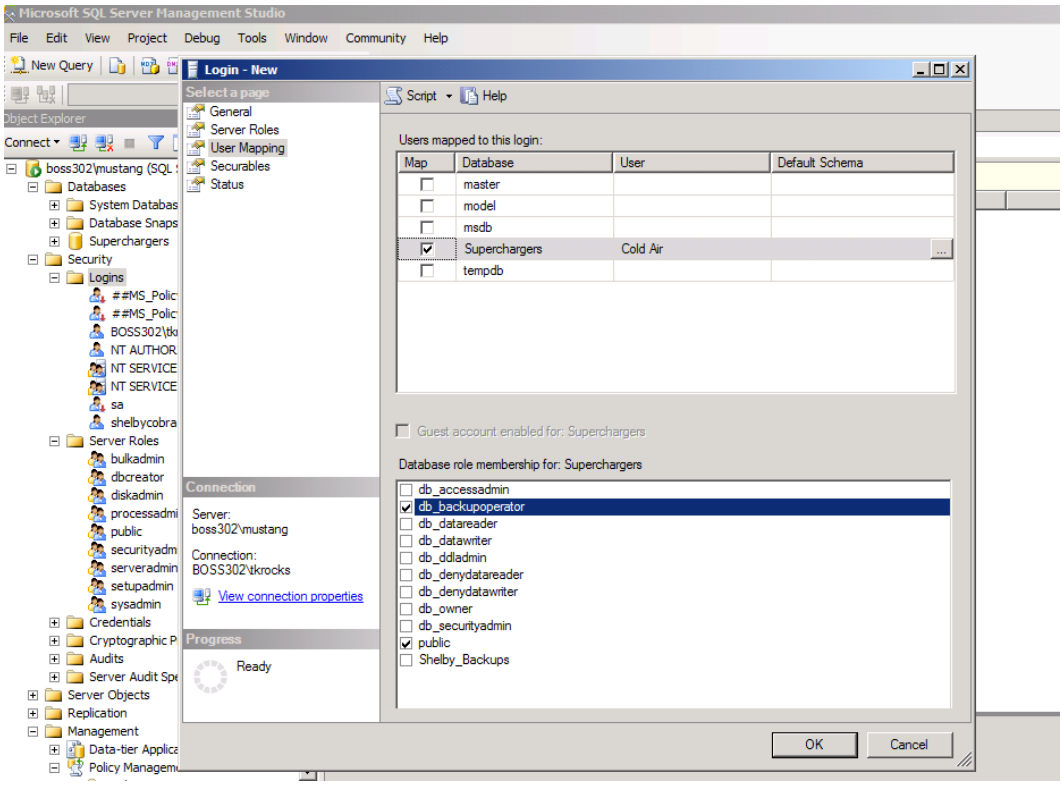


Figure 19: User Mapping for a Database Role

The Guest, Public, and DBO Principals

All SQL Server databases include the guest user, which is created anytime a database is created. If a SQL login does not have an account in a database, then it inherits the guest user rights. By default the guest user is disabled.

All databases have the database owner or dbo user added upon creation. This role is always a member of the database-level role called db_owner. This user can configure and conduct all activities in a database. Each created database is a member of the public database role. Any user who does not have permission in a database will inherit the public permissions. As with guest user, for security purposes, this user should remain without rights.

SQL Server Instance Permissions

The types of SQL Server instance permissions focused on in the exam are logon triggers, cross-database ownership chaining, and impersonation. The concept of separating security boundaries with SQL Server instances is in play here. Disparate security measures for password policies, login types, and SQL Server Authentication mode all should alert you to use a new SQL instance to create good security segregation.

Logon Triggers

Logon triggers are used to activate a stored procedure in SQL Server. Logon triggers are covered on the exam in the audit and control server portions of the questions. When audit type questions on the exam describe methods to track user activity the answer is the logon trigger. The trigger can be set to run for access to the SQL Server or a database. Logon triggers include the ability to activate stored procedures and ensure they can enforce and prevent actions. A logon trigger can ensure that a login is not running more than two sessions to an instance at any one time. The logon trigger can enforce (via checks) conditions about the user session.

Test Tip: Exam questions that ask about restricting user sessions or checking the amount of sessions are asking about the use of a logon trigger. The logon trigger is created in the master database to ensure SQL Instance wide enforcement.

Impersonation

SQL Server impersonates credentials that originate as a Windows user when communicating within SQL Server or SQL services. When SQL runs a process it will use the context of the login, and this transposes to other functions, processes, Linked SQL Servers, and database access.

SQL Server has a Common Language Runtime (CLR) built in to execute managed code within the SQL Server memory space. The SQL.Context.WindowsIdentity in the CLR is used to impersonate a Windows account with permission to access the required databases and code. This code call will set the context under which the code will execute which allows creation of Windows accounts with the appropriate permissions to execute in the CLR space.

Test Tip: Exam questions that offer answers for impersonation by adding the Windows account to the Active Directory Domain Admins group or local server Administrators group are wrong answers. The enforcement of security always dictates the least amount of privilege to execute successfully. Always evaluate exam questions in the security arena with a focus on least permissions.

Cross-Database Ownership Chaining

Database application solutions sometimes require the access of multiple databases in a SQL Server instance. In cases where this is required a chain of ownership is created across the required databases. The second database in the chain will not evaluate permissions of an object that was previously authenticated in the first member of a chain. This does provide for a speed increase in some cases; however, this feature is disabled by default.

Test Tip: Cross-Database chaining is enabled server wide via the SQL Server Management Studio and Server Properties, or with sp_configure cross db ownership chaining =1 to ON. The recommended solution for exam and management purposes is to use ALTER DATABASE to turn on cross-database chaining or individual databases that require it rather than setting it server and instance wide.

Managing Schema Permissions

Another key securable in the SQL Server security ecosystem are database schemas. Database roles, such as dbo or other logins, own or manage schemas throughout the database realm. A key enhancement with SQL Server 2005 was the introduction of user and schema separation. SQL 2008 maintains this security split. Schema objects include tables, stored procedures, views, and roles that can access and manage these interactions. The practical effect of this security feature is that a SQL login named Shelby302 can own a schema named SchemaPower, which contains a table, a stored procedure, and a user-defined function. The SQL login has access to and ownership of all of these database objects through singular schema ownership.

Test Tip: In SQL 2008, it is simple to change the owner of a schema from one login to another. This is the least disruptive administrative path. If a user named Shelby302 leaves the company, a simple change of ownership to the new login completes assignment of all the objects in the schema.

The inclusion of the schema object creates the four-part contents of a fully qualified object name in SQL Server 2008; for example, BOSS302.HEADERS.SCHEMA1.TABLE1. In other words, the SQL Server name, database name, schema name, and the object. SQL Server 2008 supports the use of synonyms to access these four-part defined objects. For example, the code below creates a synonym of **BOSS** for the BOSS302.HEADERS.SCHEMA1.TABLE1, and then conducts a **select** of all data.

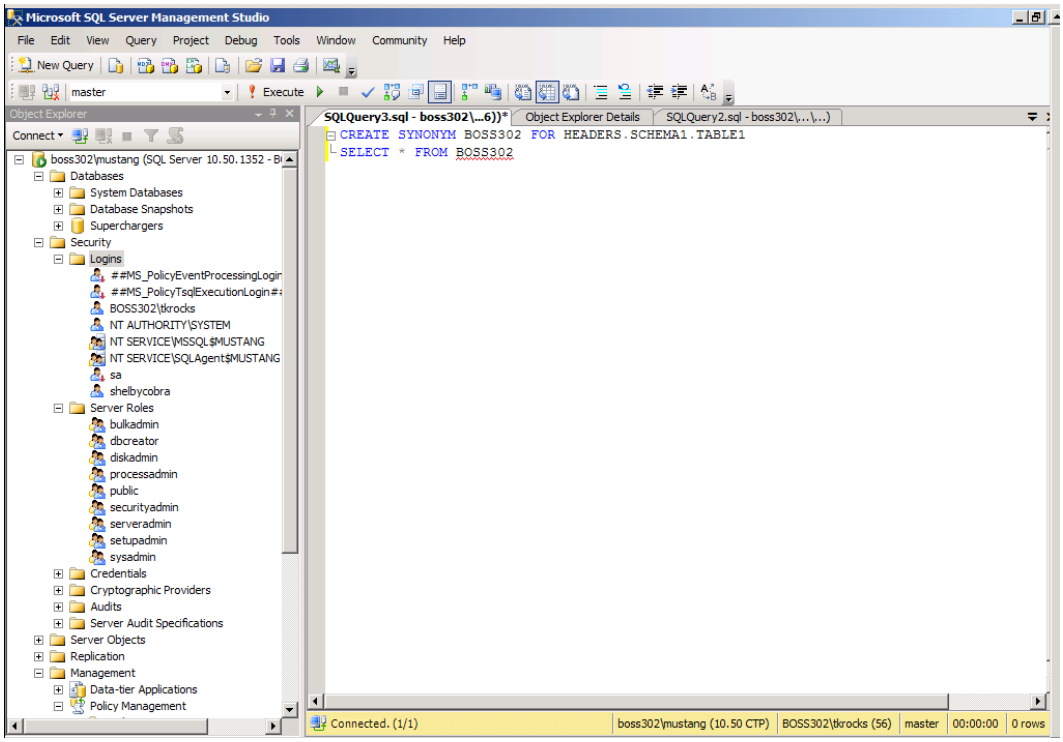


Figure 20: Creation of a Schema Object Synonym

Test Tip: The ALTER AUTHORIZATION versus the ALTER USER command quickly changes schema ownership. The movement of schema objects is accomplished quickest through ALTER SCHEMA.

Auditing SQL Server

The auditing of SQL Server is one the most important day-to-day tasks of an SQL Administrator, and this is covered in multiple converging areas on the exam. A collection of methods is reviewed below, which, taken together, present the picture of a valid SQL Server audit posture. From PCI (Payments Card Industry) certification to HIPPI (Health Care Regulations) the storage of data, which is comprised of personal or financial information, dictates that SQL administrators must have a command of SQL audit methods. SQL Server can audit activity and send the output to multiple locations to include files, the Windows Application Log, and the Windows Security log.

Test Tip: SQL Server can only access the local server's Windows Security log if the SQL Server service runs as a Local System, Local Service, Network Service, or an Active Directory based account which contains the rights to access the log.

SQL Server allows creation of objects for auditing via the CREATE SERVER AUDIT command. An audit condition is set up with this command and a destination as described above is selected. Once audit conditions are created, the SQL administrator can add events to the condition with CREATE SERVER AUDIT SPECIFICATION and CREATE DATABASE AUDIT SPECIFICATION commands. If as an SQL administrator you were tracking failed logins, you would create an audit specification that tracks the failed login and adds it to the audit file and location.

The following are examples of standard SQL Server audit specifications and can be applied to all users:

- SUCCESSFUL_LOGIN_GROUP
- LOGOUT_GROUP
- FAILED_LOGIN_GROUP
- AUDIT_CHANGE_GROUP
- SERVER_PERMISSION_CHANGE_GROUP

The following are examples of database specific audit specifications and should be applied to all databases:

- DATABASE_CHANGE_GROUP
- DATABASE_PERMISSION_CHANGE_GROUP
- DATABASE_OBJECT_ACCESS_GROUP

The following are database table specific audit specifications that ensure auditing is applied at the lowest point of the database object:

- SELECT (this audit feature captures the SELECT statement not the data!)
- INSERT
- UPDATE
- DELETE

Protection is also required for the SQL Server audit files. Ensure that you are familiar with auditing the reading and changing of the produced SQL Server audit files. An audit set on master.sys.fn_get_audit_file records any login that reads the audit file or attempts to change it.

Each database has a property called **Change Tracking**, which is enabled at the database object level and then available to any table within the database. This will track all changes to data in a database within a table and allow you to specify an interval to retain the change data.

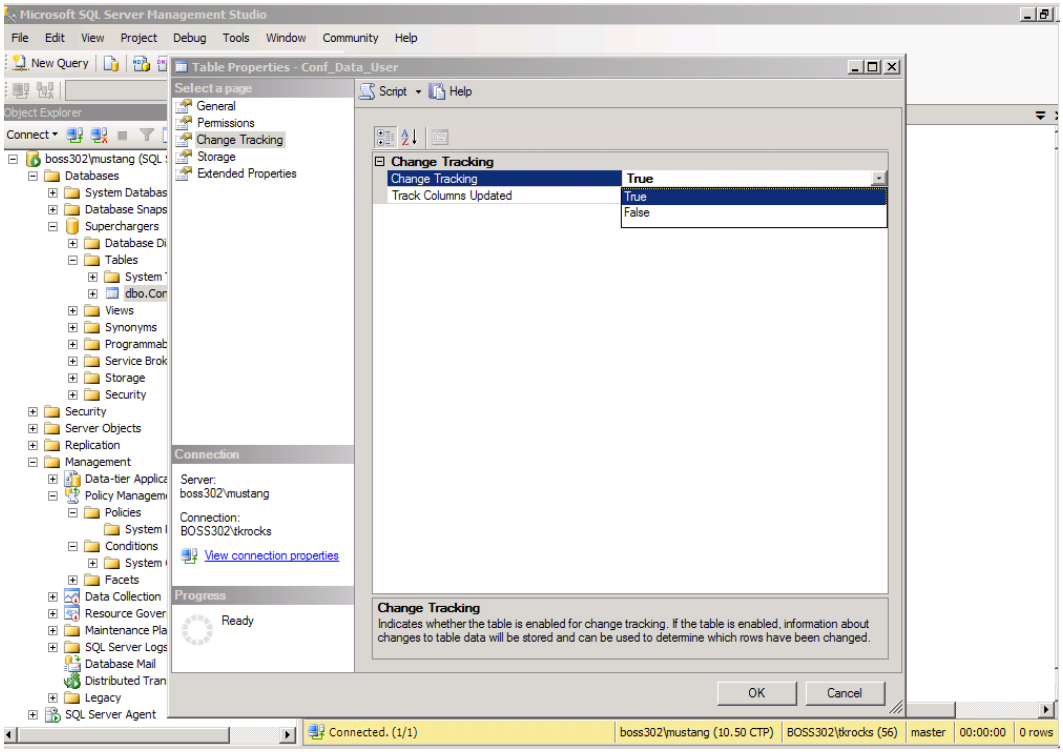


Figure 21: Change Data Tracking Configuration on Database Table

DDL Triggers

DDL triggers are similar to logon triggers in that they execute events based on some defined audit condition. DDL triggers provide the power to enforce SQL Server auditing by ensuring code execution for defined events and the execution of alerting when required. DDL triggers track specific events within databases. For example, if an SQL administrator wanted an event written every time a database object is deleted a DDL trigger would satisfy this requirement.

DDL triggers help collect audit activity in databases. For example, an SQL administrator always has the option of creating an audit table based on an XML column and inserting any DDL trigger events into the table for simple parsing and reporting later.

A very powerful feature of DDL triggers is the ability to act on CREATE_USER actions and then execute procedures to enforce permissions management. For example, if you wanted to have all users for a database receive data_writer capability on creation the DDL trigger can import the user name from the XML based data and then form an SQL statement to execute the requirement.

Test Tip: When the exam question asks how to audit an event every time it occurs and write an event, the answer should focus on the DDL trigger capability.

C2 Audit Configuration

The C2 audit mode was a previous SQL Server audit posture that matched event logging for both failed and successful attempts to access statements and objects. This mode is easily turned on in SQL Server Management Studio\Security Properties or by using sp_configure. All C2 information is stored in the sys.traces catalog view.

Test Tip: C2 auditing collects a lot of data. If the file location where the logging is saved runs out of disk space then SQL Server will shut down. To bring SQL Server back up you must free up disk space, or for the exam know that you can bring up SQL Server and bypass logging with the following command: Run: sqlservr –f.

Note that the C2 audit standard is now replaced in SQL Server (and the industry) by Common Criteria.

Common Criteria

Common Criteria is an industry security standard certification that is promulgated by the NSA DBMS Protection Profile. SQL Server 2008 is compliant when using Common Criteria compliance as an option to EAL1+.

Common Criteria addresses the following elements. Residual Information Protection (RIP) ensures the overwriting of bits in memory with a known pattern of bits before the memory is used by another process. This prevents buffer and other types of data harvesting from leftover processes in memory. Common Criteria supports the viewing of all login statistics to include successful logins, unsuccessful logins, number of login attempts, and login times. These statistics are viewed in the sys.dm_exec_sessions Dynamic Management View (DMV). The last element of Common Criteria is that any table with DENY will take precedence over any column object of that table with a GRANT.

Common Criteria audit options are enabled via the SQL Server Management Studio or with sp_configure. You must download a script from the Microsoft SQL Server Common Criteria web site to complete the configuration and restart the SQL Service.

Test Tip: When exam questions focus on logging all login attempts both successful and unsuccessful, the answer will include enabling common criteria compliance. When you review the answers to this type of question take note that logon triggers only execute AFTER logons have occurred and only capture successful login activity.

Transparent Data Encryption

Securing SQL Server data with encryption helps further the security model. SQL Server 2008 provides for Transparent Data Encryption or TDE, which provides the required cryptographic functions to encrypt and decrypt data as it is read from or written to SQL storage. This is a huge advantage because it does not require that the application use SQL Server for encrypting and decrypting data. TDE uses a DEK (Database Encryption Key), and this key is stored in the database boot record in case recovery is needed.

Test Tip: Anytime a database using TDE is moved to another SQL instance, the database encryption key requires a backup and restore to the new SQL instance. If you do not also get the certificate, you cannot restore without the key because TDE database backups are also encrypted.

A range of encryption capabilities exist with Windows 2008 and Windows 2003 which have BitLocker Drive Encryption (BDE) and the Encrypting File System (EFS), respectively. This defense layer is only at the operating system level and not within the SQL Server application. SQL Server databases do not achieve encrypted operations by only encrypting the disks with the operating systems. TDE is the answer that provides encryption, and it is preferred because it does not affect applications, such as ASP.NET web applications using a SQL database.

The use of TDE requires underlying key configuration for SQL Server. You will need to create a master key. This is accomplished via the CREATE MASTER KEY statement. The next requirement is a certificate, which like all certificate operations, can be generated from the Active Directory Certificate Authority, the local server certificate authority, or a third party certificate from VeriSign or a similar accredited certificate issuer. The ability to create the required DEK for TDE is contingent upon the above two steps.

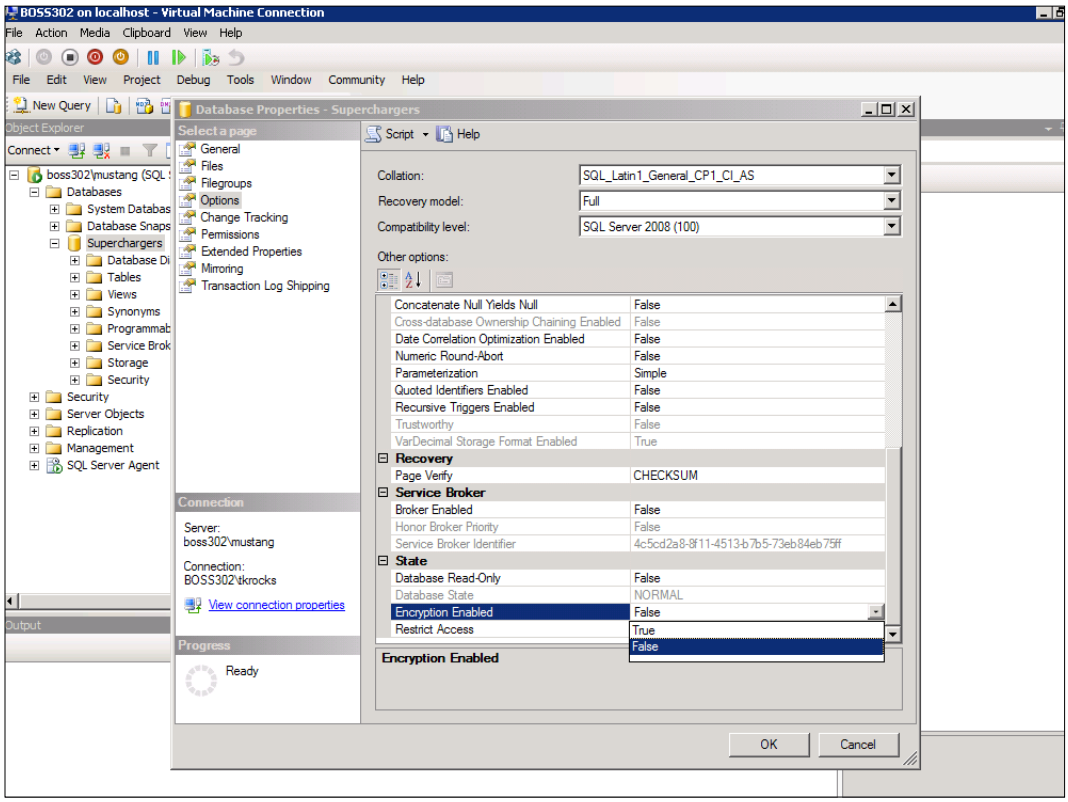


Figure 22: Enabling TDE for a SQL Database

SQL Server Surface Area

The SQL Server surface area is reduced by default to ensure that services that are not necessary after installation are not running. The SQL Server surface area is the portion of the SQL Server services exposed for use and connections via tcp ports or named pipes. An example service is database mirroring, which uses multiple ports to function so it is disabled by default and reduces the SQL Server surface area. The configuration of security for the surface is area is addressed by sp_configure and the previously mentioned SQL Server management policies. A number of configuration facets exist which allow definition of policies relating to the SQL Server surface area. The facet Surface Area Configuration contains the key surface area services for enabling or disabling. The creation of a condition using the facet allows creation of a policy to enable or disable it.

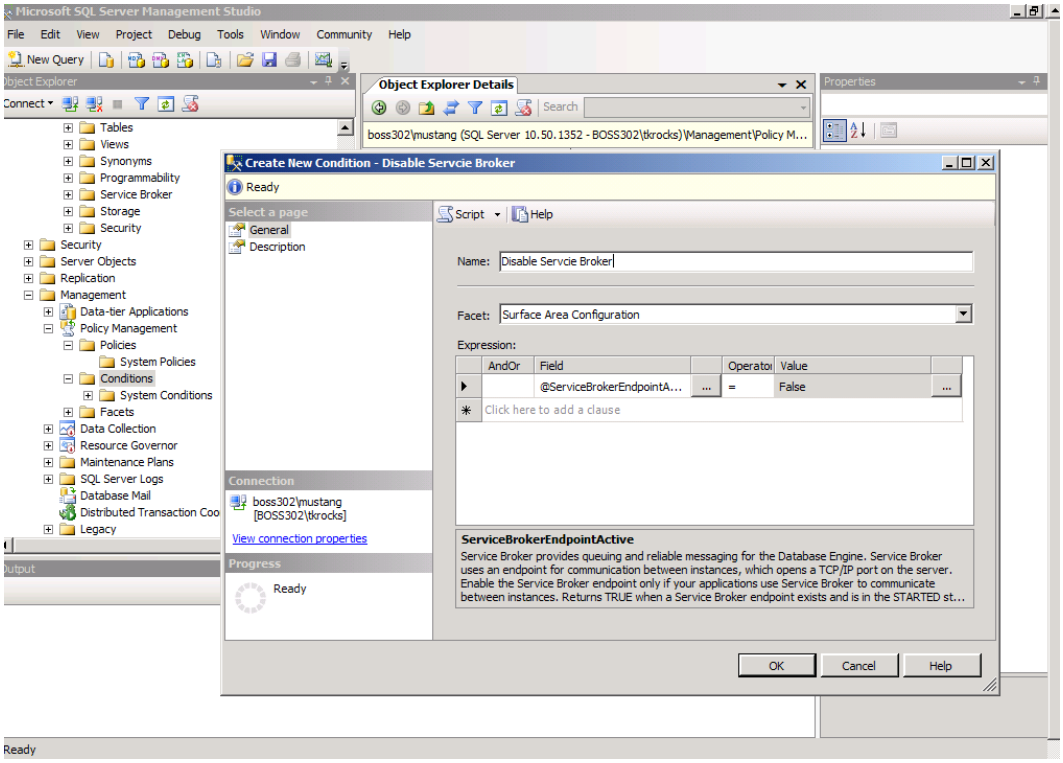


Figure 23: SQL Server policy condition using the Surface Area Configuration facet disabling Service Broker

Once a condition is created based on a preconfigured facet this condition rolls up to a policy on the server, for example, a combined security policy which will include the defined conditions, in this case, those related to surface area configuration.

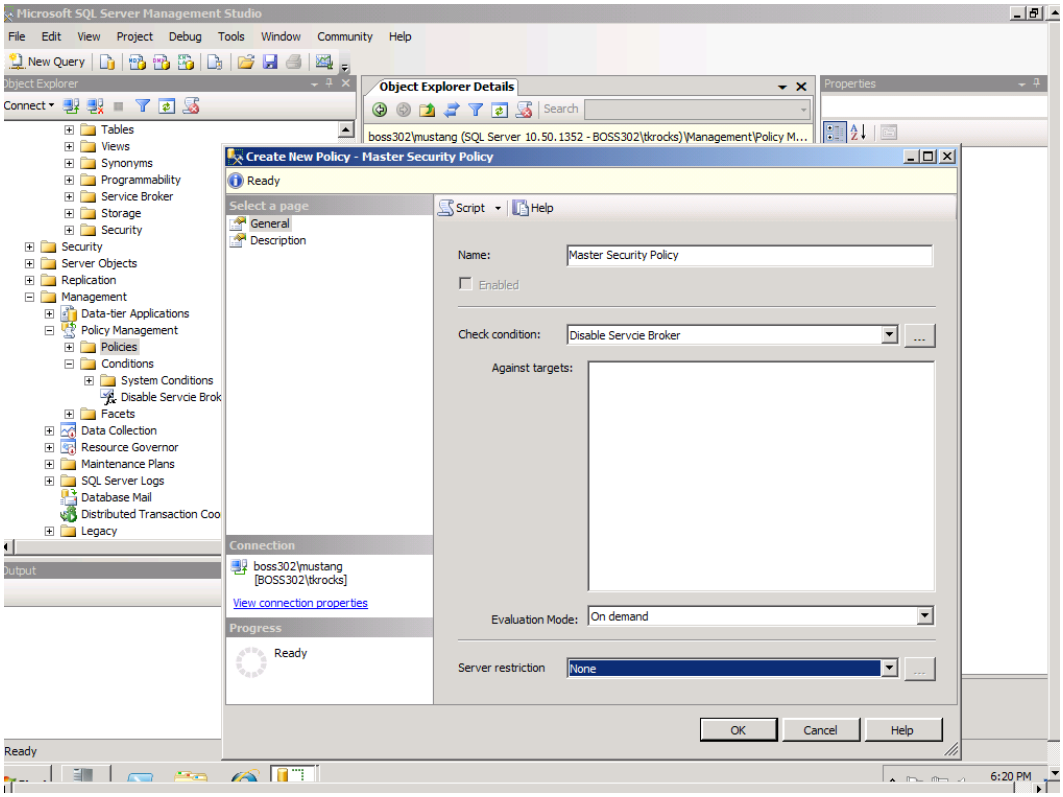


Figure 24: Adding the configured condition to a policy for Surface Area Configuration

Test Tip: In this area, the exam focuses on reusing policies for other SQL Server instances. SQL Server Management Studio allows the export of a facet as a policy to a file. On a new SQL Server instance select the exported policy file, evaluate it, and apply for any violations. Sp_configure is the answer when exam questions ask you to evaluate a surface area solution for one SQL Server instance versus multiple instances.

Domain 4 – Maintaining a SQL Server Database

Database Management and Configuration

Questions on the exam from Domain 4 cover a wide range of everyday SQL Server administrative tasks. These areas include backups, restores, database file types, checking database integrity, recovery models, and ranges of database options.

Database Files

The fundamental building blocks of SQL Server are the collection of files that make up the complete database object. Anytime a database is created two files are created by default. A database consists of a primary database file, with the extension *.mdf, and filegroups within the database object, with the extension *.ndf, and the log file, which has the extension *.ldf. A database object always has a primary file, and then filegroups become secondary. Within each SQL database object are the various elements: tables, indexes, and schemas.

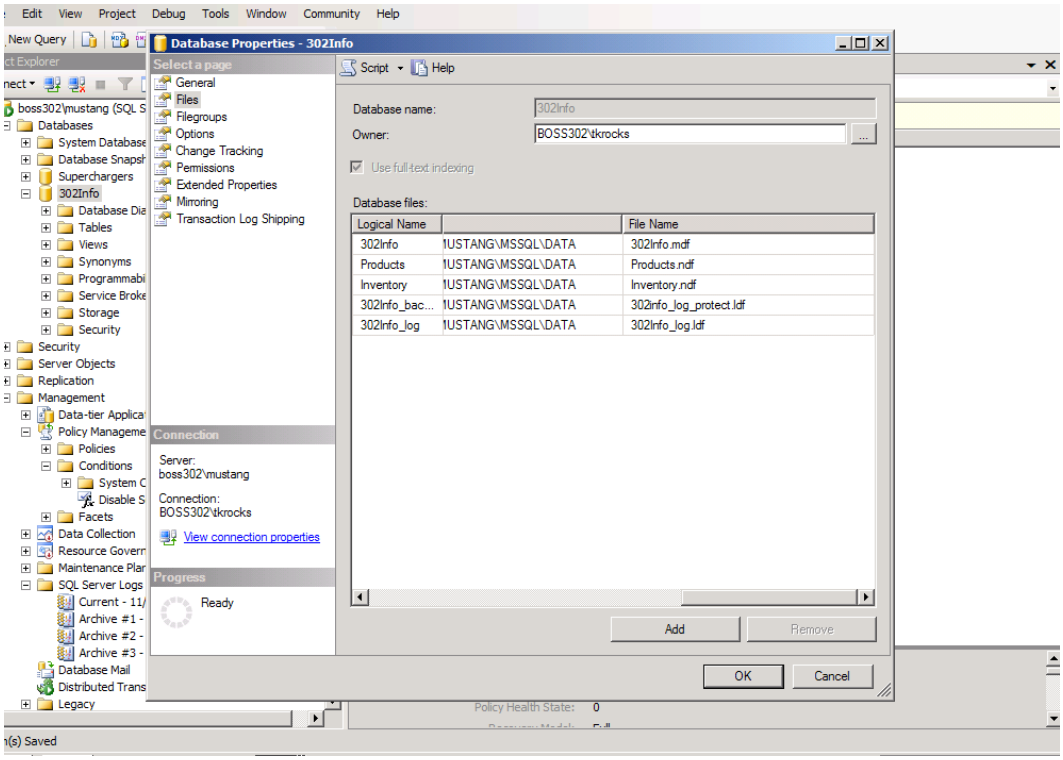


Figure 25: Database Files with extensions, Primary, Filegroups, and Logs

SQL database objects can contain multiple log files. When one log file fills up, the next log file is used. This ensures that the database object stays online and does not suffer downtime.

SQL Server database object files should be stored on separate disks or arrays to achieve fast throughput. The primary *.mdf and *.ndf files should be on different disks or arrays than the *.ldf files. This supports the goal of disk read and write operations employing different I/O paths to different physical disks. Ideally each disk subsystem path will use a different disk controller or Host Bus Adapter (HBA). The SQL Server database filegroup of *.ndf files can also benefit from performance with storage to different disk subsystems. In the exam questions note that RAID 1 is the best disk RAID type to support a log file. This also provides fault tolerance. Performance experts do not agree about the use of large RAID 5 volumes as locations for SQL Server *.mdf and *.ndf files, but this does provide a solid fault tolerance solution. In most cases high transaction based SQL Server and OLAP implementations use RAID 10 (RAID 1 + 0) SAN configurations to achieve the fault tolerance capabilities of RAID 5, with the speed of RAID 1 data reads and writes.

SQL Server 2008 stores data on a page. This is a common term used in backup and restore scenarios because at times the restore problem is to just restore a corrupted page holding some data within the row on that page. Standard page storage is 8 KB. The grouping of 8 pages together in an SQL Server database file structure forms an extent, which is 64 KB. The exam expects an understanding of SQL Server database structures to this level and questions emerge on the smallest elements of a row and table.

Database Filegroups

SQL Server database filegroups build on the common database file structures listed above and expand the options for addressing data segregation, performance, and recoverability.

A filegroup stores multiple SQL Server database files in it. The creation of a database establishes the first file in the group as the PRIMARY. SQL Server administrators in the design process can create multiple filegroup objects which become the *.ndf files listed in the definition of database files. Filegroups allow the spreading of tables and indexes across physical files. Exam questions from the SQL Server Performance section (Domain 7) have filegroups as the basis for achieving great optimization in very large database implementations. You can create multiple filegroups and place them on different volumes and achieve better I/O. When queries execute to various tables spread between filegroups performance is increased.

Filegroups also provide the foundation for good database maintenance and backup planning. Read-only data is placed into a read-only filegroup, which reduces backup time. One strategy is to place large data in different filegroups and then use full backup or differential backup at different times to protect the data. For example, a filegroup named PARTS is backed up fully on Monday and Thursday, and a filegroup named SALES is backed up fully on Tuesday and Friday. This strategy saves time because only portions of the database are backed up and on different days to avoid congestion when backing up large amounts of data.

Test Tip: For exam purposes note that a transaction log can never be in a filegroup!

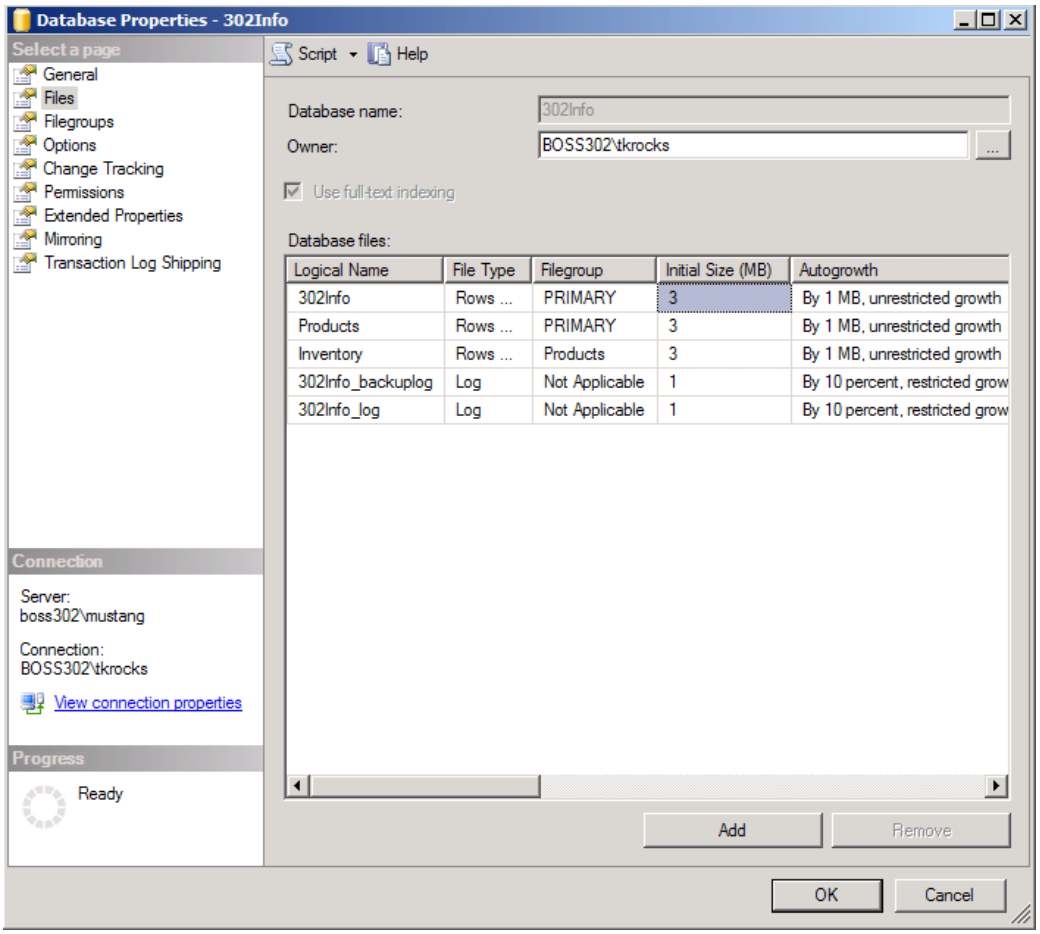


Figure 26: Database with multiple files in a filegroup

Range of Database Options

Each database created in SQL Server has a full set of options available once created. Options can be changed on databases with the SQL Server Management Studio or with the ALTER DATABASE T-SQL statement. The exam will expect complete familiarity with the SQL Server Management Studio database options section and with using ALTER DATABASE.

The ALTER DATABASE command has the following database options available:

- Optionspec
- Auto_option
- Change_tracking_option
- Cursor_option
- Database_mirroring_option
- Date_correlation_optimization_option
- Db_encryption_option
- Db_state_option
- Db_update_option
- Db_user_access_option
- External_access_option
- Parameterization_option
- Recovery_option
- Service_broker_option
- Snapshot_option
- Sql_option
- Termination

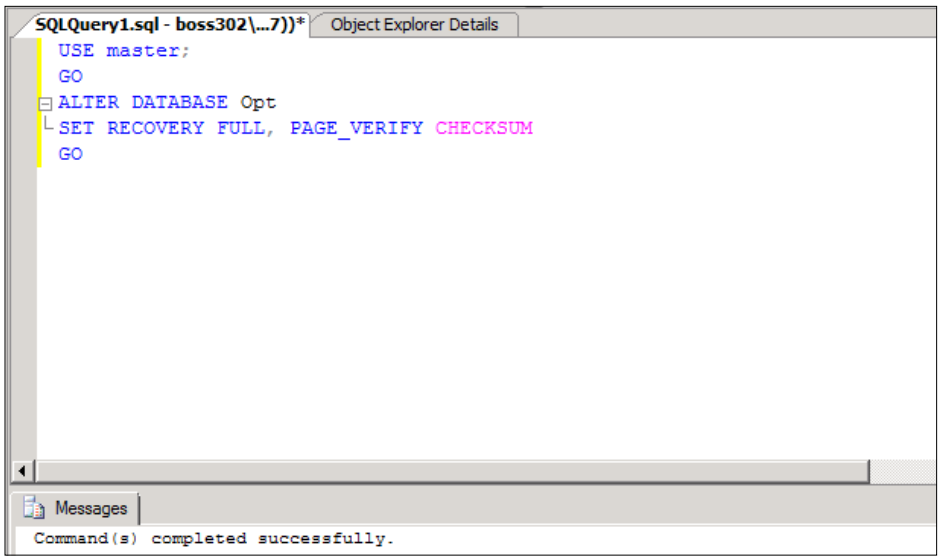


Figure 27: ALTER DATBASE command in action (changing recovery model to full)

Recovery Models

The type of recovery model used in the SQL Server database configuration will drive the restore method available for SQL server databases. SQL Server has three recovery models: full, simple, and bulk-logged. Each of these models has a different effect on how the database's transaction logs are set up. All of the recovery models are set in the Database Properties\Options node. Right-click on a database in SQL Server Management Studio and select database properties.

The simple recovery model dictates that the transaction logs are not kept. When database data is changed and committed the transaction log in simple recovery mode is also cleared, or in SQL Server terms, truncated. When a truncation of the transaction log occurs it is no longer useful in the restore of a database. When a database is in simple recovery mode the only database backup type available is full backup. An SQL Server administrator in support of production databases would not use this recovery model. If a database is backed up each hour over six hours and a failure occurs at 5 hours and 55 minutes, the data loss window is now 55 minutes of data as the administrator would restore the backup taken at hour 5.

The full recovery model does not automatically create SQL checkpoints on the transaction log and truncate them. The full recovery model allows the introduction of two new backup and restore functions: differential and transaction logs. In the full recovery model an administrator can restore one file, without the entire database, or one or more pages and a transaction log. The full recovery model enables full point in time recovery for a database for any time range covered by the backed up transaction logs.

The bulk-logged recovery model is like the full recovery model except that this mode supports bulk operation and is designed to support the full recovery model. This model is a performance-enhancing mode for support of intensive database operations such as bulk imports of data or index creation. Bulk log will reduce the log space required. However, one key difference from the full recovery model is that the bulk-logged model does not support point in time recovery!

Exam Prep: The 302Info database in Simple Recovery Mode
The 302Info database is configured with the simple recovery database option. Each night the 302Info database is backed up at 0200. One day you come to work and the Sales department reports that their .NET applications are reporting the error "Database 302Info Not Found." You check the SQL Server instance and discover a failed drive. The drive failure occurred over 10 hours after the last backup, and the company had conducted about 10,000 sales between the last backup and the drive failure. You are new to the job and not held responsible for this catastrophe. However, the Sales Executive Vice President tells you:

- Never allow the loss of more than 10 minutes of sales data.
- The company does not have much available storage space for backups.

What solution will you provide?

Solution: The 302Info database in Simple Recovery Mode
The first part of the solution is immediately apparent: the 302Info database must switch to the full recovery model. This is accomplished easily through SQL Server Management Studio and database options for 302Info. This change will not disrupt current database operations and is seamless to users. The second big requirement is to not lose more than 10 minutes of sales data. This requirement suggests scheduling a transaction log backup every 10 minutes.

The last requirement is that the recovery method cannot use much disk space. Transaction log backups at such a short interval (10 minutes) will use a minimal amount of space since they only capture what has changed in that time span.

This is another graduated walkthrough of a question type that may appear on the exam.
Keep the following in mind when answering these types of questions:

- Some of the answers will offer the differential backup as an option. This option, however, uses much disk space and backs up everything changed.
- Answers will also offer different time intervals. Always ensure you match your answer with the solution. In this case the operative timeframe is 10 minutes.

Detaching and Attaching Databases

The SQL Server Management Studio supports the easy detaching and attaching of SQL databases. This function is primarily used to move a database to a new server. A database detach will ensure that no processes or applications have the database in use and ensure that the files can be easily copied and move. Once a database is detached navigate to the file system directory containing the databasename.mdf (database file) and the databasenamelog.ldf and copy them to the new SQL Server drive location.

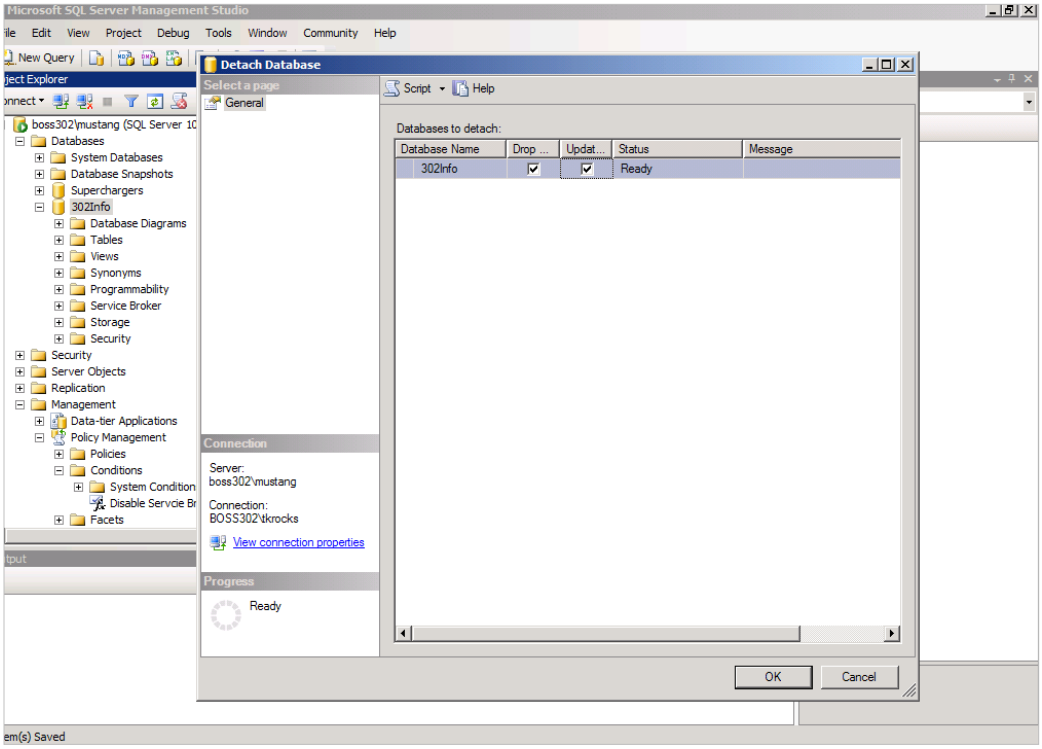


Figure 28: Detaching a SQL Server Database

Database Backups

Database backups and restores occupy a large portion of an SQL administrator's time during support or production database operations. This area is also crucial to successfully passing MS Exam 70-432. Domain 4 adds to the basics of backups and restores covered in Domain 2. In this section we will investigate more scenarios for backup and restore questions versus core definitions and how to set up and conduct these services.

Backup Type Details

When you are ready to take the MS 70-432 Exam, you should already have a solid understanding of backup and restore definitions.

Exam Prep: Backup and Restore for 302Info
You are responsible for a database called 302Info. The database is approximately 460 GB, and data exists in multiple filegroups. This full backup takes 6 hours. Only 30% of the data in the 302Info database changes daily. The Parts filegroup is read-only and rarely updates.

The business division, which uses the 302Info database for corporate sales, has requested that you devise a backup implementation that achieves the following business goals:

- No more than 2 hours of data can be lost at any time.
- The time window to back up the database, which is currently at 6 hours, is too long and must be reduced.
- If any database recovery is required the business team demands that it be as quick as possible.

Filegroups for db.302Info	Filegroup Types
PowerAccessoriesSales	Primary (150 GB)
Parts	Read-Only (200 GB)
CustomerInfo	Read-Write (110 GB)

Solution: Backup and Restore for 302Info
The scenario above will have four possible answers on the exam, but in evaluating this question, several backup concepts emerge quickly as clues to the correct answer.

The use of a read-only filegroup called Parts immediately signals that the backup option READ_WRITE_FILEGROUPS is okay to use for a partial backup, i.e., the backup command will only backup read-write filegroups and the primary filegroup. By using the T-SQL statement BACKUP DATBASE 302Info READ_WRITE_FILEGROUPS TO CompanyBackupSAN, you achieve a backup of the changed data each day, but now avoid backing up over 200 GB of static data. This will reduce the backup time to less than 3 hours, which is a 100% reduction in time. When data changes in the Parts filegroup, then schedule a backup to accommodate the new data and then move forward again with the READ_WRITE_FILEGROUPS option. This is a partial backup type.

The next portion of the question focuses on how much time can exist between data failures. So if a failure of the database occurs at 1410, and it is backed up every two hours with a differential option, then recovery can occur within the expected two-hour time window. The solution is to set up a BACKUP WITH DIFFERENTIAL every two hours for the 302Info database. This will ensure that no more than two hours of data is at risk throughout the day. For a restore, the fact that the full backup did not back up the 200 GB read-only filegroup along with the smaller size of the differential backups ensure that you will enjoy a faster recovery time.

The combination of the above two solutions allows you to quickly isolate the answer on the exam. Make sure you focus on answers with solutions that do not work so that you can quickly identify the correct answer. Examples of solutions that don't work for the above scenario are as follows:

- BACKUP DATABASE 302Info to CompanyBackupSAN. This solution maintains the current situation of a long backup timeframe.
- BACKUP DATABASE 302Info to CompanyBackupSAN WITH DIFFERENTIAL, and do this nightly. This solution fails because it does not create the base backup required for recovery and as the partial backup base.
- Backing up the databases every four hours without using a differential, which supports a quick restore, or not backing up the databases within the requested data loss prevention interval, which in this scenario is two hours.

Test Tip: The above scenario demonstrates a method you can use to evaluate the requirements of a question on the exam and how to eliminate wrong answers and isolate the correct one.

Verifying Backups

Verifying backups in SQL Server is an important administrator task and one that should happen often to ensure that when disaster strikes you are ready. The exam will expect you to know how to verify a backup during the backup process and also manually.

The Verify backup option is available when setting up a backup job through the SQL Server Maintenance Plan interface. This option will ensure that the backup is complete and the volumes are readable.

Another verification method is the option of performing checksum before writing to media. The backup process will check media but continue on when it finds an error, but not use that portion of the backup volume.

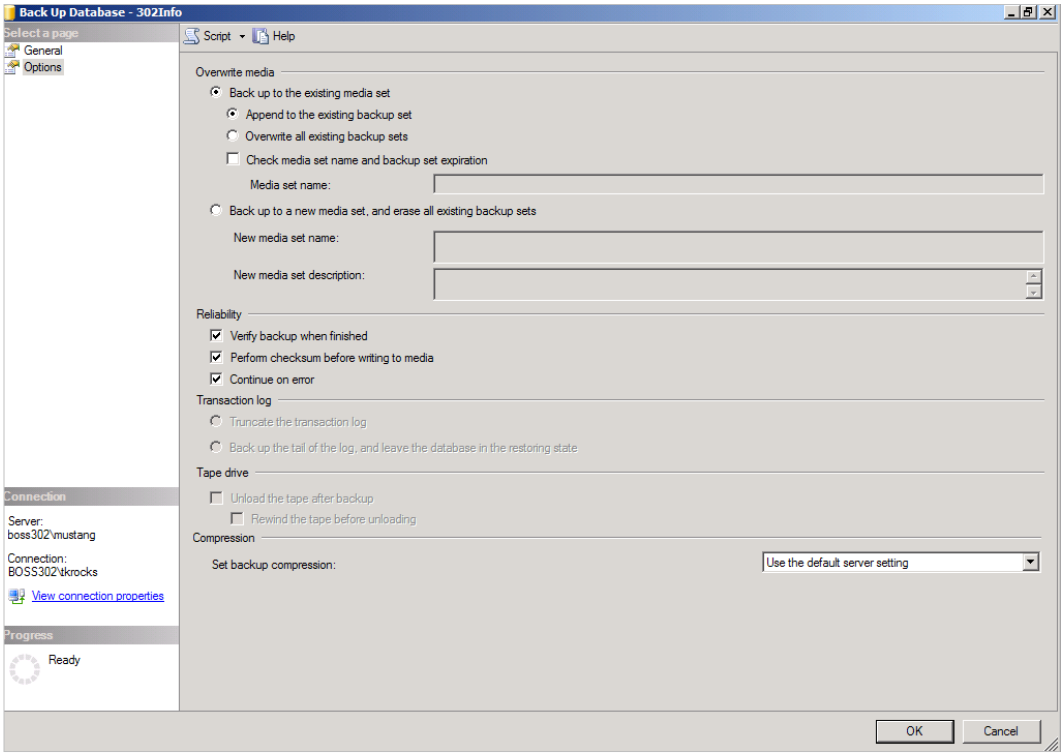


Figure 29: SQL Server Backup options for reliability and verification

Restoring Databases

The restore process for SQL server databases takes several routes. Keep in mind for the exam that the SQL Server database recovery model dictates a route for the restore operation. If the recovery model is set to simple, then a transaction log restore is not possible because it was not an option in the backup configuration.

SQL administrators will excel on the exam by understanding the following restore methods, and this will translate to successfully supporting corporate SQL database operations. The first time you conduct a restore from well managed backup options that save a company money and time you will be the hero!

Restore Types and Strategies

The following are restore strategies with an explanation of how the recovery model of the database influences them:

- Restore of the entire database.
    - Simple recovery model: Restores the entire database with no transaction log capability.
    - Full recovery model: Restores the entire database, and allows restore of any subsequent log files to the full backup. As with all full recovery model restores the last restore of a log is restored with the RESTORE WITH RECOVERY option.
- Restore of a data file, which can include different filegroup database files and read-only files.
    - Simple recovery model: Restore individual files only if the database has at least one read-only filegroup.
    - Full recovery model: Restore individual files, without restoring the entire database. Filegroups being restored are always offline.
- Restore of a Page, which can correct a damaged page or pages.
    - Simple recovery model: Not an option.
    - Full recovery model: Page restoration relies on having an unbroken chain of log backups to the current log file.
- Piecemeal restores recover databases in stages at the filegroup level, starting with the primary and working through the remaining read/write or secondary filegroups.
    - Simple recovery model: supported.
    - Full recovery model: supported.

Exam Prep: Database 302info Restore

The 302info database is 400 GB total and has 300 GB of database files located on SAN drive Y and 100 GB of log files located on SAN drive L. You have implemented the following backup schedule:

Backup Type	Backup Time
Full Backup	2300 Nightly
Differential Backup	1000 Daily
Differential Backup	1400 Daily
Transaction Log Backup	Every 30 Minutes, Daily

At 1310 PM on a Tuesday, the SAN array hosting the database file abends (abnormal end to operations) and is offline. How will you quickly recovery the database?

Solution: Restore Scenario
On the exam quickly identify your options for the restore. In this case you must note the complete loss of the SAN drive supporting the database file. The SAN drive supporting the log file is still up and working, however, this is a scenario that requires a restore to get the database back.

Keep the following steps in mind when performing a restore:

1. First and foremost – back up the transaction log!
2. Restore the last full backup.
3. Restore the latest differential backup if applicable.
4. Restore each log file backup taken after the last differential backup in sequence, with the RECOVERY option selected, until restore of the last transaction log.

When thinking about the solution above, using these steps you can successfully restore the database back to its state at 1300 or a data loss window of 10 minutes.

The backup of the transaction log is detailed below in the Transaction Log Tail Restore section. This last transaction log backup prior to the restore allows the recovery process to even restore some of the transactions between 1300 and the time of the failure at 1310. There is a differential backup conducted at 1000 daily so this will reduce the amount of time for the restore due to not having to restore logs from 2300 the night before.

Rule out answers that do not back up the log as the first step of the restore process. Beware of answers that suggest that a restore of the latest transaction log is the only requirement for logs. Also note answers that restore transactions logs from the original full backup time and do not incorporate the differential backup.

Database Snapshots

Database snapshots provide another level of recoverability and fault tolerance in protecting SQL Server. Microsoft's database snapshot technology builds on their expertise with virtual drive snapshots for Windows operating system products, but adds the important SQL unique snapshot requirements of ensuring that the database snapshot is readable by SQL Server. A snapshot is a point in time picture of the state of a database and all of its objects. Database snapshots are extremely useful for code updates that affect table structures or other database stored procedures. When a code update goes awry or some new process removes a table or index by mistake, the reversion back to the previous database snapshot saves time from a restore and ensures rapid recovery capability.

Test Tip: There is no capability within the SQL Server Management Studio graphical tools to create a database snapshot.

The database snapshot is read-only. However the contents of the created snapshot are usable with T-SQL commands for restore back into the source database.

Snapshots are created with the CREATE DATABASE command. An example is shown below.

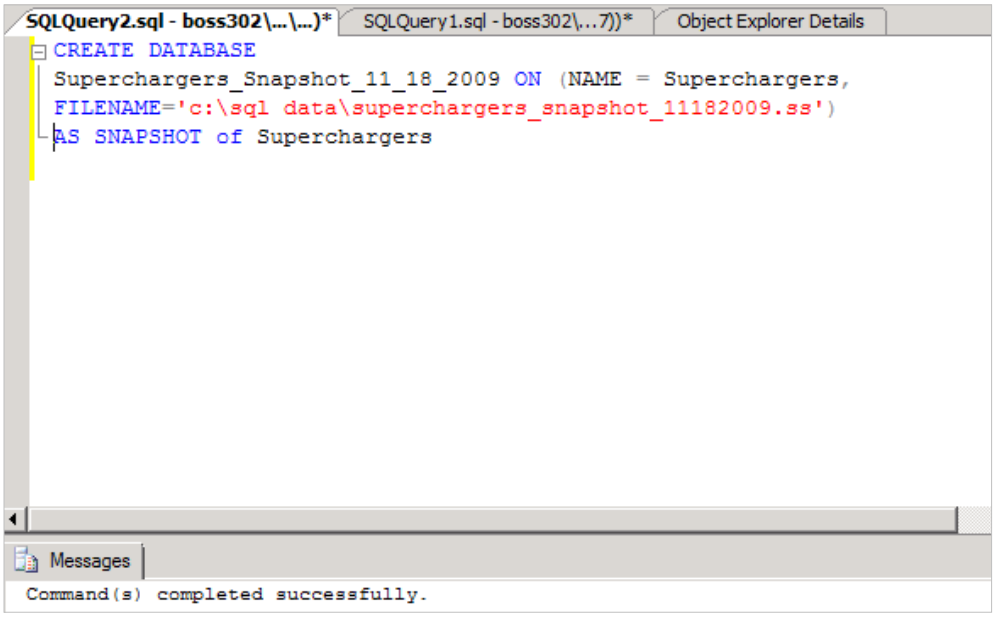


Figure 30: CREATE DATABASE command for a database snapshot

The above snapshot command created a copy of the database Superchargers. When any data changes in the database Superchargers, that data will show up in the created snapshot. The snapshot has a utility for restoring corrupted data or any of the objects in the database that are corrupt. Database snapshots receive data from the primary database via copy-on-write, which is how the snapshot stays up to date.

Test Tip: Database snapshots will grow in time just like the database in which they are taken. Database snapshots must always exist on the same SQL instance as the source database.

Database Integrity

Database integrity refers to the usability of SQL Server databases. Are they fit? Are they corrupted? The primary tool in your SQL administrative tool kit for analyzing database integrity is the DBCC command. Common SQL Server challenges for databases include torn pages that affect performance and data retrieval. Abrupt power outages can contribute to database damage. The DBCC command and its options allow a range of checks and repair activity.

DBCC CHECKDB

Database Console Commands or DBCC are used to conduct maintenance and integrity checks on databases. DBCC has a set of statements related to validation, which help ensure database integrity. The primary statement used for validation is DBCC CHECKDB. This check will ensure correct allocation and structural integrity of all objects in a database. It will identify and repair a wide range of possible errors.

Exam Prep: Using DBCC CHECKDB
You are responsible for a database named 302Info. This database is on an SQL Server 2008 instance. 302Info has six filegroups, and tables located over several filegroups. A problem has occurred and a developer has indicated a check of the database's integrity is needed as quickly as possible. What will you do?

Solution: Using DBCC CHECKDB
This question is like many posed in the exam in that some of the data given in the question is not relevant. The DBCC CHECKDB command has existed since SQL Server 2000, so the version of SQL Server is irrelevant. The filegroup locations in the question are also irrelevant since they won't affect the DBCC CHECKDB command.

The quickest answer to this question is to run DBCC CHECKDB 302Info. This will provide a quick check to verify the integrity of the database.

As you work through the exam questions some will be just this easy. Don't second-guess yourself when determining the answers as this type of question can lead to overthinking. DBCC CHECKDB has many options. You can use the WITH PHYSICAL_ONLY switch, which will only check and not repair any problems.

An exam answer for this solution using the REPAIR_FAST option is a quick clue not to use it. This option only applies to other versions of SQL Server, not SQL Server 2008.

Database Maintenance Plans

This domain focuses extensively on database files, options, backups, restores, and methods to ensure databases are available for use. The Database Maintenance Plan capability of SQL Server 2008 with both a manual and wizard driven creation process is a feature that SQL administrators will use often. Maintenance plans can set up backups of all types, index reorganization tasks, database compression, and perform other required daily tasks.

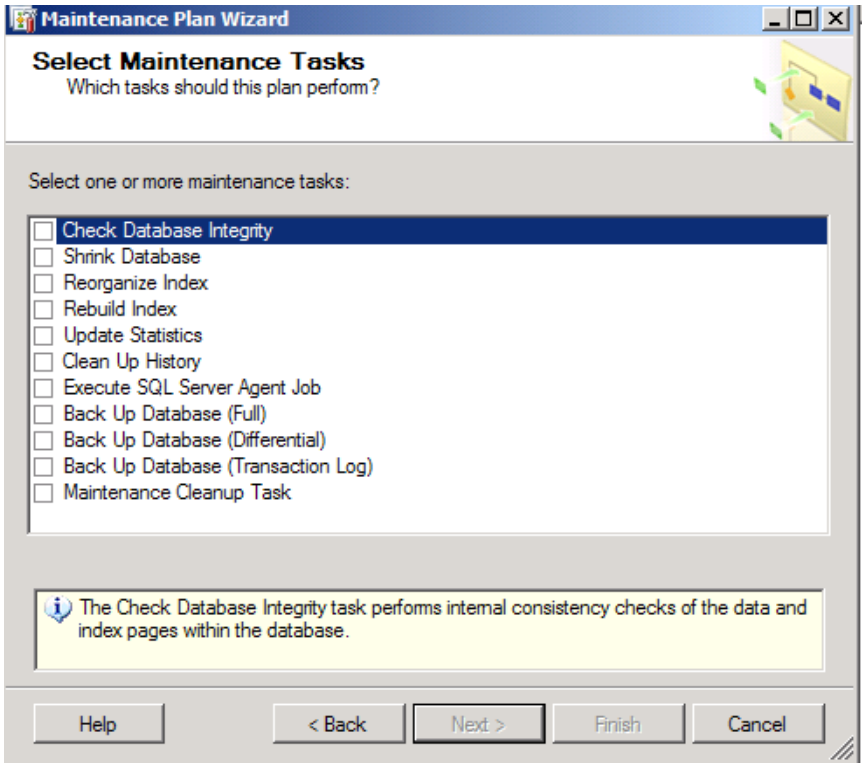


Figure 31: Preconfigured Maintenance Tasks

Maintenance Plans are built from the Maintenance Plans node in SQL Server Management Studio. The wizard supports rapid point and click creation of maintenance plans with all required options, which include time, schedule, databases to affect, types of backups, and notifications via text files and email reports. The large number of predefined maintenance tasks facilitates the quick build of multi-step plans to allow individual step schedules.

The Maintenance Plan designer allows the manual creation of database maintenance plans and editing of created plans.

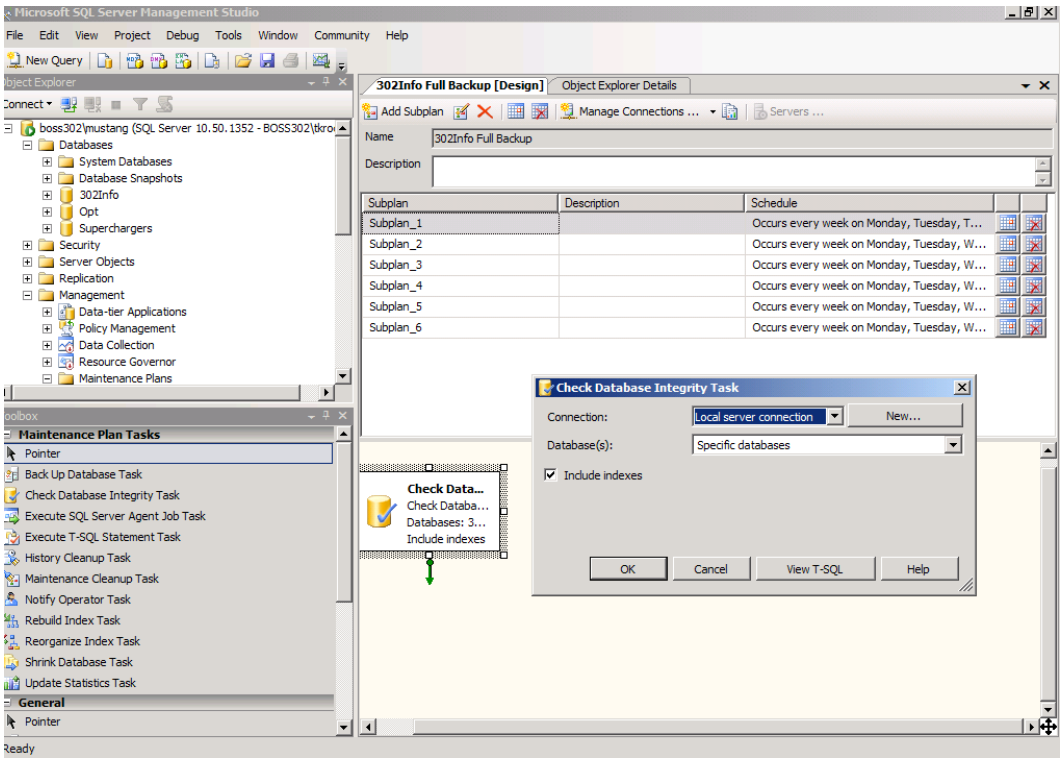


Figure 32: SQL Server Maintenance Plan Designer

Test Tip: When exam questions pose the task of conducting multiple actions with the **least administrative effort**, this points to using a SQL Server Maintenance Plan. Only members of the sysadmin role can create and modify maintenance plans.

Domain 5 – Data Management Tasks

The manipulation of data within SQL Server is what the database product is all about. The storage and retrieval of data is paramount to supporting applications and other information systems. The exam focuses heavily on these skills and as an SQL administrator so will your day-to-day activities.

Data Export and Import

SQL Server provides multiple methods to export and import data. We will review Bulk Copy Process (BPC), BULK INSERT, OPENROWNSET, and the extensive set of tools available within the SQL Server Management Studio. SQL has a dedicated service—SQL Server Integration Services (SSIS)—designed to address the export and import of data with GUI-based support and SQL T-SQL statements. Many of the scenarios on the exam and on the job include activities that require movement of data from SQL Server 2005 (and possibly SQL Server 2000) instances to new SQL Server 2008 instances. Data import and export can be part of established business processes or a single task. Data movement also focuses on hydrating SQL Server Reporting Services databases, data warehouses, and other analytical operations.

The first consideration for exporting and importing data is preparing the data. Whether using the BCP utility, BULK INSERT, or OPENROWSET, the data requires proper formatting for either import or export. The following conditions apply for data preparation for export and import:

Export

- Never use a name for a BCP export if the data file name is already in use. This will OVERWRITE the current exported data in the file.
- The SELECT permission is a minimum security requirement to export data.
- When exporting data, SQL Server 2008 will use parallel scans for export retrieval, i.e., data is not retrieved in any specific order. When the exam mentions specific order for an export, always use the **queryout** option explained below, with an ORDER BY switch.

Import

- The file must be in row and column configuration.
- Character data and native binary format is supported, including Unicode.
- The destination table for the data import must already exist.
- The target table must have a corresponding column with the imported data.
- Use XML formats to import fixed length fields with a format file.
- CSV files are not supported for BULK IMPORT operations, however, if data fields do not contain a field ender (the comma normally in a CSV file), or none of the data is in quotation marks, then CSV-based data may import.

Methods of Data Import and Export

One of the most common data import and export activities is copying data between servers. The BCP utility, BULK INSERT T-SQL commands, and OPENROWSET all play a role.

BCP

The BCP command can export or import data from a SQL Database object. The BCP utility is command line based, and accessed via bcp.exe. This utility is beneficial in that it runs on a small memory space and is fast. When exam questions focus on quick "out of band" SQL Server data import or export--think BCP! It can be accessed via the Windows command line, in script, or with SQL Server Powershell.

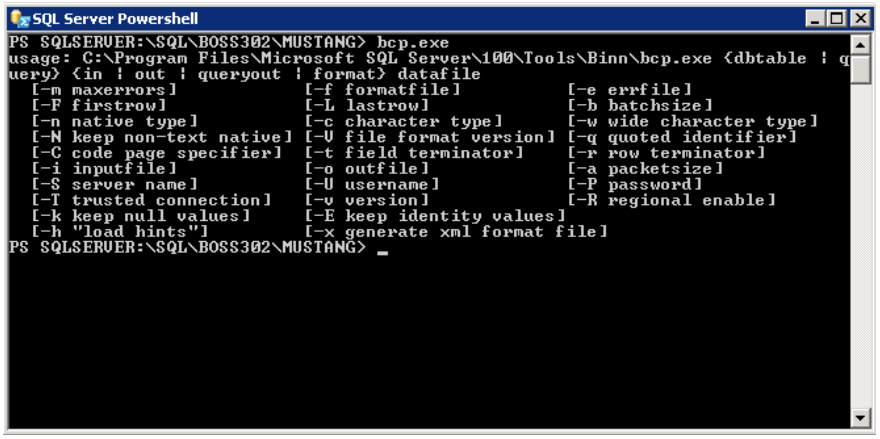


Figure 33: BCP command line with switches

The BCP utility requires an exact order of commands and is case sensitive. The BCP utility also requires a format file to set options for the data import or export.

The most commonly used syntaxes for BCP are listed below:

BCP Setting	Explanation
-a packetsize	Number of bytes in each packet for import/export, default is 4096 bytes.
-b batchsize	Number of rows to copy, default is ALL.
-c	Specifies the use of character mode for data transfers. This supports a non-SQL destination and will output data in ASCII text.
-C codepage	Sets the relevant codepage to use in the transfer (does not matter when character values are > 127 or < 32).
-e errfile	File to store error messages.
-E	Uses identify values.
-F firstrow	Determines the number to use for first row.
-f formatfile	A very important component of any BCP command syntax. The format file is a file with an extension *.fmt and contains key information for the import or export job.
-h loadhints	Hints consist of TRIGGERS during the job, ROWS_PER_BATCH, or CHECK_ CONSTAINTS, and ORDER.
-I inputfile	Used in interactive mode for responses to command prompt questions.
-k	Keeps nulls.
-L lastrow	Will set the last row.
-m maxerrors	Determines the number of errors which can occur before stopping the BCP.
-N	Determines native export for non-character data and Unicode.
-n	SQL native mode, used in transferring data between SQL Server instances or databases.
-o outfile	File for BCP output during scripted BCP use.
-P password	Password for login during BCP.
-r rowterminator	Determines when to terminate a row.
-S servername	The SQL Server name or instance.
-T	Allows use of a trusted connection.
-U username	Login name during BCP.
-w	Sets wide character use (Unicode) for imports and exports.
-x	Supports the creation of an XML file instead of a standard file as listed in the –f switch.

Table 5: BCP Settings

A sample export command with BCP using a trusted connection is listed below. The data exported is headers information from a table called Parts within the 302info database. The –T switch ensures a trusted connection at the command line.

```
bcp 302info.parts.headers out headers.dat -T
```

Exam Prep: Using BCP
You manage several databases, one of which is called Accounts. This database has several tables in which customers require the data monthly, and the data is very large. When exporting this data using SQL Server based SSIS packages, you notice it takes a very long time and affects the capacity of your SQL Server during the export. Since your operations are 24/7, any latency inducing operations are not good. What actions should you take?

Solution: Using BCP
The answers on the exam will focus on many methods for exporting and importing and appear to always mix in an XML file to help control data that is not formatted correctly. Other web service terms such as XSLT and XSD do not APPLY to data import and export with BCP. The focus must be on time and efficiency, which BCP adheres to since it runs in separate memory space, has the capability of running queries for data, and is fast. As you evaluate the possible answers remember that BCP is always good for offsite movement of data because its modes can support character-based export since the target system may not be a SQL Server. Never forget that BULK INSERT is as advertised; it can only import data. BCP is very scriptable in Cscript, Powershell, and SQL Agent jobs.

BULK INSERT

The BULK INSERT command is very similar in syntax to BCP but only addresses import, not export of data. Note for the exam that BULK INSERT capability is related to setting the BULK-LOGGED recovery model. Use this recovery model during BULK INSERT to speed up the process with less transaction log writes. Database recovery models are flexible and can shift between BULK-LOGGED and FULL with no issues. A sample BULK INSERT is listed below where table data is inserted into the Parts table object of the database headers from a file named lineitem.tbl, with a field terminator switch and row terminator switch.

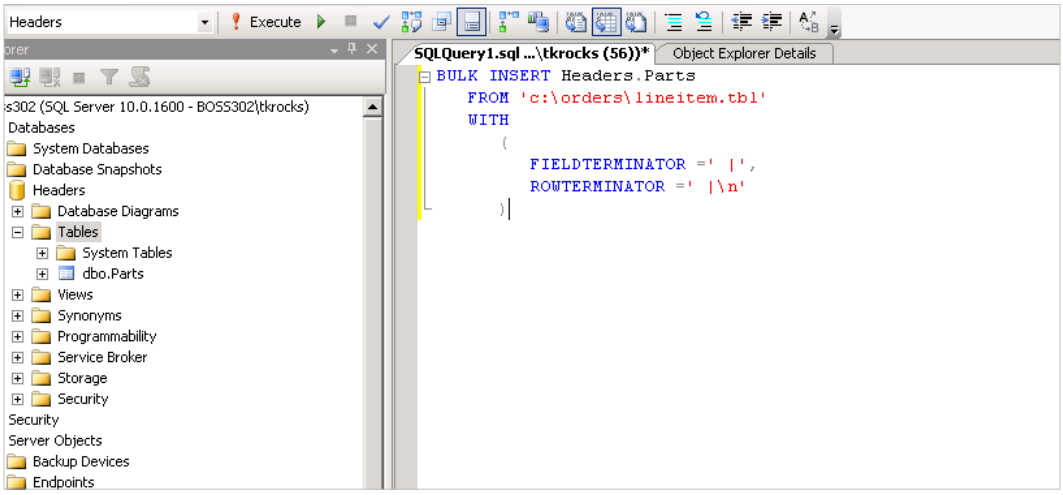


Figure 34: BULK INSERT Command Example

The advantages of BULK INSERT are that it is great for importing large amounts of data, and the T-SQL syntax switches are very similar to BCP. BULK INSERT has a corresponding recovery model to enhance the insertion of very large amounts of data: BULK-LOGGED. This command also works very well with constraints that include NULL values, clustered indexes, and foreign key constraints.

OPENROWSET

OPENROWSET provides another SQL Server 2008 method for data import. OPENROWSET is a function that is used among SQL Servers for one-time, ad hoc data access. It is an OLE DB function called with the FROM statement, or INSERT, UPDATE, and DELETE T-SQL commands. OPENROWSET also can read bulk data from a file and then return it as a row set. This command allows rapid return of a row from a source and insertion of the data into the destination.

The following example demonstrates the use of OPENROWSET with a SELECT statement and SQL Server OLE DB as the provider. Different data providers may be invoked as required. The provider is listed first, then the connection information, followed by the SELECT syntax, the FROM syntax, and additional commands as required. One implementation note for using OPENROWSET is that Ad Hoc Distributed Queries must be configured to on, or a state of 1. This is disabled by default.

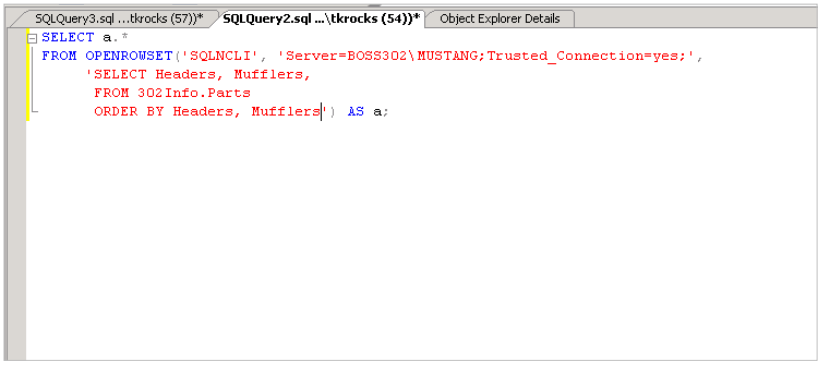


Figure 35: Example of OPENROWSET Data Import

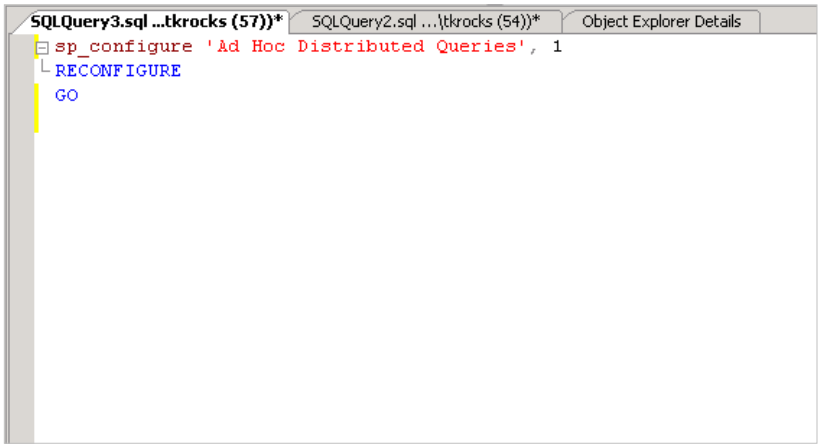


Figure 36: Enabling Ad Hoc Distributed Queries

Exam Prep: Using OPENROWSET
You manage a database named 302info on a SQL Server 2008 instance. The sales team requires a data import from another server running SQL Server 2005 on an as needed basis. You set up an OPENROWSET function to complete the task but each day it fails. How should you troubleshoot this?

Solution: Using OPENROWSET
This type of exam question requires a quick examination of the answers for why OPENROWSET is failing. The first area to check in SQL Server 2008 is the state of Ad Hoc Distributed queries, which is turned off by default. Make sure you enable this. The answers will take on multiple paths. One path is almost always the use of SQL Server Integration Services (SSIS), but in this limited ad hoc case, the solution is an OPENROWSET one. SSIS provides a much more extensive bulk data import and export transformation capability.

SQL Server Management Studio GUI Data Import and Export Capability

SQL Server 2008 provides a rich point and click wizard to set up data import and export packages via tasks established in the SQL Server Management Studio. The graphical import and export wizard creates packages that can import and export data in an ad hoc as required manner or as a scheduled job. The SQL Server import and export wizard created packages can be managed by the SQL Server Business Intelligence Development Studio and are SSIS packages. This wizard is accessed by right-clicking a database in SQL Server Management Studio and selecting Import or Export under Tasks.

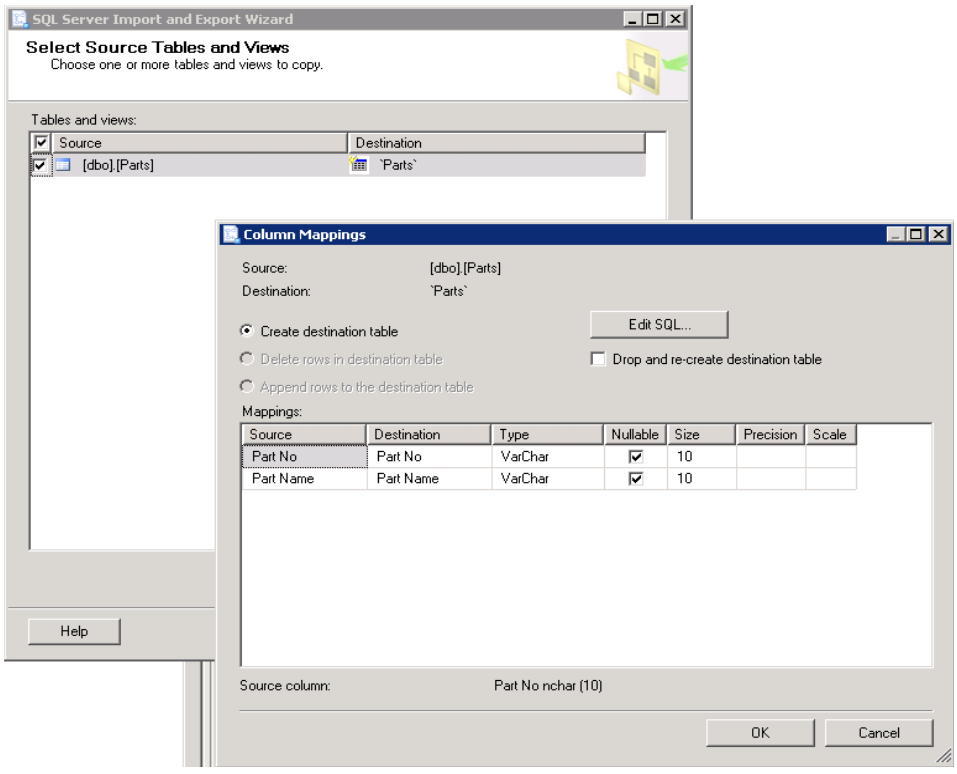


Figure 37: SQL Server GUI Data Import and Export Wizard

The point and click wizard allows setting all package elements for the data import or export activity. Key settings created in the wizard include:

- Data Source (SQL Server, OLE DB, Microsoft Excel, or Microsoft Access)
- Authentication Methods
- Destination Server
- Data copy options to include one or more tables or views, or a query to establish content
- Table Mappings
- Preview of Results
- Destination Options to include deletion of rows, or appending
- The ability to run the package immediately or save it for future use as an SSIS package
- Ability to send exported data to multiple formats to include flat files, Microsoft Excel, and Microsoft Access formats

Test Tip: Exam questions that describe regularly scheduled table data copies and import or export of data from Microsoft Excel and Microsoft Access to SQL Server are indicating the use of SSIS.

Data Partitions

SQL Server 2008 tables and indexes can exist in more than one partition. This is termed horizontal data partitioning because groups of rows are organized into individual partitions based on an indentifed column. This type of table organization is critical to large database efficiency and performance. Partitioning considerations include if the table will host data used in different ways, and any query or other database operations that are latent or not executing properly. Additionally, you can partition a table using the concept of vertical partitions, which is spreading the columns of a table across multiple tables versus the rows.

If a large table for auto parts exists called PARTS, and the primary database operations against it are INSERT or DELETE (some of the data is only used for SELECT queries), a partition could occur on the PARTS types, and isolate the items used in SELECT queries from the ones used in INSERT and DELETE functions. Another good example would be an order table, in which the current month's data was used primarily for INSERT or DELETE operations. The previous month's data, which is mostly used for SELECT queries, represents a target for partitioned tables. This ensures database functions do not scan or query an entire large table.

The following commands support the partition function: CREATE PARTITION FUNCTION, ALTER PARTITION FUNCTION, and DROP PARTITION FUNCTION. The exam focuses on an understanding of these commands and when to use them to manage a data partition. Each new partition must have a range called the split range, and the values must be unique from the existing table. If a partition function takes in the year 2006 for orders, then the new range must start with a different range, 2007, for example. The merging of data into the new partition is accomplished once a range is created.

The steps for creating a table partition are:

1. Create or use an existing partition function that sets the correct range boundaries (CREATE PARTITION FUNCTION).
2. Create or use an existing partition scheme tied to the partition function.
3. Create the table using the partition scheme.

This sample function demonstrates the creation of four partitions for each year with an int column.

```
CREATE PARTITION FUNCTION myDateRangeFP1 (int)
AS RANGE LEFT FOR VALUES (2006, 2007, and 2008)
```

This sample function demonstrates the alteration of an existing partitioned table in which orders received in 2009 would split out.

```
ALTER PARTITION FUNCTION myDateRangeFP1 ()
SPLIT RANGE (2009)
```

A data partition is easily dropped with the following command:

```
DROP PARTITION FUNCTION myDateRangeFP1
```

Exam Prep: Altering a Data Partition
You manage a large database named PARTS hosted on SQL Server 2008. The PARTS database has a table called INVENTORY, which is partitioned on its ReceivedDate column using the PFReceivedDate function. This partition uses one month values.

You need to write a script that moves the older data for more than one year to the partitioned table for PFRecivedDate into an empty partition. Which code should you use?

Exam Solution: Alerting a Data Partition
This example alerts you to use the ALTER PARTITION FUNCTION with appropriate syntax and switches. You will run ALTER PARTITION FUNCTION PFReceivedDate() SPLIT RANGE(datelastyear). This will split out the data from last year. Then you can merge the data into the empty partition, and add to it as needed.

Test Tip: The partitioned table feature is only available in SQL Server 2008 Enterprise and Developer versions, not Standard.

Data Compression

SQL Server 2008 has several methods for data compression within the database object itself. The exam expects an understanding of when to employ data compression based on the situation and how it will affect the parameters of the database object.

SQL Server 2008 row compression allows the storage of fixed length data in variable length format, and page compression attacks data redundancy.

SQL Server 2008, unlike SQL Server 2005, extends the capability of row compression to all fixed length data types, to include integer, char, and float data types. This type of support allows data compression without the application using the data requiring any type of change to access it. The goal is to reduce byte storage consumption. If a common integer can fit in 1 byte, there is no reason to use 4 bytes to store it. By using variable length storage format 3 bytes are saved. Variable characters are now a standard option on column definitions.

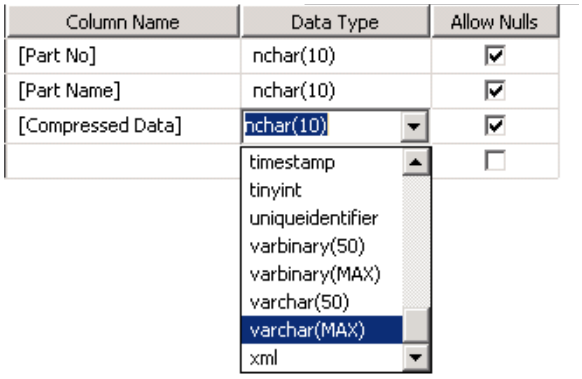


Figure 38: Example so variable character data types

Page compression is the first place to start when ensuring data within a SQL Server database is compressed and uses less space. Page compression first acts on a row, and then dictionary based compression for multiple instances of the same values. When databases have duplicate data in many of the fields, space is conserved with this method of compression. In most exam questions on data compression clustered indexes are in use because in a large table they would exist anyway. When page compression is used on a table heap, any existing pages are not compressed, only new ones. A rebuild of the heap is required to ensure that the compression is used for all pages. On the exam when instances of clustered indexes are in use for tables with compression ensure that any maintenance conducted to support compression drops the clustered index first.

Exam Prep: Data Compression
You run a database called Headers. The Headers table includes products that are dependent on other products for use. You need to store the number of dependent products for each product. For example: 1 to 10 Products.

The 302Info .NET web application requires the ability to do an efficient sort based on the number of parts. You need to create a NumberofParts column. What is the solution?

Exam Solution: Data Compression
This solution is focused on using varchar capability within SQL Server 2008. Due to the requirement of accessing filtered data, the solution must include use of a filtered index on the created column to provide for the number of dependent parts. Filtered indexes on sparse columns support efficient sorting on the data. You should create the NumberofParts column as a sparse column of type varchar, which will enforce data compression and reduce storage.

For questions like this on example on the exam look for answers that use Null in their solution as this would immediately discount them. Also watch out for answers using non-clustered indexes in the solution. In this example a non-clustered combined with nullable columns would result in no storage savings. Also note you can't add a sparse column to a clustered index.

Maintaining Indexes

SQL Server administrators must understand all types of indexing for database tables to ensure accuracy when answering exam questions. The types of SQL Server indexes include:

- Clustered Indexes
- XML Indexes
- Non-clustered Indexes
- Composite Indexes
- Unique Indexes

Certain index types are linked to the column type in use. A spatial index can only be used with a spatial data type. A spatial index as with other indexes requires a primary key prior to creation. A spatial index supports unique SQL Server data type features in the area of geometry and geography, which use the spatial data types. Indexes are an underlying performance enhancement, which support rapid queries and speed of data access. SQL Server database tables can contain clustered or non-clustered indexes. A clustered index will keep the actual data rows at the leaf level of the index. Normally, they are in ascending or descending order. Only one clustered index can exist on any one table or view. The term heap refers to a table that has no clustered index – think slow or small! Non-clustered indexes differ from clustered indexes in that they don't actually contain the data in the rows, but only pointers to the data. Non-clustered indexes can't be sorted, but you can create multiple non-clustered indexes per table or view. In SQL Server 2008, 999 non-clustered indexes are supported. The composite index and the unique index are two more forms of SQL Server table indexing. A composite index will contain multiple columns, up to 16. A unique index, as you may have guessed, is based on a unique table parameter such as a primary key. For example, when you create a primary key in a table an automatic unique clustered index is created. XML columns in tables require an XML index which will create an indexed view of all the XML tags, values, and paths of the XML data.

As an SQL administrator you will need to understand the commands available in SQL Server 2008 to change, delete, or add indexes. These tasks are always included in other SQL Server maintenance operations. They are also included in performance tuning as dropping and rebuilding indexes is a fairly common SQL Server administrator task. The exam will expect fluency with the following T-SQL commands:

- CREATE INDEX
- CREATE SPATIAL INDEX
- CREATE XML INDEX
- DROP INDEX

Exam Prep: Alter Index
You support a database called 302EnginesDB. It contains a table called PonyCars. The following table describes the index setup of the table. IX_PRIMARY is a clustered index. The other listed indexes are all non-clustered.

Column Name	Type of Data	Data In Indexes
CarID	Char(10)	IX_PRIMARY IX_CARTYPE IX_CARPRICE
Type	Varchar(25)	IX_CARTYPE
Description	XML	None
Value	Decimal	IX_CARPRICE

Performance testing indicates you need to add the Description column to the IX_CARTYPE index as an included column. Table downtime should be minimized. What is the solution?

Exam Solution: Alter Index
You would use the CREATE INDEX WITH DROP EXISTING and perform this command offline.
The requirement to rebuild this index offline is due to adding an XML column, which is a large data type. You can rebuild the index online if a large data type is not included. Anytime a column is added to an index you must drop the existing index. In this case ALTER INDEX will not work because you can't add a column with this command. In exam questions that focus on solving slow query time due to index fragmentation, ALTER INDEX is a solution with the REBUILD WITH ONLINE switch.

Test Tip: Exam questions that focus on index management will always use DBCC commands in possible answers. Take note that these commands, while good for database maintenance, do not address most index management issues.

Domain 6 – SQL Server 2008 Monitoring and Troubleshooting

Troubleshooting SQL Server problems is primarily an SQL administrator's job and will factor into the exam questions. The ability to rapidly solve SQL Server issues is related to exam objectives in this domain and being able to rapidly troubleshoot problems. The areas of SQL Server service statuses, problems created by concurrency such as a dreaded deadlock, SQL Server Job Agent issues, and how to quickly find error information all contribute to problems that occur with SQL Server 2008 and how to quickly assess and remediate those problems. In production situations SQL Server 2008 downtime is not an option, and this domain will demonstrate methods to quickly solve problems that affect uptime.

SQL Server Service Problems

SQL Server's uptime is related to the functioning of the product features that are installed.
These features are listed as services and reflect the range of SQL Server capabilities. The most important service is the SQL Server database engine, which is termed the SQL Server (name of instance) service.
A quick look in the SQL Server Configuration Manager under the SQL Server Services shows the status of all SQL Server Services. Services include the SQL Server Browser, SQL Server Agent (name of instance), SQL Full-text Filter Daemon Launcher (name of instance), and any other installed SQL services, for example, SQL Reporting Services (SRS). SQL Server status depends on logons, the start mode defined, and that no system issues or SQL Server configuration has stopped SQL Server from starting.

At times the problem is that the SQL Server service will not start. You need to immediately troubleshoot this, and the quickest way is by changing the sqlserver.exe file to use the –f switch to start up. The –f switch starts SQL Server in a minimal configuration mode and allows a single connection to the SQL Server.
This type of startup mode can address multiple issues, for example, when logging has filled up the drive it is assigned to and the SQL Server service can't start up. This option is easily accessed through the properties section\advanced tab of the SQL Server service in the SQL Server Configuration Manager.

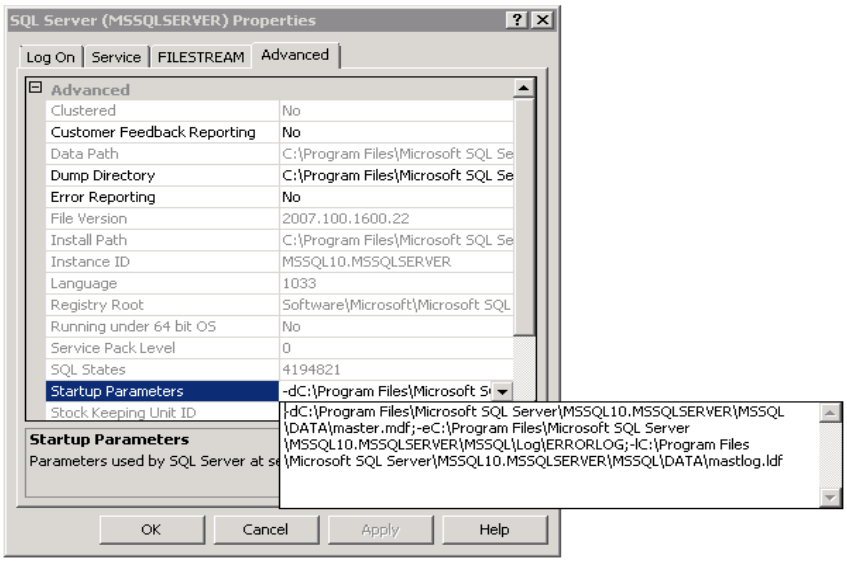


Figure 39: SQL Server Advanced Startup Parameters

If the SQL Server Agent is set to start automatically, this must be disabled prior to launching the SQL Server service with the –f option. This is due to the –f mode option only allowing one connection to the SQL server. If the SQL Server Agent starts automatically this service will use the one allowed connection to the SQL Server service.

SQL Server provides the option of a Dedicated Administrator Connection (DAC) to make a connection to SQL Server 2008 when normal connection attempts fail. DAC access only works on the server itself. However, you have the option to use the sp_configure stored procedure to configure the DAC to allow remote connections. But this must be enabled prior to a problem that requires this special connection. The DAC is usable via sqlcmd and requires the special administrator switch –A. The best strategy is to use the DAC to connect to the master database, which is usually available when problems occur and the SQL instance is running.

Troubleshooting SQL Server services also requires an understanding of service startup types. The SQL Server Browser service allows the use of the SQL Server Resolution Protocol (SSRP) to resolve which port SQL Service is listening on. This obviates the requirement to use standard SQL Server ports such as 1434, and allows other port use to avoid scans for SQL Server services by malicious programs. Additionally, this helps support multi-instance SQL Server use for the same physical service. The SQL Server Browser service is not set to start by default. If you plan on using this, then set the properties of this service in the SQL Server Configuration Manager to the automatic startup type. The exam will pose various questions that focus on service use such as the SQL Server Browser service, which doesn't start automatically. In this case there is no need to over think the question, just focus on ensuring required services are started automatically.

Test Tip: Services like the SQL Server Browser service do not require extensive permissions to run. When evaluating exam questions make sure you find solutions that are paired using the SQL Server Browser service with the Local Service account versus the Local System or the Network System account. The Local Service account has a reduced permission set compared to the Local System account with extensive permissions, which the SQL Server Browser service does not require.

C2 audit mode and the common criteria along with other logging choices can impact disk space and serve as the cause for SQL Server failure. When logging this much data, disk space is impacted. If smaller drives are used as the location for audit log files and the physical disk fills up, SQL Server will not start. The solution on the exam in this case can range from starting SQL Server with the –f switch to moving the audit files to another drive with more space.

The account under which an SQL Server service runs also presents an area for troubleshooting. If you use all Active Directory domain accounts as the service accounts for SQL Server services and any Active Directory group policy options require the Active Directory accounts to change the password (or they expire), this will result in a failure of the SQL Server service using that Active Directory account. Normally, you can use Powershell scripts or other management scripts to change SQL Server Active Directory account passwords within the required time, which ensures that the account is changing passwords per company policy or best practice. However, when an Active Directory account expires, locks out, or requires a password change the SQL Server service in question can switch to Local System to get SQL Server started again immediately while troubleshooting. Again, for exam purposes, remember that the SQL Server Configuration Manager supports rapid change of SQL Server service account information.

Exam Prep: Troubleshooting SQL Server Services
You have two Microsoft SQL Servers on your network. One is an SQL Server 2005 instance, and the other is an SQL Server 2008 instance. SQL Server is configured with default connection information. The SQL Server Agent runs on both SQL instances.

A new corporate application will access both SQL Servers for its data. You are working with the application developers to troubleshoot any connectivity problems that may arise.

The developers report that they can only find connections to the SQL Server 2005 instance, but are forced to specify a port when connecting to the SQL Server 2008 instance.

You are required to ensure both SQL Servers can return connection information automatically without specifying a port. The solution should not change the established network security. What is the solution?

Exam Solution: Troubleshooting SQL Server Services
In this type of exam question the challenge of connecting to SQL Server 2008 is presented. The service in question should immediately come to mind--the SQL Server Browser service. It is not configured to run automatically in SQL Server 2008, which explains why the developers cannot enumerate or find it automatically. The simple solution is the correct one--you need to start the SQL Server Browser service and configure the service to start automatically. Answers that focus on changing an Active Directory service account are not correct. Other answers on the exam for this scenario tend to focus on the SQL Server Agent service, which again is not related to this issue.

Concurrency Problems

SQL Server operations sometimes suffer latency or dropped transactions due to blocks, deadlocks, or locks. These are problems that require rapid identification with the SQL Server 2008 tools available.

As SQL Server reads and writes data to a database locking occurs to support concurrent operations. This is SQL Server's method to ensure that the same data is not written to the same row/page at the same time. This is also the heart of shared database access by users and applications.

The exam focuses on the tools available to troubleshoot concurrency problems and when to use them. The SQL Server tools for diagnosing concurrency problems are the SQL Server Profiler, SQL Dynamic Management Views (DMV), and the activity monitor.

Blocking occurs when a lock is created and then escalated based on the size of the scan. For example, a scan of data in a table that exceeds 5,000 locks will cause a lock escalation and ultimately block the next process requesting the reading or writing of data. Blocks by default do not timeout. You can monitor SQL Server for block conditions using the SQL Profiler and filter on the event (Lock: Escalation event), the Dynamic Management View (DMV), or regular polling of system information for lock conditions. These tools will help gather the information on blocks such as the block and lock time per object, which will lead to a plan to remediate the condition. The SQL Profiler is available within the SQL Server Management Studio and allows you to record information over time for a deeper analysis of the blocking issue. The results are recorded to a trace file and are available for examination.

Test Tip: On the exam, questions focusing on time as a component of troubleshooting the lock issue suggest the use of SQL Profiler.

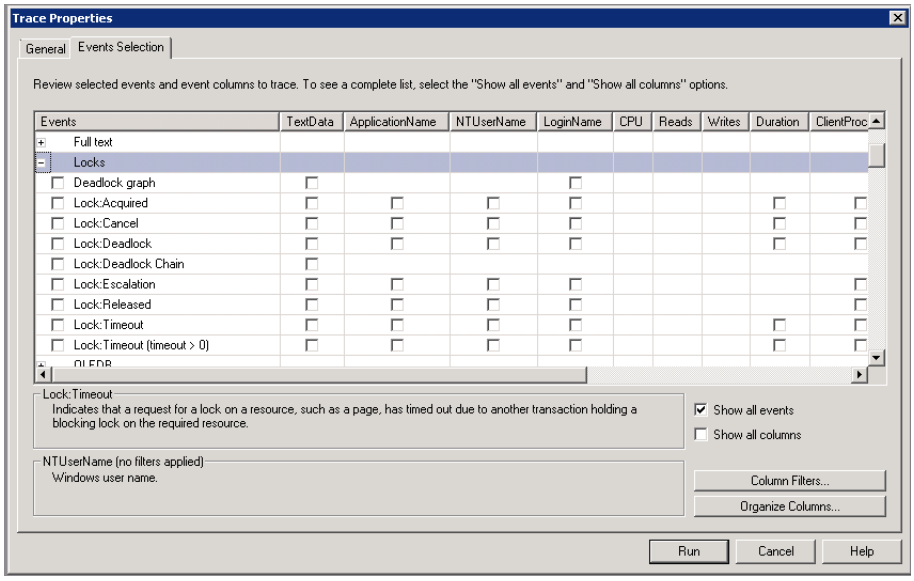


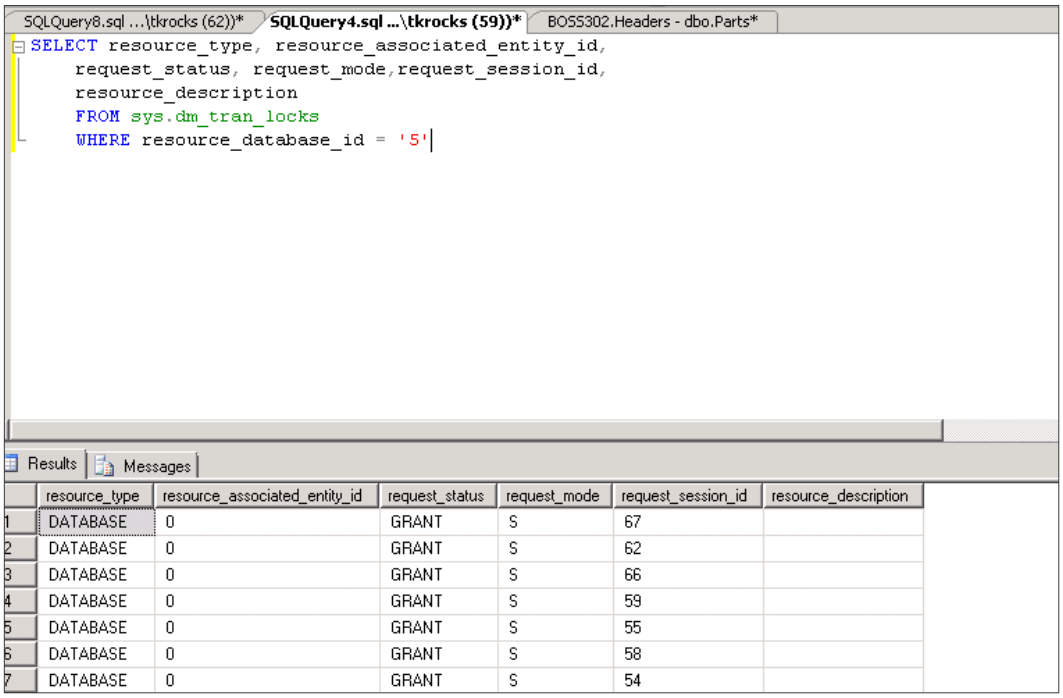
Figure 40: SQL Server Profile Lock Events for Tracking Block Conditions

Deadlocks occur when several tasks within the same SQL Server memory space block each other by having a lock on a resource that another database function is trying to lock. This situation is caused by a variety of resources to include locks, worker threads, memory, parallel query execution, and multiple active result sets. SQL Server is configured to automatically search for deadlocks every 5 seconds with the database engine deadlock detection scheme. The resolution to a deadlock is made by the database engine and is a result of choosing a thread to continue. This results in an error 1205, which is prevalent in the exam questions on how to track deadlock conditions. Additionally, the trace flags 1204 and 1222 exist to document deadlock conditions that have occurred. You can turn on trace flag 1204 as a standard event to write to the SQL Server error log, which will allow parsing of this error by your management activities and response to the occurrence. The flags list all relevant information for the deadlock to include what process was running, the SPID, the mode, the row in question, the object, the key, and other metadata that is available. When using SQL Profiler to track deadlock activity ensure you select the Deadlock Graph as an option.

Exam Prep: Troubleshooting SQL Server Deadlocks
You manage a SQL Server 2008 instance. Customers report that purchases often timeout with a strange error. You must quickly determine which resource is causing this problem. How can you determine the problem quickly?

Exam Solution: Troubleshooting SQL Server Deadlocks
In this type of exam question you must monitor the SQL Server processes and determine what process is causing the issue. This is conducted by the SQL Server Profiler and uses the deadlock graph event. This will provide detailed historical data on the problem and allow you to view the process causing the deadlock. Many exam answers in this area will suggest the use of a dynamic management view with tran lock. However, this only shows current locks and does not reveal any history. For example, you may need to run the SQL Profiler and create a trace file that spans the entire time period in which the problems appear. The Activity Monitor also seems like a good choice for this type of problem, but it does not provide historical data either, only a just in time view of what is locked currently. Sometimes the answers will also have an option to use sp_lock. But note that sp_lock is a retired legacy command that will not provide any help with troubleshooting a deadlock issue.

Dynamic management views, in general, provide views and functions to return server state information to support health monitoring, diagnosing of problems, and performance tuning. The sys.dm_tran_locks dynamic management view is very important when troubleshooting calls for quick views of transaction locks at various times of the day. Think of this dynamic view as a snapshot for quick views and the ability to rapidly gleam information about a deadlock condition. Figure 41 below demonstrates the use of the above dynamic management view to display lock information for a database with an ID of 5.



	resource_type	resource_associated_entity_id	request_status	request_mode	request_session_id	resource_description
1	DATABASE	0	GRANT	S	67	
2	DATABASE	0	GRANT	S	62	
3	DATABASE	0	GRANT	S	66	
4	DATABASE	0	GRANT	S	59	
5	DATABASE	0	GRANT	S	55	
6	DATABASE	0	GRANT	S	58	
7	DATABASE	0	GRANT	S	54	

Figure 41: Sys.dm_trans_locks Dynamic Management View

SQL Server Agent Problems

The SQL Server Agent is responsible for running many of the configured management jobs for SQL Server and associated jobs to support database applications, data movement, and a host of important database tasks. When the SQL Server Agent fails, rapid troubleshooting must ensue to correct the problem, especially when the SQL Server Agent defined job is critical to production operations. As covered in other domains, the notification features of the SQL Server Agent are powerful and ensure that you know when a failure occurs.

One of the main areas for SQL Server Agent job failures is the security context in which the job is run, or security contexts for steps in a job. SQL Server Agent can use a proxy account to run jobs, which are the credentials of a Windows user. The SQL Server Agent can use the security context of the established proxy user to impersonate it and run the job. SQL Server Agent job failures can fail quickly based on the role of the user configured for the job and SQL Server role of the user, or the roles of the user within msdb. SQL Server Agent proxy accounts are created in the SQL Server Management Studio\SQL Server Agent object, under the Proxies section. Select the subsystem for the proxy account upon creation.

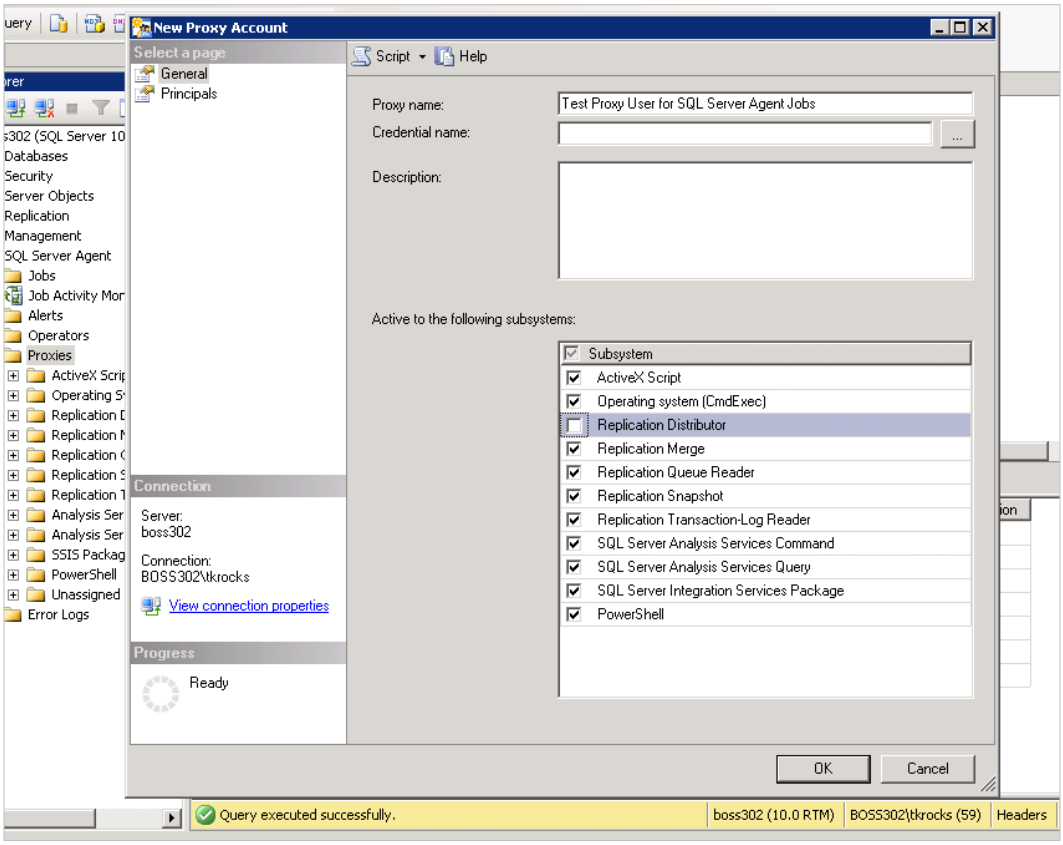


Figure 42: SQL Server Agent Job Proxy Account Creation

The EXECUTE AS command is also a security context to support SQL Server Agent job execution. In some cases the proxy account or other security context may not have the correct permissions to execute a T-SQL job step. The EXECUTE AS command allows the use of a security context to complete the T-SQL step. The SQL Server Agent proxy can't be used with a T-SQL job step because it is a Windows user account. The stored procedure sp_enum_proxy_for_subsystem provides a return of which proxies have permission for which SQL Server Agent job subsystems. This is another useful tool in troubleshooting permission problems.

Another area of focus for the successful execution of SQL Server Agent jobs is the ability to connect to required SQL instances. In multi-instance SQL Server configurations the SQL Server Agent can serve in a master role, and then target other SQL Server instances for job execution. However, to ensure that jobs work without SQL Server port or instance connectivity issues make sure you create a SQL Server alias for the target servers. Additionally, during the creation of the Master/Target SQL Server job setup you can specify the servers for connections.

Helpful diagnostic information for the status of steps in SQL Server Agent jobs can be obtained by configuring each job step to log detailed information to a table. The contents of this table can be accessed quickly with the following stored procedure: sp_help_jobsteplog. The SQL Server Agent core options support the settings for the maximum number of rows entered for each job and the maximum size of the log about the job.

Finding Error Information

SQL Server 2008 provides several areas for rapid access to error information. These areas include: the error log, the agent log, the job execution history, and event logs. SQL Server administrators will study these logs daily to quickly diagnose problems, and the exam will test your knowledge of where to find relevant information for reported problems.

The SQL Server error log is the first place to check for problems with SQL Server. This log file is located in the default location of the first SQL Instance, which is \\installdrive\program file\Microsoft sql server\ mssql10.mssqlserver\mssql\log\errorlog. The most recent file doesn't have an extension. This log will contain SQL Server specific errors, not the type of errors you may find in one of the Windows Server event logs. The size and location of the error file is adjustable via the SQL Server Configuration Manager. The SQL Server Management Studio has a log file viewer utility under the Management node, which supports the aggregated view of several critical SQL Server logs. They are the Database Mail log, the SQL Server logs, the SQL Server Agent logs, and the Windows NT logs of the server hosting SQL Server. The log file view contains a filter setting to rapidly search the logs for keywords, such as the user, computer, and text contained in the message and a Source. Additionally, any of these logs are exportable for further review or upload into other systems.

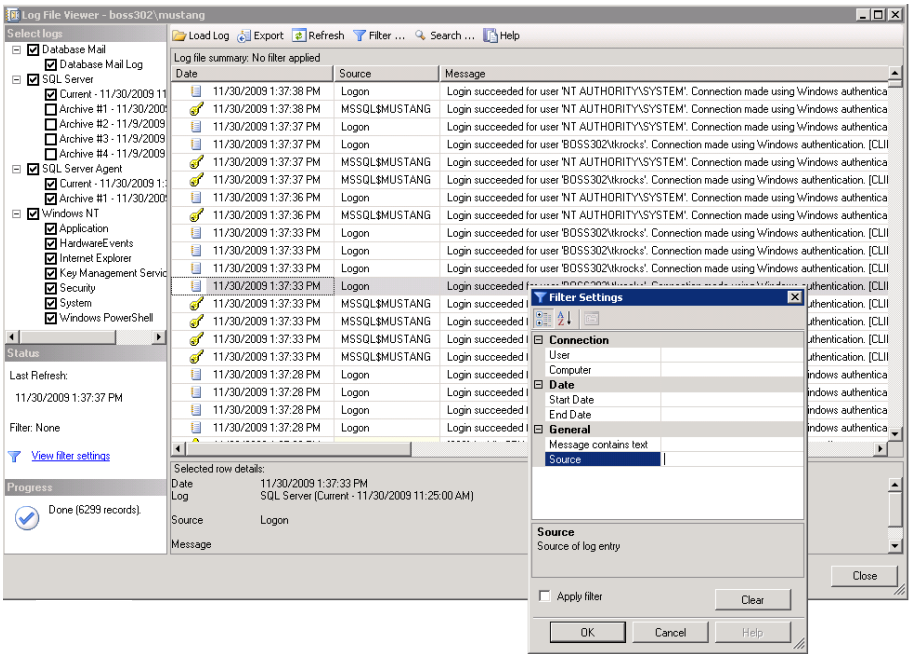


Figure 43: SQL Server 2008 Combined Log View with Filter Settings

Custom error information can be written to the SQL Server logs or Windows event logs. The management stored procedure xp_logevent writes user defined messages to a log destination. Stored procedures in use in SQL Server can harness this capability to ensure database errors are meaningful and assist in the troubleshooting process. Anytime a stored procedure raises an error message, they are automatically stored in the sys.messages table. A query against this table provides any errors logged by stored procedures regardless of their severity level.

Exam Prep: Finding SQL Error Information for Job Execution
You manage 10 instances of Microsoft SQL Server 2008. You have one instance that has a problem every hour with the SQL Server Agent service. Troubleshooting this problem requires detailed information about the job execution. Execution trace messages should write to the SQL Server Agent error log. How will you achieve this with in the quickest manner?

Exam Solution: Finding SQL Error Information for Job Execution
This exam question requires SQL Server Management Studio. A quick right-click on the SQL Server Agent node will make the change to this tool easily. This type of question will present solutions that focus on triggers that do not address this logging question. Other solutions suggest using DBCC as a form of turning trace flags on. Again, this does not address configuring the SQL Server Agent to write trace execution messages.

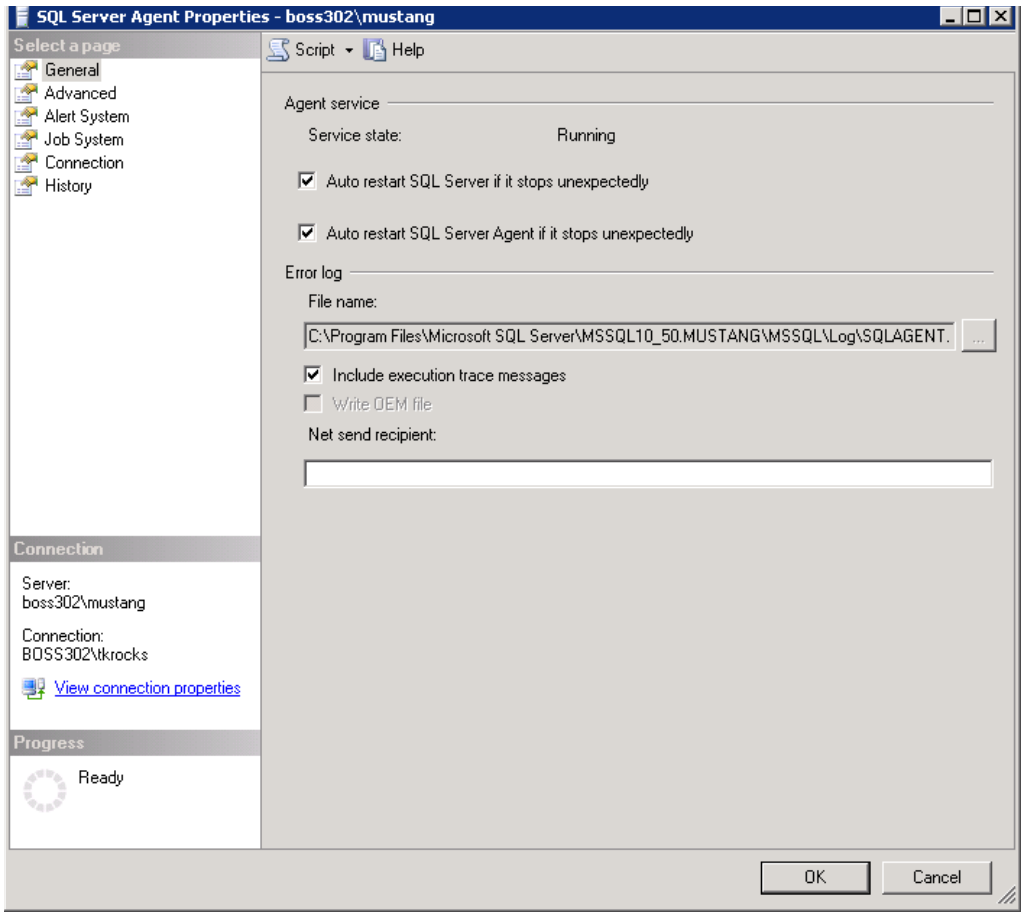


Figure 44: Setting the Execution Trace Messages Option for SQL Server Agent

Domain 7 – Optimizing SQL Server Performance

SQL Server 2008 performance is always important in real world use and for this exam. SQL Server responses that are latent or fail impact the applications and solutions that use the database tier. Imagine ecommerce solutions that fail to return product inventories or process orders as consumers are clicking and waiting. SQL administrators are expected to rapidly identify problems and correct them through data analysis and trending of SQL Server operations. The tools covered in this domain provide, in depth, the data that classifies performance issues and fixes them quickly.

The Resource Governor

The SQL Server Resource Governor is a member of the collection of SQL Server 2008 policy initiatives. The resource governor allows the definition of groups or resource pools that use SQL Server resources, and the placement of classifiers or policy statements on the group. A resource pool is comprised of the physical resources of the server--CPU and memory.

The processing of constraining a resource works as follows:

1. An incoming session to SQL Server is classified, e.g., as a report server query.
2. The session is identified through its user-defined classifier function via Policy Management.
3. The session is routed to a workload group, e.g., Report_Server.
4. The workgroup will use the resource pool defined to it, e.g., Report_Server_Pool.
5. The resource pool enforces the limits required by the defined application.

An illustrative example involves preventing users in a group called MassiveQueryFun from using over 40% of the SQL Server's CPU time when running queries. The group in question is restricted much like a restrictor plate on an engine in NASCAR racing. The restriction or policy element is the classifier that activates the thresholds.

The resource governor is accessed via the Management Node of the SQL Server Management Studio. Setup can occur via T-SQL statements. The elements of defining a resource with limits are the resource pool, the workload group settings, and then the creation of classification functions for the resource pool. Once the resource classification is registered with the resource governor then enable it and monitor the performance. Here the DMVs allow quick query information about the resource use. The example below establishes three workload groups under the default system resource pool and assigns the SQL Server login 'power' to the group, while establishing a classifier and then enabling it.

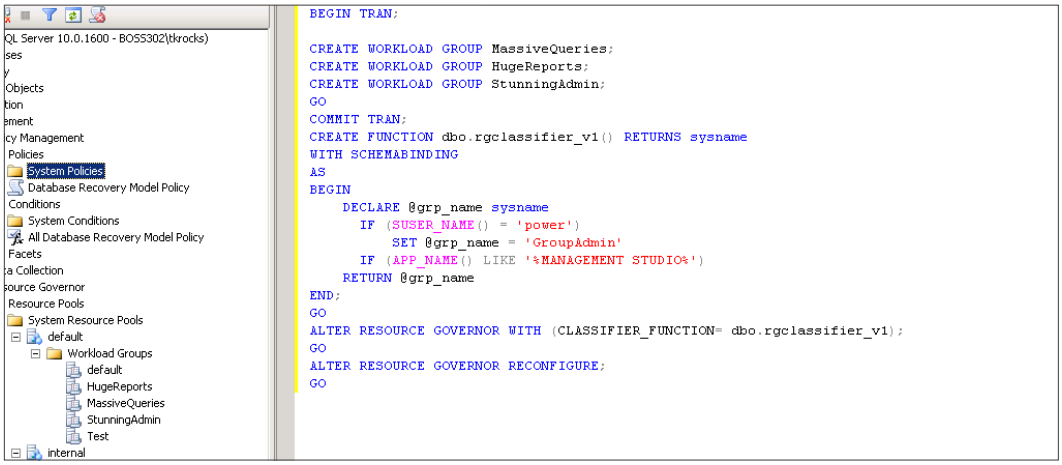


Figure 45: Resource Governor Setup T-SQL Statements

Exam Prep: Using the Resource Governor
You are responsible for managing an SQL Server 2008 server. A group exists called CrazyEddiesQueries. This group conducts resource-intensive queries. You are a smart SQL administrator so you create a workload group in Resource Governor and monitor the performance via a DMV query when members of CrazyEddiesQueries execute a query.

The CPU use during this query period hovers around 92%. You determine that CrazyEddiesQueries must be limited. What should you do?

Exam Solution: Using the Resource Governor
In this case, most of the elements of limiting the CPU are in place. You have a workload group in place; however, a classifier is not established to limit members of the CrazyEddiesQueries group from using too much CPU time. You will create the classifier function to ensure the CPU limits are enforced.

You can quickly eliminate the exam answers that do not associate any workload groups with internal resource pools. SQL Server as its own instance uses the internal resource pools, and this won't work for user defined classifier functions. Rebuilding indexes always seems like a good solution to a performance issue, but this could use MORE CPU time than it would reduce, so avoid any index building activities. Max degree of parallelism settings also figure into possible solutions for this type of problem. The max degree of parallelism settings applies to the number of server processors to use in executing a query versus addressing "individual" queries. This is again a system wide setting that won't help address specific queries that require boundaries.

In summary, always think of the resource governor in terms of its elements. Create a resource pool with a given name. Assign the values for minimum and maximum CPU or memory use. Create a workload group to use the resource pool. Then the established classifier function will return the name of the workload group in which SQL Server will execute the query.

Database Engine Tuning Advisor

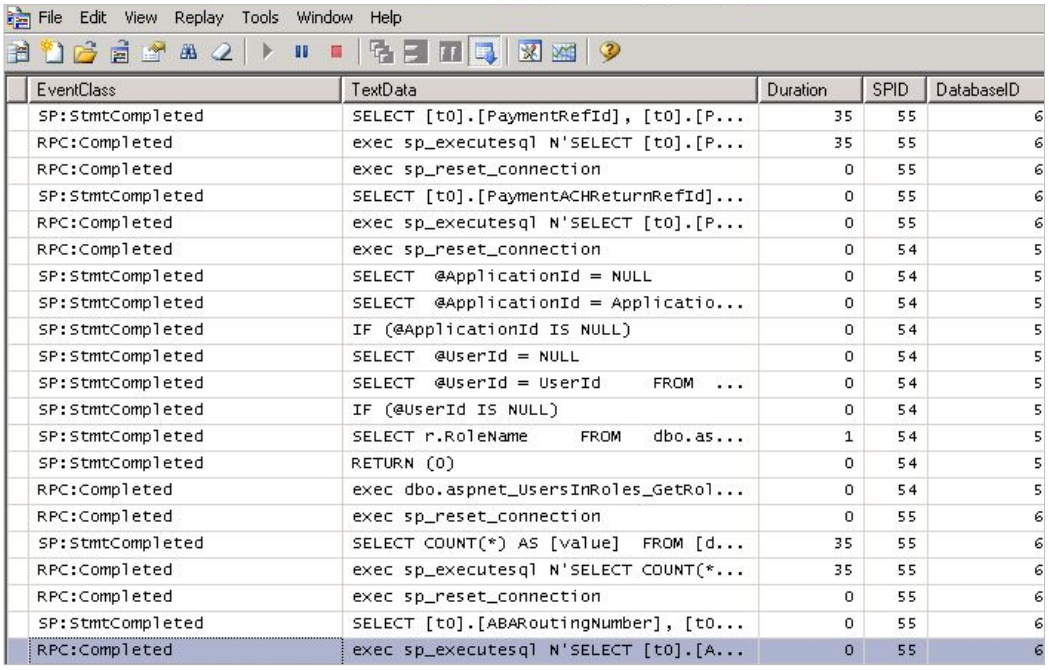
The database engine tuning tool is another weapon in the battle to optimize SQL Server. This is a graphical-based tool that runs from SQL Server Management Studio and provides recommendations based on settings selected for optimizing a database. The main focus of the database engine tuning tool is recommending the best configuration with indexes, index views, and partitions. The tool conducts processes that would be quite complex to analyze manually.

The tool analyzes collected database workloads from trace files or tables collected via the SQL Server Profiler. The recommendations of the database engine tuning tool include:

- Best configuration of indexes for databases based on query optimization
- Suggests aligned or non-aligned partitions
- Suggests index views
- Conducts analysis of the possible changes, in effect a query test for index use, spanning the query across tables, and actual query performance
- Allows input of design choices to affect the actual database design
- Small query support
- Provides a complete array of reports documenting recommendations
- Allows some simulation of database server conditions, e.g., CPU constraints

The basic steps for database engine tuning are:

1. Create a workload for the database tuning activity either in T-SQL script by collecting a trace file or collecting trace data to a table. This should represent the work conducted by the database you want to tune.
2. Determine what you want to tune: indexes, index views, or partitioning best practices.
3. The database engine tuning output is in the form of logs, summaries, and recommendations rolled up into reports.
4. If necessary, you can use the results to perform hypothetical modeling by changing parameters and analyzing them.
5. Once you agree with tuning recommendations you can apply them to the database in question, or schedule for a later time.



EventClass	TextData	Duration	SPID	DatabaseID
SP:StmtCompleted	SELECT [t0].[PaymentRefId], [t0].[P...	35	55	6
RPC:Completed	exec sp_executesql N'SELECT [t0].[P...	35	55	6
RPC:Completed	exec sp_reset_connection	0	55	6
SP:StmtCompleted	SELECT [t0].[PaymentACHReturnRefId]...	0	55	6
RPC:Completed	exec sp_executesql N'SELECT [t0].[P...	0	55	6
RPC:Completed	exec sp_reset_connection	0	54	5
SP:StmtCompleted	SELECT @ApplicationId = NULL	0	54	5
SP:StmtCompleted	SELECT @ApplicationId = Applicatio...	0	54	5
SP:StmtCompleted	IF (@ApplicationId IS NULL)	0	54	5
SP:StmtCompleted	SELECT @UserId = NULL	0	54	5
SP:StmtCompleted	SELECT @UserId = UserId FROM ...	0	54	5
SP:StmtCompleted	IF (@UserId IS NULL)	0	54	5
SP:StmtCompleted	SELECT r.RoleName FROM dbo.as...	1	54	5
SP:StmtCompleted	RETURN (0)	0	54	5
RPC:Completed	exec dbo.aspnet_UsersInRoles_GetRol...	0	54	5
RPC:Completed	exec sp_reset_connection	0	55	6
SP:StmtCompleted	SELECT COUNT(*) AS [value] FROM [d...	35	55	6
RPC:Completed	exec sp_executesql N'SELECT COUNT(*...	35	55	6
RPC:Completed	exec sp_reset_connection	0	55	6
SP:StmtCompleted	SELECT [t0].[ABARoutingNumber], [t0...	0	55	6
RPC:Completed	exec sp_executesql N'SELECT [t0].[A...	0	55	6

Figure 46: SQL Server Profiler Trace File Using Database Tuning Template

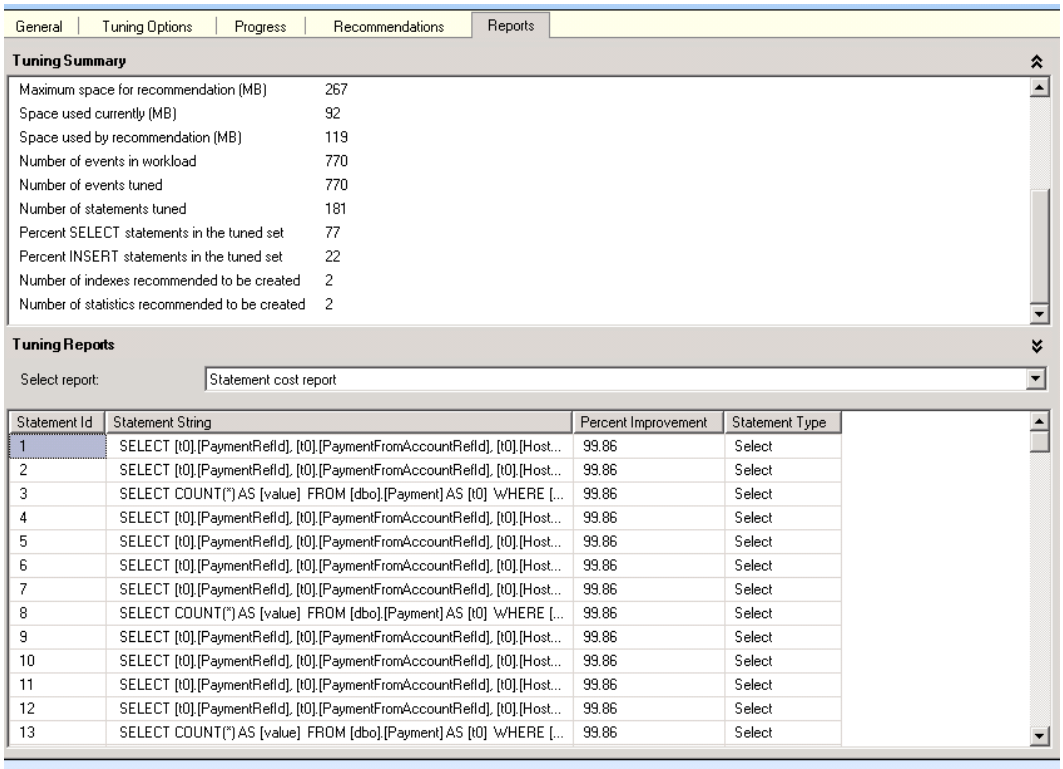


Figure 47: SQL Server Database Tuning Advisor Report Recommendations

Security is a consideration when using database tuning. Database tuning is activated by a member of the sysadmin role. Once database tuning has been initialized by a member of the sysadmin role, then you can extend the right to run database tuning to logins that have the db_owner role for a database. Any user who is a member of the db_owner role can run the database tuning advisor for that database.

Test Tip: The database tuning advisor is also executable via the command line with dta.exe versus using the graphical flavor of it. The command version of dta.exe has an extensive list of syntax switches, which are easily accessed by typing dta.exe /?.

Exam Prep: Using Database Tuning Advisor
You manage an SQL Server 2008 instance called Racing\Performance. You create a workload with the SQL Server Profiler and analyze it with the database tuning advisor. Two statements run by users Mufflers and Sparks are not tuned. You need to tune these statements, while maintaining security. How will you achieve this?

Exam Solution: Using Database Tuning Advisor
In this question we are asked to suppress login names within the database tuning advisor. The reason is that when user names exist in the database tuning advisor it has impersonated the user when running the statements in question. For database tuning to work it must be able to access the execution plan, which is possible via the SHOWPLAN permission. The column named LoginName should be removed, which will allow database tuning to make recommendations as it impersonates a user with the proper credentials. In any type of security context question on the exam with database tuning advisor avoid answers that grant too much security and those that grant security with no relation to the issue at hand. An example is an answer that indicates you should assign users to the processadmin role to facilitate database tuning recommendations.

Analyzing Trace Data with SQL Server Profiler

The SQL Server Profiler provides a great tool to capture the data needed to optimize SQL Server database operations and ensure you can respond in a timely fashion to performance challenges. The exam objectives expect the ability to configure SQL Server Profiler to gather data for specific problem types and ensure you can filter the data and collect what is needed in relation to the problem under investigation. The SQL Server profiler is accessed via the Performance Tools program group, or by clicking on Tools in the SQL Server Management Studio.

The security required to run the SQL Server Profiler is the same as the T-SQL statements used to create traces. Users must have at least the ALTER TRACE permission. One security note is that if a user has VIEW SERVER STATE, ALTER TRACE, and SHOWPLAN permissions they can view the login and other security information in the traces. This security aspect is important to note while evaluating exam questions. It is very applicable when analyzing queries with SHOWPLAN results on.

SQL Server Profiler comes with several default templates for monitoring and includes ones for monitoring stored procedures, T-SQL execution, and tuning. Any SQL Profiler monitoring session can be saved as a template for future use. Saved SQL Profiler templates can be exported and imported by other SQL Server Profiler instances. The SQL Server Profiler includes a deadlock graph option to help identify deadlocks between processes. SQL Server Profiler traces can be saved to the Windows Server file system or to an SQL Server table. SQL Server Profiler traces saved in tables can be queries for additional event data. Figure 46 illustrates the SQL Server Profiler in action, with event class recording.

SQL Server Profiler is commonly used to:

- Identify poorly written or performing queries
- Find deadlock causes
- Identify the performance characteristics of stored procedure execution
- Monitor SQL Server audit activity, such as logins or logoffs
- Show granular user based activity with T-SQL use
- Collect data for performance testing or other replay activities
- Collect data for use in database tuning activities

Exam Prep: Using SQL Server Profiler
You manage an SQL Server 2008 instance. An application team has asked for an XML formatted report that shows processes and resources that are participating in deadlocks. How can you provide this quickly?

Exam Solution: Using SQL Server Profiler
The solution is to run an SQL Server Profiler session and select the Deadlock graph event under the Locks events section. This supports collecting information about resources and processes that have become deadlocked. The SQL Server Profiler supports extraction of the Deadlock events to a separate XML file.

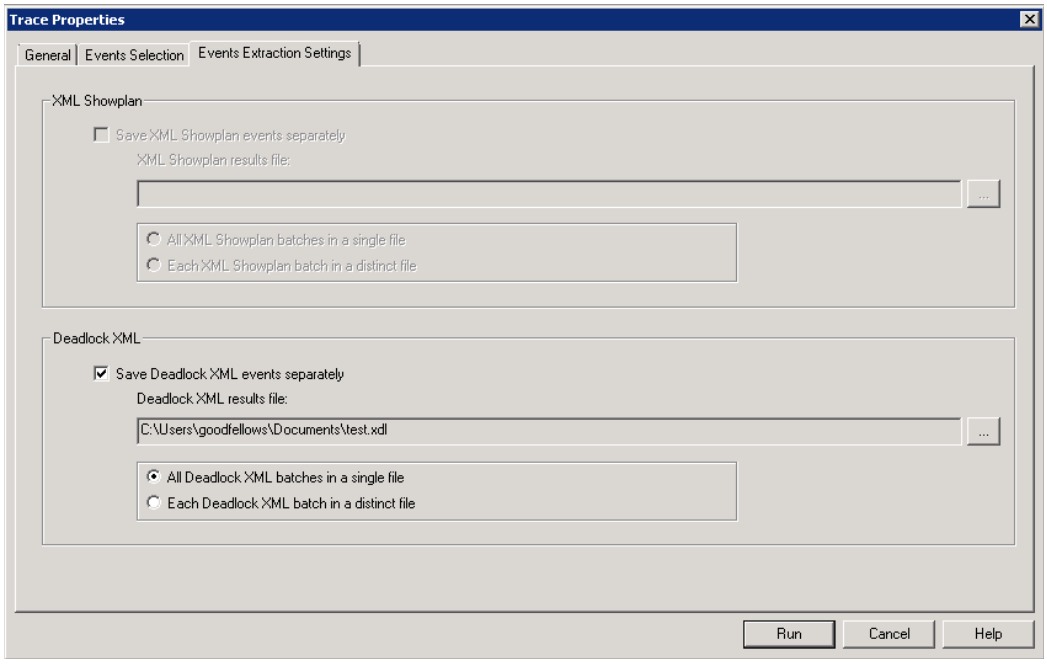


Figure 48: SQL Server Profiler XML Extract Screen for Deadlock Events

Analyzing Performance Data with Dynamic Management Views (DMV)

SQL Server DMVs provide a rapid way to find out the current state of an SQL operation. DMVs such as sys.dm_exec_query_stats provide instant point in time data on the state of queries, while other DMVs show wait states and many other elements. The use of the DMV is predicated on a snapshot type data need versus sustained trending provided by tools like Performance Monitor and SQL Server Profiler.
Queries with DMVs will return server state information and can provide the data required for diagnosing issues or tuning performance. DMVs are grouped into sections of use and span the entire range of SQL Server functions.

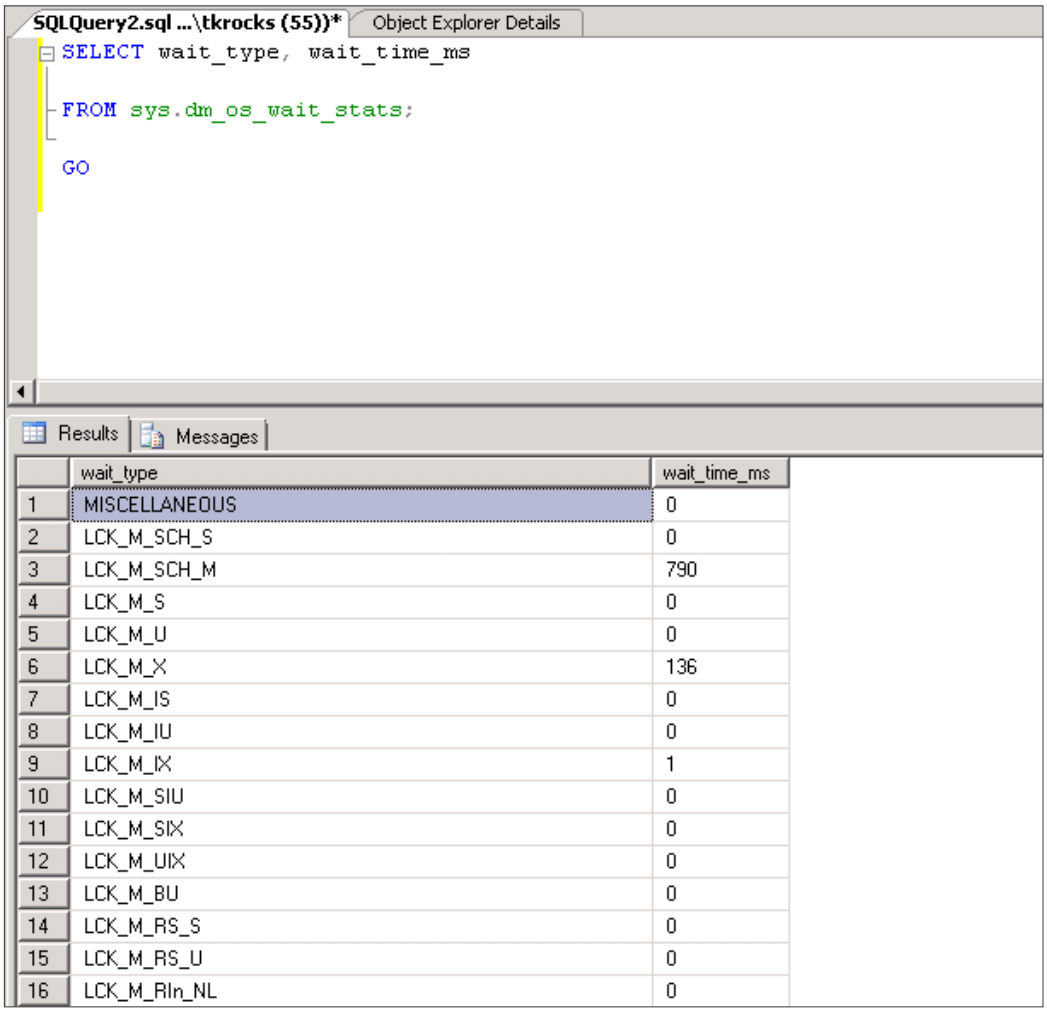
DMV Categories

- Security
- Transactions
- Operating System
- Extended Events
- Service Broker
- Resource Governor
- Replication
- Query Notifications
- Objects
- I/O
- Indexes

- Full-Text Search
- Execution
- Database
- Mirroring
- CLR
- Change Data/Audit

When using DMVs make sure you understand that there are thousands of possible T-SQL combinations of DMV command syntax (this will be covered on the exam). However, always prefix the name of the view or function by using the sys. schema. Then add the syntax for the function you query. Security considerations require a minimum of SELECT permission on the object and VIEW SERVER STATE or VIEW DATABASE STATE permissions.

A sample DMV query with results is shown below.



	wait_type	wait_time_ms
1	MISCELLANEOUS	0
2	LCK_M_SCH_S	0
3	LCK_M_SCH_M	790
4	LCK_M_S	0
5	LCK_M_U	0
6	LCK_M_X	136
7	LCK_M_IS	0
8	LCK_M_IU	0
9	LCK_M_IX	1
10	LCK_M_SIU	0
11	LCK_M_SIX	0
12	LCK_M_UIX	0
13	LCK_M_BU	0
14	LCK_M_RS_S	0
15	LCK_M_RS_U	0
16	LCK_M_RIn_NL	0

Figure 49: DMV of System Wait States

Exam Prep: Using DMV for Optimizing SQL Server
You manage a server running SQL Server 2008 Enterprise. The accounting department has reported a performance issue with their queries. They report they are running for 30 minutes versus the normal time of 2 minutes. You have to track down the syntax of this long running query. How will you solve this?

Exam Solution: Using DMV for Optimizing SQL Server
The DMV sys.dm_exec_query_stats provides information about queries and lists their running times. This will provide the name of the query that is running too long. Once you have the name identified with the above DMV, then you can specify a SELECT statement to identify the syntax of the query. Use the syntax SELECT * FROM sys.dm_exec_sql_text(<sql_handle>). This will identify the syntax of the query. Once you have the syntax you can analyze why the query is running so long. This type of DMV question on the exam expects an understanding of SQL Server T-SQL clause formatting. Some possible answers will provide the proper syntax, but not have required actors present such as (). For example, this is a possible solution answer: SELECT * FROM sys.dm_exec_sql_text<sql_handle>. However, examine this solution closely and notice the clause is not valid. In this case you should avoid the use of WHERE clauses as this is an argument not a search. Other common exam answers in this area provide a DMV in the answer that does not correspond to the problem at hand. In this case a long running query will not require the use of sys.dm_exec_query_optimizer_info, for example, as this DMV is related to database work tuning, not process running queries. Any variation of the hundreds of DMV categories are fair game on the exam, but you can optimize your test chances by ensuring answers follow the conventions of SQL T-SQL syntax, and the DMVs apply to the issue in the question.

Performance Monitoring With the System Monitor

The System Monitor available with Windows 2008 Server is another tool you can use to troubleshoot problems that are related to the server operating system and its hardware. The System Monitor has a large amount of available counters or classes of attributes that provide an array of detailed information. The installation of SQL Server 2008 on a Windows 2008 server automatically adds in a class of performance counters for use in analyzing parameters associated specifically with SQL Server. System Monitor is accessed via the Performance Monitor in the Administrative Tools program group. The System Monitor allows a collection of data in a snapshot mode and the trending of data and recording of it to database tables or file format. The use of System Monitor and its current incarnation in Windows 2008 Performance Monitor is an entire subject and science that eclipses the scope of this exam manual.

System monitor use is primarily related to trending memory use, CPU use, disk I/O, and the large amount of counter objects available with each specific Windows 2008 Server subsystem or installed applications.

Interpreting the results of System Monitoring data is the goal of the questions in this area of the exam. If you are monitoring server RAM and Process: Working Set of RAM shows Maximum use at 4 GB and your server only has 4 GB of RAM total then you know a problem exists. You must also correlate settings from SQL Server to what is observed in System Monitoring collected data. You become like a detective in that you must be able to pinpoint what performance issues are occuring and how they relate to SQL Server. Keep in mind that on the exam the primary hardware influences on SQL Server devolve to RAM, Disk speed, and CPU use.

Exam Prep: Using Performance Monitor to Optimize SQL Server
Your SQL Server 2008 server has 16 GB of RAM and supports several other applications.

SQL Server Memory Configuration
Min Server Memory = 4 GB
Max Server Memory =8 GB

The Sales group complains often of poor performance when they use other applications on the server. How can you improve performance without any hardware investments?

System Monitor Counters:

Counter Object	Minimum	Maximum
Process: Working Set	2 GB	3 GB
SQL Server: Buffer Manager: Buffer Cache Hit Ratio	90	97
SQL Server: Memory Manager: Total Server Memory (KB)	12 GB	16 GB

Exam Solution: Using Performance Monitor to Optimize SQL Server
Like so many exam questions always make sure you read the question closely. In this case the question does not ask you to do anything to increase SQL Server performance. This question poses a situation where SQL Server is performing fine, but causing other server based applications to suffer. The counters reveal that SQL Server is only using a maximum of 3 GB of memory, yet configured to take a minimum of 4 GB from the operating system's pool of memory. A simple decrease of the SQL Min Server Memory to 3 GB will allocate 1 GB back to the server's operating memory pool and make it available to other server applications. This type of question specifies that no hardware investment occur in the solution. One answer will invariably propose adding RAM, which is immediately ruled out. Any solutions that propose increasing resources to SQL Server are also wrong.

Domain 8 – Implementing High Availability

This collection of software and hardware based solutions for high availability combine to provide SQL Server the capability for continuous operations, without downtime. This exam domain expects fluency with each method, what situations to use them in, and how to implement and manage them.

SQL Server provides several methods to ensure continuous operations. Database mirroring is an inherent SQL Server 2008 solution, and it allows the replication of all transactions to a second database with monitoring to decide when it should take over. SQL Server is completely API aware of Microsoft Windows Server 2008 R2's cluster capabilities, and it will install as a native service. SQL Server clustering using the features of Windows 2008 R2 server provides the ability to protect against single server hardware failure, disk failure, or a failure of the SQL Server software bits. Transaction log shipping provides protection at the SQL Server layer by supporting database movement of all transactions to a second database, which is in a single use mode and ready to become the primary in case of failure. SQL replication supports the synchronization of data between databases in a consistent state. Data is distributed via replication to different SQL Server databases, and when necessary, to mobile clients.

Database Mirroring

SQL 2008 database mirroring is a database availability solution that literally mirrors all data from one server database to another, provided each database is on a different SQL Server instance. One server is a primary copy or one in use, and the other is termed a standby server. The SQL Server 2008 database mirroring software and administration functions provide for the synchronization of data between the two databases in such a way that no transaction is every dropped during the failover to the secondary standby database. The database mirror can also use a standby database copy without up to date synchronization, which can result in some data loss, but still provide a server ready to pick up database operations in the event of a failure to the primary.

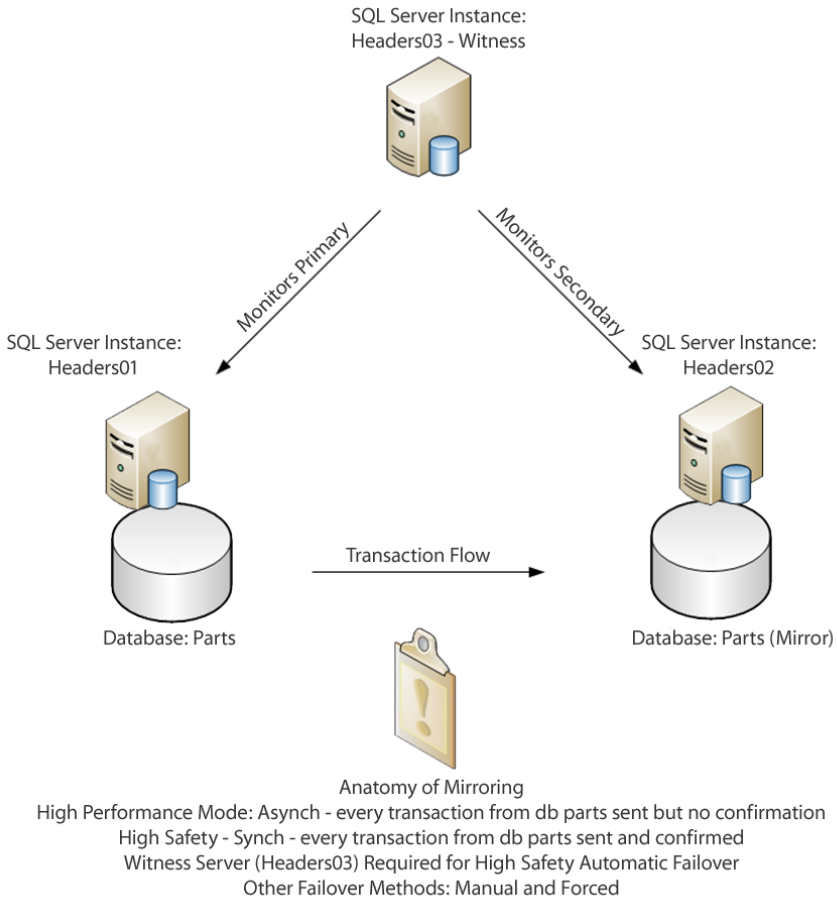


Figure 50: Database Mirroring Schematic

The figure above demonstrates the elements of a database mirroring configuration:

- Two SQL instances are identified, and then a database is selected for mirroring (in this case the db **Parts**).
- The mirroring wizard is run via SQL Server Management Studio, or you can use ALTER DB to setup mirroring via T-SQL.
- A mode is selected for the mirroring operation. In most cases, for complete recoverability, use the High Safety mode with automatic failover, which requires a witness server.
- The witness function works much like the witness server in Windows 2008 Server clustering and the quorum drive concept:
    - If the witness server loses connection to the primary mirror server (Headers01) and is connected to Headers02, a failover will occur (i.e., the second database server in the mirroring configuration will serve as the primary database instance).
    - When the server Headers01 comes back online it will receive a message from the witness server that Headers02 is now the primary server in the mirror setup, and this will cause Headers01 to become the mirror database and to synchronize transactions**.**

This is a very elegant solution and one that in production situations induces 2-5 seconds of connectivity for applications such as .NET web applications.

Exam Prep: SQL Server Database Mirroring
You manage multiple SQL Server 2008 servers for your company. You set up database mirroring. You use automatic failover. An unexplained error occurs in the primary database in the mirrored databases. The current database state is synchronizing. The witness state is not working. You need to act quickly to fail the mirror over, and data loss is acceptable.

Exam Solution: SQL Server Database Mirroring
This exam style question immediately should prompt you to know the method of failover is now the forced method. The forced method is one used to force failover, but does not guarantee that data loss will not occur. This method of forcing a service change is normally used in a disaster recovery situation, where the primary member of the mirror may never come back online. One other clue in the question pointing to the direness of the situation is that the current database state is synchronizing, which means they are not synchronized anymore.

Manual failovers require both the primary and mirror server to be online and the database to be synchronized. Replacing the mirror server is not a good solution at any point in SQL Server database mirroring operations.

There are many security considerations with SQL Server database mirroring. One upper level concern is if users are going to login to SQL Server databases with SQL credentials, then you must ensure you create all logins and database mappings in the mirror target SQL Server because you cannot mirror the master database on the primary mirror server, which houses all logins and mappings. Database object users and assignments will exist due to mirroring.

SQL Server Clusters

Microsoft SQL Server is a resource for clustering with Microsoft Windows 2008 R2 clustering services. This type of cluster provides complete failover in protecting the SQL Server instance from a failure due to any type of hardware failure. In a Microsoft Windows 2008 cluster implementation, the SQL Server instance uses a shared disk resource normally located on a SAN and is set up from the start as a cluster from the SQL Server installation. A SQL Server cluster consists of at least two servers configured to run an SQL instance that is configured to work on both servers. The SQL cluster implementation creates a virtual SQL Server with its own IP address and dedicated set of disk resources. The SQL Server cluster-aware installation will load the SQL bits to all servers identified in the cluster setup at the same time during installation. Windows quorum uses a quorum and witness concept to determine which cluster members are viable and can participate in supporting cluster resources. Windows clustering uses a heartbeat based monitoring system to constantly interrogate the status of cluster members. In best practice cases the Windows cluster node heartbeat network interface cards are dedicated to heartbeat operations and set up on a private VLAN for traffic segregation to ensure their function is not encumbered by normal Windows cluster server operations. The emergence of Windows 2008 Hyper-V virtualization services has added a new dimension to cluster operations with the capability of managing cluster members as virtual guests. Windows 2008 Hyper-V hosts can also be clustered at this time with Windows 2008 cluster services.

Test Tip: A SQL Server cluster using Windows 2008 R2 clustering features is managed from the Microsoft Windows 2008 R2 cluster administrator utility, NOT from SQL Server Configuration for tasks such as restarting the SQL Service.

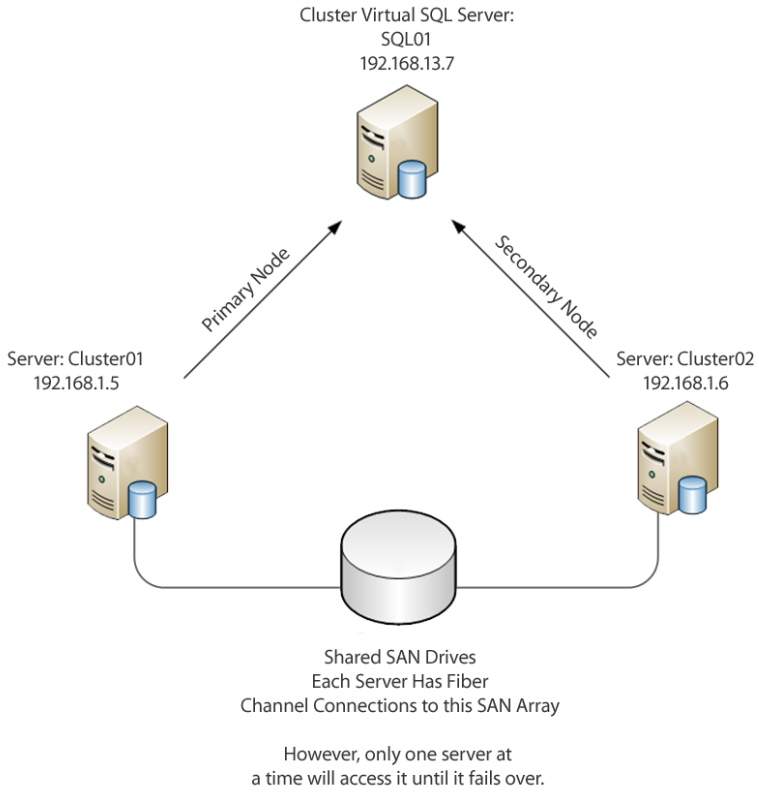


Figure 51: SQL Server Cluster Implementation

Figure 51 shows a simple Windows 2008 R2 SQL cluster with active/passive configuration. The Windows cluster is comprised of two nodes: Cluster01 and Cluster02. Once Windows 2008 R2 clustering is configured, a third virtual or cluster node is created, for example, ClusterNode. It is, however, comprised of the two physical servers listed above. This cluster can now provide failover for standard Windows services: DHCP, file sharing, disk drives, and other cluster available resources.

Once a cluster is established and functional, a SQL Server installation is possible that will detect the Windows cluster and allow the running of the setup path to establish a SQL cluster instance. This will create the virtual SQL cluster SQL01 with its own IP address. The SQL cluster instance will have several prerequisite resources to launch, the primary one being the disk drives that are cluster accessible and meet the definition of a shared cluster resource. The virtual SQL cluster instance named SQL01 is what clients and applications would target in their connection string definitions and for other database operations.

The failover process can identify which cluster nodes are available as possible owners of the cluster SQL instance. If Cluster02 was a failover host for Cluster01, and Cluster01 suffered a bluescreen/memory dump issue, then a failover of all SQL instances and services would occur to Cluster02.

This is amazing technology and represents a top-level solution for protecting SQL Server availability. The failover process also provides quite a bit of management support; you can update and patch servers in a cluster, as required, without downtime as you failover resources to allow for the updates to occur. Server reboots and other types of maintenance such as a RAM addition can also occur during the production cycle of a day due to failover capability.

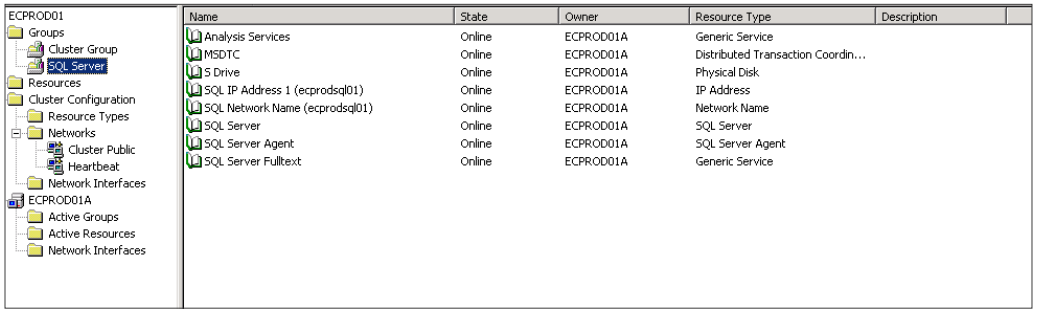


Figure 52: Cluster Resources for an SQL Server Cluster

Figure 52 shows a SQL Server cluster implementation with the established SQL Server cluster group and member cluster resources such as Analysis Service, DTC, SAN Drives, SQL Server virtual IP, SQL Server virtual name, the SQL Server service, SQL Server Agent, and full-text search. These cluster resources combine to form the SQL cluster instance.

Again, if the S Drive above is not online, then the SQL cluster will not start as it is a dependent resource for the SQL Server service. For security, each node in an SQL cluster will use the same service accounts to run the SQL Server service. This account should be protected and managed properly to ensure it does not lockout or become unavailable, as this will cause the SQL cluster to fail.

Exam Prep: SQL Server Clustering
You support six instances of SQL Server 2008 running on different computers configured as failover clusters. You are required to add a seventh SQL node to the Windows 2008 failover cluster. The least amount of effort should be used. How will you accomplish this?

Exam Solution: SQL Server Clustering
This type of question will present several solutions that are designed to test your knowledge of how to initiate a clustered SQL instance. To configure an SQL Server instance as a node in a cluster you must run the SQL Server 2008 setup. This will present you with the option of adding a node to a cluster. This is the same for removing a node from a cluster. The SQL Server Management Studio is of no use here, and neither is a T-SQL based solution like sp_configure.

Transaction Log Shipping

Transaction log shipping is an older SQL Server method of high availability but one that does not have the expense of SQL Server clustering. It does not have the rapid failover capability of SQL database mirroring but is very viable to ensure your SQL database transactions are replicated and available for use in the event of a problem to the database setup as the log shipping primary.

Log shipping automatically sends transactions from a primary database to a secondary database, on separate SQL Server instances. The log shipped data is continuously applied to the secondary database as it arrives. An option exists to use a third SQL instance to monitor the history and status of all log shipping operations between the primary and secondary database.

Test Tip: Log shipping secondary databases do not automatically come online for a failover operation if the primary database fails. You must start the secondary database with a full restore of the primary database backup using NORECOVERY or STANDBY.

Log shipping conducts a series of SQL jobs to achieve the movement of transactions. A backup job occurs on the primary database every 15 minutes, but this is configurable. A copy job copies the primary database backups to the secondary server. The job is called Log Shipping Copy and is created when log shipping is created on the secondary server. A restore job is also created on the secondary server, which contains the copied backup files, and all the relevant backup history.

During a log shipping recovery operation the following tasks are conducted:

- Apply any uncommitted backups from the backup file share established for log shipping.
- Apply any uncommitted transaction log backups to the secondary database.
- Conduct a tail log backup of the primary database if it is available.
- Recover the secondary database to place it in a consistent state, and bring it online.
- Advise clients of the database instance name where the previously primary database is now running.
- If desired, set up other secondary log shipping targets for the new primary database.

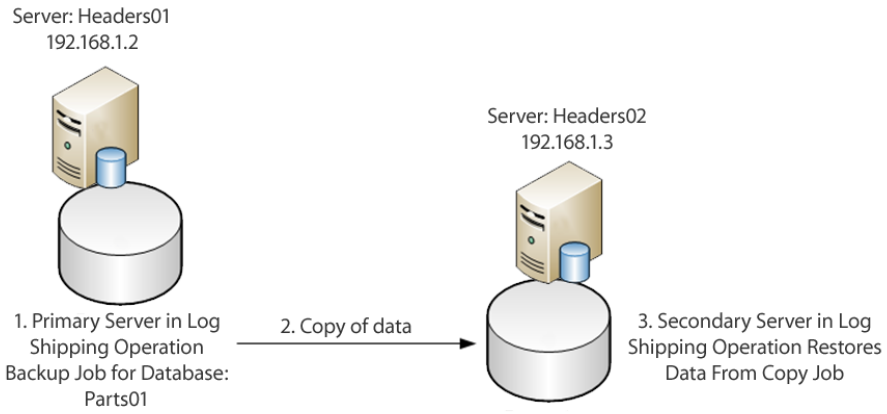


Figure 53: Log Shipping Schematic

Exam Prep: SQL Server Log Shipping
You manage several SQL Server 2008 servers. Headers02 is configured as a secondary log shipping server for Headers01. You are conducting maintenance on Headers01 and need to failover first to Headers02. What step should you perform first?

Exam Solution: SQL Server Log Shipping
In this case you would back up the transaction log on Headers01 with the NORECOVERY option, which will get the tail of the transaction log and allow you to bring the log shipping secondary database up to date with transactions it may not have from the previous logs shipped. This is planned cutover, so there is not a physical problem with the primary server, which would dictate different actions on the log shipping target server. Some exam solutions appear valid but ensure the above options which direct you to conduct a backup of Headers01 with the COPY ONLY option as this will not provide the tail log information needed by the secondary.

SQL Replication

SQL Replication provides SQL Server the capability of pushing data to multiple endpoints in a controlled and synchronized fashion. SQL Replication can replicate data between SQL Server and a client, such as a remote inventory system or mobile notebook users. In the high availability space, SQL Replication can replicate data from SQL instance to SQL instance. This can also include supporting data warehousing or reporting.

SQL Server replication uses a push or pull type model to replicate data predicated on standard taxonomies of publisher, subscriber, publications, articles and subscriptions. The publisher is a source database pushing data out to other instances. The distributor is a database store that houses transaction data about the replication process to include replication status, metadata, and in some cases queues data for subscribers. The publisher and distributor can be the same database. Subscribers are databases that receive the publisher's data. A subscriber can be bidirectional in that it passes data back and forth to publishers. An article is a database object in a publication. This can be views, tables, or indexes. Data published as articles supports SQL Replication filtering, which is important for determining the data composition of what is moved. A publication is a grouping or article, which can be from more than one database. Lastly, the subscription is a request for data from a subscriber. The subscription determines what will be received, when, and where. Subscriptions are push or pull type replication.

SQL Replication is a feature that supports the following types of publication:

- **Merge Publication** - A snapshot of the data occurs and then changes are tracked with triggers. When subscribers connect up for the changes, all changes are published since the previous replication.
- **Transactional Publication/Transactional Publication With Updatable Subscriptions** - A snapshot of the data occurs at the publication database and then any changes are delivered to the subscribers as they occur. With updatable subscriptions, changes made at the subscriber are applied back to the publisher.
- **Snapshot Replication** – Snapshots are point in time replications and no updates are tracked.

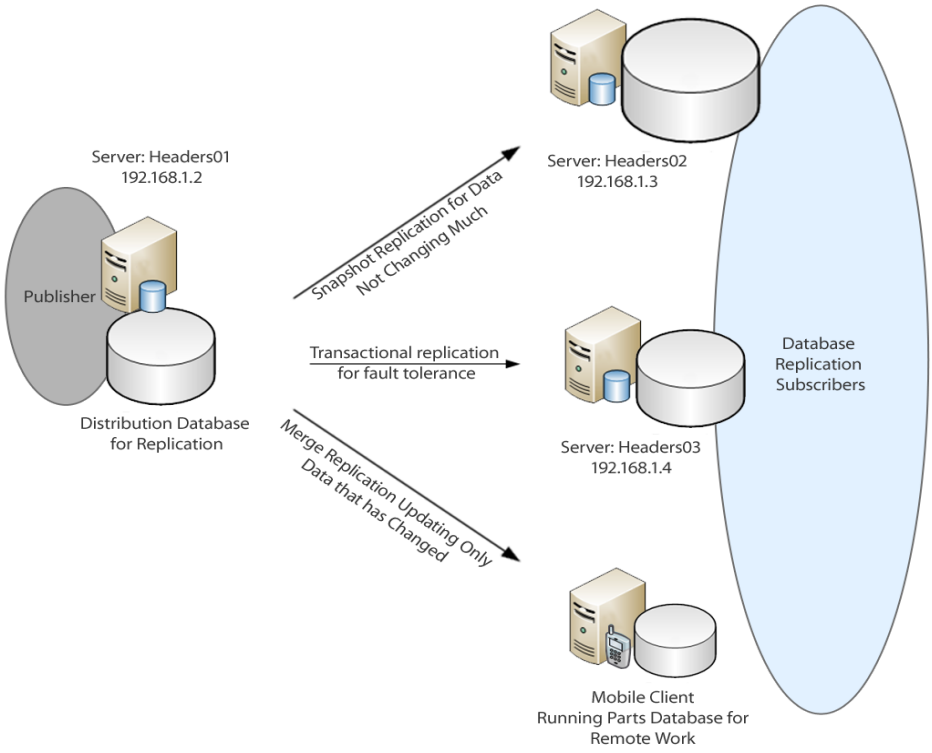


Figure 54: SQL Server Replication Using Multiple Methods of Replication

Exam Prep: SQL Server Replication
You manage multiple SQL Server 2008 instances. You have a main office in Atlanta and a branch office in Seattle. The branch office is connected to corporate via a WAN connection. There is one SQL Server 2008 instance per office. The SQL Server 2008 instance in Atlanta contains a database named Parts, and a table with 2 million rows named Parts_Stock. You set up merge replication between the Parts_Stock table and a table named Parts_Stock in the remote database instance located in Seattle. A WAN failure for two days causes the most recent replication merge to expire. You need to get the remote copy of Parts_Stock up to date as quickly as possible. You should complete this without much effort. How will you solve this problem?

Exam Solution: SQL Server Replication
In this case you will create a new subscription. A tablediff will identify the table changes and use the new subscription to apply the differences. You will not need to create a snapshot. This is the least cost in terms of bandwidth and effort for a solution. Exam answers that focus on the publication portion of the replication setup in this case expend too much effort because it is harder to set up the publication side than a subscription.

Practice Questions

Chapter 1

1. You work as the DBA for a medium-sized organization. One of the employees in the Accounting department has asked for your assistance. She needs a solution that will allow her to transfer data from a MySQL database into a SQL Server database. Additionally, the data must be modified in several ways during the process. Which SQL Server add-on component will meet her needs? Select the best answer.

   - ❍ A. SSAS
   - ❍ B. SSRS
   - ❍ C. SSIS
   - ❍ D. Database Mirroring

2. You are configuring database mail and have been asked to ensure that fault tolerance is provided in order to accommodate for random mail server outages. Which of the following actions can provide this fault tolerance? Select the best answer.

   - ❍ A. Set the logging level to verbose
   - ❍ B. Configure multiple mail profiles
   - ❍ C. Configure the account retry delay to 90 seconds
   - ❍ D. Set the database executable minimum lifetime to 900 seconds

3. You are using a mail server to send mail from SQL Server 2008's database mail feature. The mail server has intermittent problems that may cause it to be unavailable for 30 to 90 seconds. You cannot use any other mail servers. What database mail configuration parameters may help ensure that email messages are delivered? Choose TWO.

   - ❍ A. Account retry delay
   - ❍ B. Create additional profiles
   - ❍ C. Logging level
   - ❍ D. Account retry attempts

Chapter 2

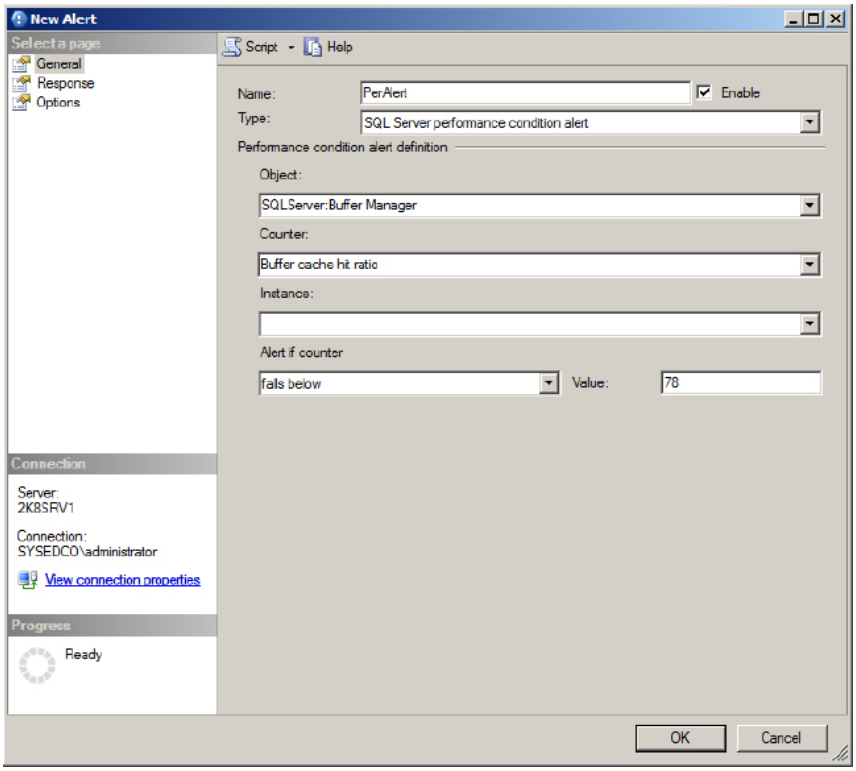
1. When you configure an alert in the SQL Server Agent, only two response options are provided. What are these TWO options? Choose TWO.

   - ❍ A. Execute job
   - ❍ B. Notify operator
   - ❍ C. Shutdown server
   - ❍ D. Check the integrity of a database

2. You are configuring an alert. Based on the screen in Exhibit 1, what is the value of buffer cache hit ratio that would fire the alert? Select the best answer.

- ❍ A. Less than 7.8 percent
- ❍ B. Less than 78 percent
- ❍ C. Greater than 78 percent
- ❍ D. Greater than 7.8 percent

Exhibit(s):



3. You manage a single SQL Server 2008 instance with a single database called Engineering that is in the Simple recovery mode. You have been creating Full backups every night for several months. Now, the database has grown so large that the full backups take more than two hours. You want to perform Full backups every weekend and a different type of backup nightly. You must use the built-in backup feature of SQL Server 2008. What kind of backup can you perform that cooperates with the weekend Full backups, but requires less time? Select the best answer.

- ❍ A. Incremental
- ❍ B. Modified
- ❍ C. Transaction Logs
- ❍ D. Differential

Chapter 3

1. You are the DBA for a large SQL Server 2008 database application. When you arrived at work this morning, you had several messages waiting from users. The users were complaining that they cannot access the data. They are receiving access denied errors. All users could access the data the previous day. At the end of the day yesterday, you performed the following tasks:-Manual backup of the database-Manual snapshot of the database-Modified the ownership of the call_routing table-Added seven new usersWhich action is the likely cause of the problem? Select the best answer.

   - A. Manual backup of the database
   - B. Manual snapshot of the database
   - C. Added seven new users
   - D. Modified the ownership of the call_routing table

2. You are running SQL Server 2008 Enterprise Edition. You want to enable a server-level audit procedure using SQL Server Audit. What THREE steps are required to implement a server-level audit procedure? Choose THREE.

   - A. Create a server audit specification
   - B. Enable the audit
   - C. Create an audit object
   - D. Create the server-level permission for the audit

3. You are installing a test server that will run SQL Server 2008 Enterprise Edition. The installation OS will be Windows Server 2008. You want to ensure that remote connections are not allowed. How will you configure the installation so that only local connections are allowed but outgoing network connections can be made? Select the best answer.

   - A. Unplug the network cable
   - B. Disable the TCP/IP provider
   - C. Block TCP port 21 in the firewall
   - D. Use the sp_configure 'Block Net-Conn', 1 command

Chapter 4

1. You are creating a backup plan. Part of the plan includes the backup of the transaction logs for your databases. All databases run on SQL Server 2008 Standard Edition and the OS used is Windows Server 2008. You plan to use the following code to backup the transaction logs: BACKUP LOG database_name TO DISK='logback.bak';In addition to backing up the log, what would this code do? Select the best answer.

   - A. Truncate the log
   - B. Backup the database
   - C. Restore the database
   - D. Truncate the database

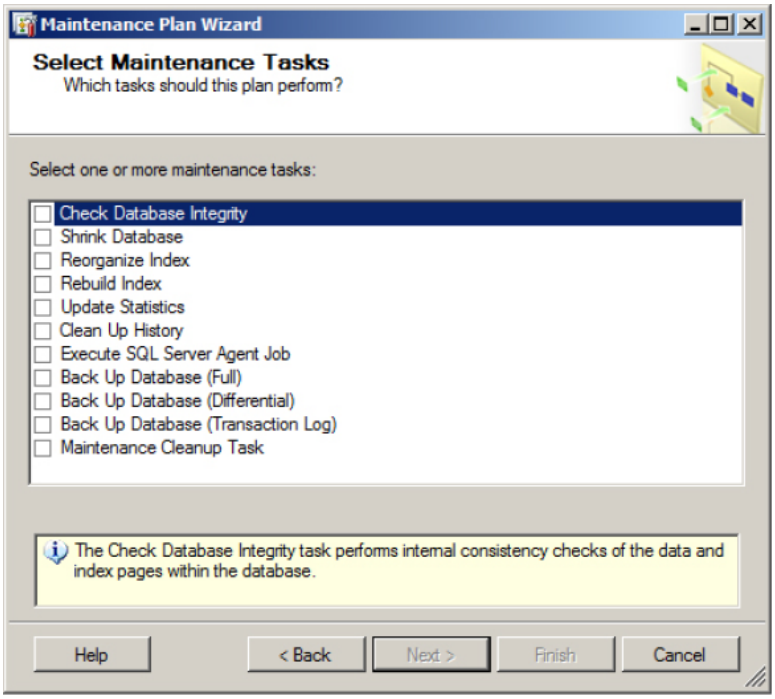
2. You are the DBA for a small company in Florida. You must restore a database named Sales to 3:17 yesterday afternoon. The current date is Wednesday, June 3, 2009. A full backup is made every night to a file named day-fback.bak. The transaction log is backed up each day at 12 PM and is backed up to a file named day-log.bak. Each week, the files are overwritten. Which THREE of the following four commands would accurately restore the database to the desired point-in-time if run in the proper order? Choose THREE answers.

- A. RESTORE LOG Sales FROM DISK='TUESDAY-log.bak' WITH NORECOVERY, STOPAT = 'Jun 2, 2009, 3:17 PM';
- B. RESTORE DATABASE Sales WITH RECOVERY;
- C. RESTORE DATABASE Sales FROM DISK='TUESDAY-fback.bak' WITH NORECOVERY;
- D. RESTORE LOG Sales FROM DISK='TUESDAY-log.bak' WITH NORECOVERY, UNTIL = 'Jun 2, 2009, 3:17 PM';

3. You are creating a Maintenance Plan using the wizard. You want to make sure the database is not corrupt and perform a complete backup of the database. In Exhibit 1, what options should you check? Choose TWO.

- A. Rebuild Index
- B. Back Up Database (Full)
- C. Shrink Database
- D. Check Database Integrity

Exhibit(s):



Chapter 5

1. You are the DBA for a carefully planned database. In spite of the intensive planning, the data files are still growing very large. You've been asked to look for a solution that reduces the storage space consumed by the database. The performance of the database is not an issue, as the users have been very happy with the performance; however, space is quickly running out on the server and the server administrator has indicated that more drive space cannot be purchased until the next budget cycle. What feature can be used to conserve storage space? Select the best answer.

   - ❍ A. Data compression
   - ❍ B. Transparent data encryption
   - ❍ C. Table partitioning
   - ❍ D. Filestreams

2. What kinds of pages exist in a SQL Server index? Choose THREE.

   - ❍ A. Intermediate pages
   - ❍ B. Leaf pages
   - ❍ C. Root pages
   - ❍ D. Branch pages

3. You are attempting to create an index with the WITH ONLINE = ON clause. However, it is not working. You are running the SQL Server instance on Windows Server 2008 Standard Edition and the SQL Server instance is running the Standard Edition of SQL Server 2008. Service pack 1 has not been applied to the Windows Server. Several users are connected to the database. Why is the index command not working? Select the best answer.

   - ❍ A. Because Windows Server 2008 Enterprise Edition is required
   - ❍ B. Because users are connected
   - ❍ C. Because service pack 1 has not been applied
   - ❍ D. Because SQL Server 2008 Enterprise Edition is required

4. A database uses the collation named French_CI_AI. Without looking at documentation, what would this collation name tell you by itself? Choose three.

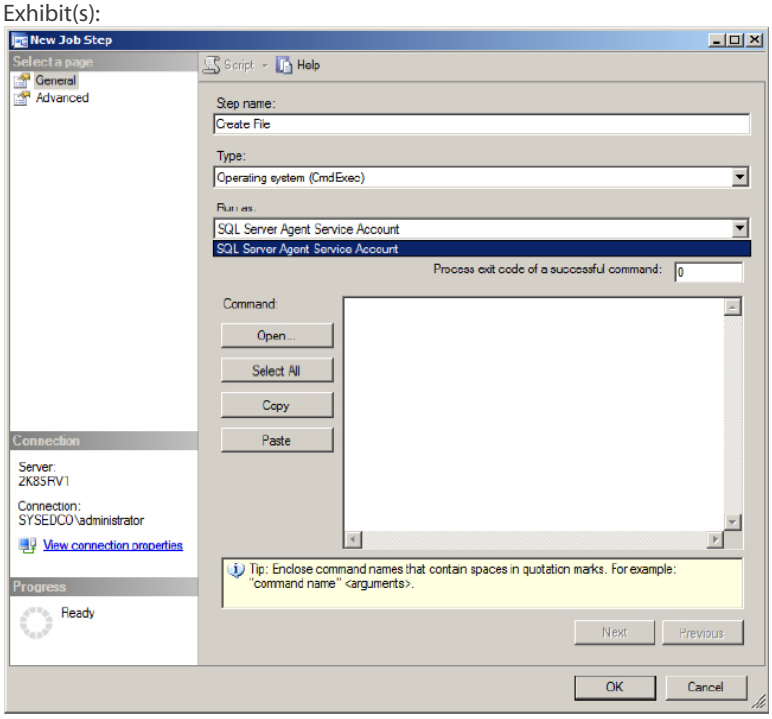
   - ❍ A. Case insensitivity
   - ❍ B. Accent insensitivity
   - ❍ C. The French codepage is used
   - ❍ D. Albanian icons

Chapter 6

1. You must configure a SQL Server Agent job to use a proxy account for access to the Windows file system. The job runs on a SQL Server 2008 Standard Edition server. You have taken the following steps:1. Right-click on the Credentials folder and select New Credential.2. In the New Credential windows enter a Credential name of OSCommand and select a user with appropriate permissions as the identity and provide a password.3. Click OK to create the Credential.When you attempt to create the job step and use the proxy account, you see the Run As option shown in Exhibit 1. What steps do you need to take to use the proxy account based on the OSCommand credential? Select the best answer.

   - ❍ A. You must use the CREATE JOB STEP command since proxies are not available in the GUI interface.
   - ❍ B. Proxies are only supported on Enterprise Edition.
   - ❍ C. Create a new Operating System (CmdExec) proxy in the Proxies container within the SQL Server Agent node.
   - ❍ D. The proxy account is configured as the owner of the job and not using the Run As parameter of a step.

Exhibit(s):



2. You are creating a proxy object for use with a SQL Server Agent job. You want a single object to provide all needed access for the job. The job runs several steps including the following actions:-Powershell-Batch files-VBScript scripts-Transact-SQLWhat SQL Server Agent subsystems should be enabled for the proxy object? Choose all that apply.
   - A. SQL
   - B. Operating System (CmdExec)
   - C. ActiveX Script
   - D. PowerShell

3. Which of the following tools can be used to see the most recent run of a SQL Server Agent job? Choose all that apply.
   - A. Configuration Manager
   - B. Job Activity Monitor
   - C. Job Manager
   - D. Log File Viewer

Chapter 7

1. Which of the following are valid recommendations that DTA can make for performance improvements after a workload analysis? Choose all that apply.
   - A. Adding indexes
   - B. Reserved memory recommendations
   - C. Deleting indexes
   - D. Partitioning tables

2. You want to schedule a Database Engine Tuning Advisor analysis to run during the night. You plan to implement a SQL Server Agent job. On what command line tool must the job call to perform the analysis? Select the best answer.
   - A. DETA.EXE
   - B. DTA.VBS
   - C. DETA.VBS
   - D. DTA.EXE

3. Which one of the following is NOT an item that you define within a SQL Server Profiler trace? Select the best answer.
   - A. Events
   - B. Stored Procedure
   - C. Data columns
   - D. Filters

4. You want to evaluate fragmentation statistics for each index in a database. Which DMV will you query? Select the best answer.
   - ❍ A. sys.dm_db_index_operational_stats
   - ❍ B. sys.dm_db_index_fragments
   - ❍ C. sys.dm_db_index_physical_stats
   - ❍ D. sys.dm_db_index_physical_fragments

Chapter 8

1. You are planning a database mirroring implementation. Which roles, related to mirroring, must be configured on separate SQL Server instances? Choose all that apply.
   - ❍ A. Primary
   - ❍ B. Witness
   - ❍ C. Mirror
   - ❍ D. Reflection

2. You are implementing replication. High availability is important. You've chosen to implement replication with the following options:-Immediate Updating Subscribers-Publisher-Distributor-Subscriber Which one should be changed or removed to allow for high availability?
   Select the best answer.
   - ❍ A. Publisher
   - ❍ B. Distributor
   - ❍ C. Immediate Updating Subscribers
   - ❍ D. Subscriber

Answers & Explanations

Chapter 1

1. Answer: C

Explanation A. Incorrect. SQL Server Analysis Services (SSAS) is used to analyze data and perform business intelligence operations.

Explanation B. Incorrect. SQL Server Reporting Services (SSRS) is used to generate reports against SQL Server data.

Explanation C. Correct. SQL Server Integration Services (SSIS) will provide extraction, transformation and loading.

Explanation D. Incorrect. Database mirroring is a high availability solution that creates an exact duplicate (mirror) of a database on a separate server.

2. Answer: B

Explanation A. Incorrect. Configuring the logging level to verbose will provide more information about a failure, but it does not provide for fault tolerance in any way.

Explanation B. Correct. Database mail allows for the configuration of multiple mail profiles so that a secondary or tertiary profile may be used if the first mail profile is not functioning.

Explanation C. Incorrect. Setting the account retry delay to 90 seconds will potentially give the mail server more time to become available again, but it does not create a fault tolerant scenario.

Explanation D. Incorrect. This setting determines how long the database mail send utility will remain in memory. Keeping it in memory longer prevents reloading it again and again, but it does not improve fault tolerance - even though it may improve performance.

3. Answers: A, D

Explanation A. Correct. The default for this setting is 60 seconds. In this scenario, setting the value to 90 seconds would increase the likelihood that the email could be sent on the second attempt.

Explanation B. Incorrect. Creating additional profiles will not help since you can only use the one mail server.

Explanation C. Incorrect. Configuring the logging level only determines the detail of the logs.

Explanation D. Correct. By default the account will be retried only once. By increasing this value, you ensure that the email will get out when the mail server becomes available.

Chapter 2

1. Answers: A, B

Explanation A. Correct. You can execute any pre-configured job when an alert fires.

Explanation B. Correct. You may notify an operator when an alert fires.

Explanation C. Incorrect. You may be able to shutdown the server through a job, but the alert can only call on the job.

Explanation D. Incorrect. You can check database integrity through a job, but the alert can only call on the job.

2. Answer: B

Explanation A. Incorrect. The correct response is less than 78 percent.

Explanation B. Correct. Anything less than 78 percent would fire this alert.

Explanation C. Incorrect. The correct response is less than 78 percent.

Explanation D. Incorrect. The correct response is less than 78 percent.

3. Answer: D

Explanation A. Incorrect. Incremental backups are not supported by the internal SQL Server backup tools.

Explanation B. Incorrect. No such backup feature exists in SQL Server.

Explanation C. Incorrect. Backing up the transaction logs for a database in the simple recovery mode would not provide full recoverability.

Explanation D. Correct. Differential backups are used to backup only changed data pages in the database.

Chapter 3

1. Answer: D

Explanation A. Incorrect. Backing up the database should not impact permissions.

Explanation B. Incorrect. Creating a snapshot should not impact permissions.

Explanation C. Incorrect. Adding users should not cause existing users to lose access.

Explanation D. Correct. The change in ownership has likely resulted in a break in the ownership change.

2. Answers: A, B, C

Explanation A. Correct. The server audit specification will specify the item to audit and map it to the audit object.

Explanation B. Correct. The final step is to enable the audit.

Explanation C. Correct. The audit object will define the target of the audit.

Explanation D. Incorrect. Permissions are not required since the audit procedure runs within the SQL Server database engine service.

3. Answer: B

Explanation A. Incorrect. This action would prevent outgoing network connections as well.

Explanation B. Correct. The TCP/IP provider enables remote connections. You can disable them by turning off the provider in the SQL Server Configuration Manager.

Explanation C. Incorrect. While this might prevent FTP traffic, it will do nothing to prevent SQL Server access on port 1433.

Explanation D. Incorrect. No such command exists.

Chapter 4

1. Answer: A

Explanation A. Correct. By default, without the use of the TRUNCATE_ONLY parameter, the log file is truncated after the backup completes. This truncation means that the backed up portion of the log is emptied for reuse.

Explanation B. Incorrect. The BACKUP LOG command never backs up the database.

Explanation C. Incorrect. The RESTORE DATABASE command is used to restore a database.

Explanation D. Incorrect. While the command will truncate the log file, it will not do so to the database.

2. Answers: A, B, C

Explanation A. Correct. This is the second step in the process.

Explanation B. Correct. This is the third step in the process.

Explanation C. Correct. This is the first step in the process.

Explanation D. Incorrect. The proper WITH parameter is STOPAT and not UNTIL.

3. Answers: B, D

Explanation A. Incorrect. Rebuild Index will drop the indexes and regenerate them.

Explanation B. Correct. Back UP Database (Full) will perform a complete backup of the database.

Explanation C. Incorrect. Shrink Database will shrink the physical size of the database file based on free space within the file.

Explanation D. Correct. Check Database Integrity will perform a DBCC CECKDB operation against selected databases.

Chapter 5

1. Answer: A

Explanation A. Correct. As long as the server is running Enterprise Edition, data compression can be used. Data compression is not available in Standard Edition.

Explanation B. Incorrect. TDE does not conserve storage space.

Explanation C. Incorrect. Partitioning does not conserve storage space.

Explanation D. Incorrect. Filestreams do not conserve storage space.

2. Answers: A, B, C

Explanation A. Correct. The intermediate pages contain branches leading to other intermediate pages or leaf pages.

Explanation B. Correct. The leaf pages contain the actual data stored in the index.

Explanation C. Correct. Each index contains one and only one root page.

Explanation D. Incorrect. While a B-tree structure is used, the pages are not called branch pages.

3. Answer: D

Explanation A. Incorrect. SQL Server 2008 Enterprise Edition is required, but Windows Server 2008 Standard Edition can run the Enterprise Edition of SQL Server.

Explanation B. Incorrect. If SQL Server 2008 Enterprise Edition were in use, the users could be connected and the command could run.

Explanation C. Incorrect. SP1 is not needed for online operations; however the Enterprise Edition of SQL Server 2008 is required for online operations.

Explanation D. Correct. Online index operations require SQL Server 2008 Enterprise Edition.

4. Answers: A, B, C

Explanation A. Correct. The CI stands for case insensitivity.

Explanation B. Correct. The AI stands for accent insensitivity.

Explanation C. Correct. The word French clearly denotes the French language.

Explanation D. Incorrect. AI does not stand for Albanian icons, it stands for accent insensitivity.

Chapter 6

1. Answer: C

Explanation A. Incorrect. A proxy object must be created that is based on the credentials configured in these three steps.

Explanation B. Incorrect. Proxies are supported on both Enterprise and Standard editions.

Explanation C. Correct. Proxies are based on credentials. Creating the credential is only the first step. Next, you need to expand the SQL Server Agent node and then the Proxies node. Now, right-click on the Operating System (CmdExec) node and selectn New Proxy. Enter the proxy name, credential name of OSCommand and check Operating System (CmdExec) in the Subsystem section. Click OK to create the proxy.

Explanation D. Incorrect. The proxy account is configured using the Run As parameter; however, a credential is the foundation for a proxy object. The proxy object must be created in the Proxies node of the SQL Server Agent node.

2. Answers: B, C, D

Explanation A. Incorrect. Transact-SQL commands are handled internally without the need for a proxy.

Explanation B. Correct. The batch files will require CmdExec to operate.

Explanation C. Correct. The VBScript code will require the ActiveX Script subsystem to function.

Explanation D. Correct. The PowerShell subsystem is used to execute PowerShell commands.

3. Answers: B, D

Explanation A. Incorrect. The SSCM is used to configure SQL Server services and it does not show information about job histories.

Explanation B. Correct. The Job Activity Monitor can be used to see that last run and whether it was successful or not.

Explanation C. Incorrect. No such utility exists in SQL Server.

Explanation D. Correct. The Log File Viewer is displayed when you right-click a job and select View History.

Chapter 7

1. Answers: A, C, D

Explanation A. Correct. One of the simplest ways to improve database performance is to add appropriate queries. DTA can recommend additional indexes.

Explanation B. Incorrect. DTA makes no recommendations related to instance configuration.

Explanation C. Correct. You can often improve performance by deleting indexes that are not being utilized. DTA can recommend dropping indexes.

Explanation D. Correct. Table partitioning can be recommended by DTA, though it is not by default.

2. Answer: D

Explanation A. Incorrect. The acronym for the Database Engine Tuning Advisor is DTA and not DETA. Therefore the command is DTA.EXE.

Explanation B. Incorrect. The command is an executable (DTA.EXE) and not a script.

Explanation C. Incorrect. The command is an executable (DTA.EXE) and not a script.

Explanation D. Correct. With DTA.EXE you can configure an analysis process within a job.

3. Answer: B

Explanation A. Incorrect. You select events when building the trace.

Explanation B. Correct. You can capture information about stored procedures, but they are not part of the trace configuration.

Explanation C. Incorrect. You can select specific data columns related to the events when building the trace.

Explanation D. Incorrect. You can create filters when building the trace.

4. Answer: C

Explanation A. Incorrect. This DMV reports locking, latching and access stats for the indexes and not fragmentation.

Explanation B. Incorrect. No such DMV exists.

Explanation C. Correct. This DMV will report fragmentation statistics.

Explanation D. Incorrect. No such DMV exists.

Chapter 8

1. Answers: A, B, C

Explanation A. Correct. The Primary role must exist in a separate instance from the Mirror or Witness roles.

Explanation B. Correct. The Witness role must exist in a separate instance from the Primary and Mirror roles.

Explanation C. Correct. The Mirror role must exist in a separate instance from the Primary and Witness roles.

Explanation D. Incorrect. No such role exists in a mirror relationship.

2. Answer: C

Explanation A. Incorrect. The publisher is required. Without it, you will have no replication at all.

Explanation B. Incorrect. The distributor is required. It may be the same server as the publisher, but it is required.

Explanation C. Correct. If high availability is important, you should use queued updating subscribers, which can process transactions even if the publisher is unavailable.

Explanation D. Incorrect. If you have no subscribers, you have no need for replication.