Microsoft

OFFICIAL MICROSOFT LEARNING PRODUCT

10775A

Administering Microsoft® SQL Server® 2012 Database

Product Number: 10775A

Part Number: X18-29125

Released: 05/2012

MICROSOFT LICENSE TERMS
OFFICIAL MICROSOFT LEARNING PRODUCTS
MICROSOFT OFFICIAL COURSE Pre-Release and Final Release Versions

---

These license terms are an agreement between Microsoft Corporation and you. Please read them. They apply to the Licensed Content named above, which includes the media on which you received it, if any. These license terms also apply to any updates, supplements, internet based services and support services for the Licensed Content, unless other terms accompany those items. If so, those terms apply.

BY DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT DOWNLOAD OR USE THE LICENSED CONTENT.

If you comply with these license terms, you have the rights below.

---

1. **DEFINITIONS.**

   a. "Authorized Learning Center" means a Microsoft Learning Competency Member, Microsoft IT Academy Program Member, or such other entity as Microsoft may designate from time to time.

   b. "Authorized Training Session" means the Microsoft-authorized instructor-led training class using only MOC Courses that are conducted by a MCT at or through an Authorized Learning Center.

   c. "Classroom Device" means one (1) dedicated, secure computer that you own or control that meets or exceeds the hardware level specified for the particular MOC Course located at your training facilities or primary business location.

   d. "End User" means an individual who is (i) duly enrolled for an Authorized Training Session or Private Training Session, (ii) an employee of a MPN Member, or (iii) a Microsoft full-time employee.

   e. "Licensed Content" means the MOC Course and any other content accompanying this agreement. Licensed Content may include (i) Trainer Content, (ii) sample code, and (iii) associated media.

   f. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program, and (iii) holds a Microsoft Certification in the technology that is the subject of the training session.

   g. "Microsoft IT Academy Member" means a current, active member of the Microsoft IT Academy Program.

   h. "Microsoft Learning Competency Member" means a Microsoft Partner Network Program Member in good standing that currently holds the Learning Competency status.

   i. "Microsoft Official Course" or "MOC Course" means the Official Microsoft Learning Product instructor-led courseware that educates IT professionals or developers on Microsoft technologies.

j. "Microsoft Partner Network Member" or "MPN Member" means a silver or gold-level Microsoft Partner Network program member in good standing.

k. "Personal Device" means one (1) device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular MOC Course.

l. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.

m. "Trainer Content" means the trainer version of the MOC Course and additional content designated solely for trainers to use to teach a training session using a MOC Course. Trainer Content may include Microsoft PowerPoint presentations, instructor notes, lab setup guide, demonstration guides, beta feedback form and trainer preparation guide for the MOC Course. To clarify, Trainer Content does not include virtual hard disks or virtual machines.

2. **INSTALLATION AND USE RIGHTS**. The Licensed Content is licensed not sold. The Licensed Content is licensed on a one copy per user basis, such that you must acquire a license for each individual that accesses or uses the Licensed Content.

2.1 Below are four separate sets of installation and use rights. Only one set of rights apply to you.

a. **If you are a Authorized Learning Center:**

i. If the Licensed Content is in digital format for each license you acquire you may either:

1. install one (1) copy of the Licensed Content in the form provided to you on a dedicated, secure server located on your premises where the Authorized Training Session is held for access and use by one (1) End User attending the Authorized Training Session, or by one (1) MCT teaching the Authorized Training Session, or
2. install one (1) copy of the Licensed Content in the form provided to you on one (1) Classroom Device for access and use by one (1) End User attending the Authorized Training Session, or by one (1) MCT teaching the Authorized Training Session.

ii. You agree that:

1. you will acquire a license for each End User and MCT that accesses the Licensed Content,
2. each End User and MCT will be presented with a copy of this agreement and each individual will agree that their use of the Licensed Content will be subject to these license terms prior to their accessing the Licensed Content. Each individual will be required to denote their acceptance of the EULA in a manner that is enforceable under local law prior to their accessing the Licensed Content,
3. for all Authorized Training Sessions, you will only use qualified MCTs who hold the applicable competency to teach the particular MOC Course that is the subject of the training session,
4. you will not alter or remove any copyright or other protective notices contained in the Licensed Content,

5. you will remove and irretrievably delete all Licensed Content from all Classroom Devices and servers at the end of the Authorized Training Session,
6. you will only provide access to the Licensed Content to End Users and MCTs,
7. you will only provide access to the Trainer Content to MCTs, and
8. any Licensed Content installed for use during a training session will be done in accordance with the applicable classroom set-up guide.

b. **If you are a MPN Member.**

i. If the Licensed Content is in digital format for each license you acquire you may either:

1. install one (1) copy of the Licensed Content in the form provided to you on (A) one (1) Classroom Device, or (B) one (1) dedicated, secure server located at your premises where the training session is held for use by one (1) of your employees attending a training session provided by you, or by one (1) MCT that is teaching the training session, **or**
2. install one (1) copy of the Licensed Content in the form provided to you on one (1) Classroom Device for use by one (1) End User attending a Private Training Session, or one (1) MCT that is teaching the Private Training Session.

ii. You agree that:

1. you will acquire a license for each End User and MCT that accesses the Licensed Content,
2. each End User and MCT will be presented with a copy of this agreement and each individual will agree that their use of the Licensed Content will be subject to these license terms prior to their accessing the Licensed Content. Each individual will be required to denote their acceptance of the EULA in a manner that is enforceable under local law prior to their accessing the Licensed Content,
3. for all training sessions, you will only use qualified MCTs who hold the applicable competency to teach the particular MOC Course that is the subject of the training session,
4. you will not alter or remove any copyright or other protective notices contained in the Licensed Content,
5. you will remove and irretrievably delete all Licensed Content from all Classroom Devices and servers at the end of each training session,
6. you will only provide access to the Licensed Content to End Users and MCTs,
7. you will only provide access to the Trainer Content to MCTs, and
8. any Licensed Content installed for use during a training session will be done in accordance with the applicable classroom set-up guide.

c. **If you are an End User:**

You may use the Licensed Content solely for your personal training use. If the Licensed Content is in digital format, for each license you acquire you may (i) install one (1) copy of the Licensed Content in the form provided to you on one (1) Personal Device and install another copy on another Personal Device as a backup copy, which may be used only to reinstall the Licensed Content; or (ii) print one (1) copy of the Licensed Content. You may not install or use a copy of the Licensed Content on a device you do not own or control.

d. **If you are a MCT.**

i. For each license you acquire, you may use the Licensed Content solely to prepare and deliver an Authorized Training Session or Private Training Session. For each license you acquire, you may install and use one (1) copy of the Licensed Content in the form provided to you on one (1) Personal Device and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Licensed Content. You may not install or use a copy of the Licensed Content on a device you do not own or control.

ii. **Use of Instructional Components in Trainer Content.** You may customize, in accordance with the most recent version of the MCT Agreement, those portions of the Trainer Content that are logically associated with instruction of a training session. If you elect to exercise the foregoing rights, you agree: (a) that any of these customizations will only be used for providing a training session, (b) any customizations will comply with the terms and conditions for Modified Training Sessions and Supplemental Materials in the most recent version of the MCT agreement and with this agreement. For clarity, any use of "*customize"* refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

2.2 **Separation of Components.** The Licensed Content components are licensed as a single unit and you may not separate the components and install them on different devices.

2.3 **Reproduction/Redistribution Licensed Content**. Except as expressly provided in the applicable installation and use rights above, you may not reproduce or distribute the Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.

2.4 **Third Party Programs**. The Licensed Content may contain third party programs or services. These license terms will apply to your use of those third party programs or services, unless other terms accompany those programs and services.

2.5 **Additional Terms**. Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to that respective component and supplements the terms described in this Agreement.

3. **PRE-RELEASE VERSIONS.** If the Licensed Content is a pre-release ("**beta**") version, in addition to the other provisions in this agreement, then these terms also apply:

a. **Pre-Release Licensed Content.** This Licensed Content is a pre-release version. It may not contain the same information and/or work the way a final version of the Licensed Content will. We may change it for the final version. We also may not release a final version. Microsoft is under no obligation to provide you with any further content, including the final release version of the Licensed Content.

b. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software, technologies, or products to third parties because we include your feedback in them. These rights

survive this agreement.

c. **Term**. If you are an Authorized Training Center, MCT or MPN, you agree to cease using all copies of the beta version of the Licensed Content upon (i) the date which Microsoft informs you is the end date for using the beta version, or (ii) sixty (60) days after the commercial release of the Licensed Content, whichever is earliest ("**beta term**"). Upon expiration or termination of the beta term, you will irretrievably delete and destroy all copies of same in the possession or under your control.

4. **INTERNET-BASED SERVICES**. Classroom Devices located at Authorized Learning Center's physical location may contain virtual machines and virtual hard disks for use while attending an Authorized Training Session. You may only use the software on the virtual machines and virtual hard disks on a Classroom Device solely to perform the virtual lab activities included in the MOC Course while attending the Authorized Training Session. Microsoft may provide Internet-based services with the software included with the virtual machines and virtual hard disks. It may change or cancel them at any time. If the software is pre-release versions of software, some of its Internet-based services may be turned on by default. The default setting in these versions of the software do not necessarily reflect how the features will be configured in the commercially released versions. If Internet-based services are included with the software, they are typically simulated for demonstration purposes in the software and no transmission over the Internet takes place. However, should the software be configured to transmit over the Internet, the following terms apply:

a. **Consent for Internet-Based Services.** The software features described below connect to Microsoft or service provider computer systems over the Internet. In some cases, you will not receive a separate notice when they connect. You may switch off these features or not use them. By using these features, you consent to the transmission of this information. Microsoft does not use the information to identify or contact you.

b. **Computer Information.** The following features use Internet protocols, which send to the appropriate systems computer information, such as your Internet protocol address, the type of operating system, browser and name and version of the software you are using, and the language code of the device where you installed the software. Microsoft uses this information to make the Internet-based services available to you.

- **Accelerators**. When you use click on or move your mouse over an Accelerator, the title and full web address or URL of the current webpage, as well as standard computer information, and any content you have selected, might be sent to the service provider. If you use an Accelerator provided by Microsoft, the information sent is subject to the Microsoft Online Privacy Statement, which is available at go.microsoft.com/fwlink/?linkid=31493. If you use an Accelerator provided by a third party, use of the information sent will be subject to the third party's privacy practices.

- **Automatic Updates.** This software contains an Automatic Update feature that is on by default. For more information about this feature, including instructions for turning it off, see go.microsoft.com/fwlink/?LinkId=178857. You may turn off this feature while the software is running ("opt out"). Unless you expressly opt out of this feature, this feature will (a) connect to Microsoft or service provider computer systems over the Internet, (b) use Internet protocols to send to the appropriate systems standard computer information, such as your computer's Internet protocol address, the type of operating system, browser and name and version of the software you are using, and the language code of the device where you installed the software, and (c) automatically download and install, or prompt you to download and/or install, current Updates to the software. In some cases, you will not receive a separate notice before this feature takes effect.

By installing the software, you consent to the transmission of standard computer information and the automatic downloading and installation of updates.

- **Auto Root Update**. The Auto Root Update feature updates the list of trusted certificate authorities. you can switch off the Auto Root Update feature.
- **Customer Experience Improvement Program (CEIP), Error and Usage Reporting; Error Reports**. This software uses CEIP and Error and Usage Reporting components enabled by default that automatically send to Microsoft information about your hardware and how you use this software. This software also automatically sends error reports to Microsoft that describe which software components had errors and may also include memory dumps. You may choose not to use these software components. For more information please go to <http://go.microsoft.com/fwlink/?LinkID=196910>.
- **Digital Certificates**. The software uses digital certificates. These digital certificates confirm the identity of Internet users sending X.509 standard encrypted information. They also can be used to digitally sign files and macros, to verify the integrity and origin of the file contents. The software retrieves certificates and updates certificate revocation lists. These security features operate only when you use the Internet.
- **Extension Manager**. The Extension Manager can retrieve other software through the internet from the Visual Studio Gallery website. To provide this other software, the Extension Manager sends to Microsoft the name and version of the software you are using and language code of the device where you installed the software. This other software is provided by third parties to Visual Studio Gallery. It is licensed to users under terms provided by the third parties, not from Microsoft. Read the Visual Studio Gallery terms of use for more information.
- **IPv6 Network Address Translation (NAT) Traversal service (Teredo)**. This feature helps existing home Internet gateway devices transition to IPv6. IPv6 is a next generation Internet protocol. It helps enable end-to-end connectivity often needed by peer-to-peer applications. To do so, each time you start up the software the Teredo client service will attempt to locate a public Teredo Internet service. It does so by sending a query over the Internet. This query only transfers standard Domain Name Service information to determine if your computer is connected to the Internet and can locate a public Teredo service. If you

    · use an application that needs IPv6 connectivity or
    · configure your firewall to always enable IPv6 connectivity

  by default standard Internet Protocol information will be sent to the Teredo service at Microsoft at regular intervals. No other information is sent to Microsoft. You can change this default to use non-Microsoft servers. You can also switch off this feature using a command line utility named “netsh”.
- **Malicious Software Removal**. During setup, if you select “Get important updates for installation”, the software may check and remove certain malware from your device. “Malware” is malicious software. If the software runs, it will remove the Malware listed and updated at www.support.microsoft.com/?kbid=890830. During a Malware check, a report will be sent to Microsoft with specific information about Malware detected, errors, and other information about your device. This information is used to improve the software and other Microsoft products and services. No information included in these reports will be used to identify or contact you. You may disable the software’s reporting functionality by following the instructions found at

www.support.microsoft.com/?kbid=890830. For more information, read the Windows Malicious Software Removal Tool privacy statement at go.microsoft.com/fwlink/?LinkId=113995.

- **Microsoft Digital Rights Management.** If you use the software to access content that has been protected with Microsoft Digital Rights Management (DRM), then, in order to let you play the content, the software may automatically request media usage rights from a rights server on the Internet and download and install available DRM updates. For more information, see go.microsoft.com/fwlink/?LinkId=178857.
- **Microsoft Telemetry Reporting Participation**. If you choose to participate in Microsoft Telemetry Reporting through a "basic" or "advanced" membership, information regarding filtered URLs, malware and other attacks on your network is sent to Microsoft. This information helps Microsoft improve the ability of Forefront Threat Management Gateway to identify attack patterns and mitigate threats. In some cases, personal information may be inadvertently sent, but Microsoft will not use the information to identify or contact you. You can switch off Telemetry Reporting. For more information on this feature, see http://go.microsoft.com/fwlink/?LinkId=130980.
- **Microsoft Update Feature**. To help keep the software up-to-date, from time to time, the software connects to Microsoft or service provider computer systems over the Internet. In some cases, you will not receive a separate notice when they connect. When the software does so, we check your version of the software and recommend or download updates to your devices. You may not receive notice when we download the update. You may switch off this feature.
- **Network Awareness**. This feature determines whether a system is connected to a network by either passive monitoring of network traffic or active DNS or HTTP queries. The query only transfers standard TCP/IP or DNS information for routing purposes. You can switch off the active query feature through a registry setting.
- **Plug and Play and Plug and Play Extensions**. You may connect new hardware to your device, either directly or over a network. Your device may not have the drivers needed to communicate with that hardware. If so, the update feature of the software can obtain the correct driver from Microsoft and install it on your device. An administrator can disable this update feature.
- **Real Simple Syndication ("RSS") Feed**. This software start page contains updated content that is supplied by means of an RSS feed online from Microsoft.
- **Search Suggestions Service**. When you type a search query in Internet Explorer by using the Instant Search box or by typing a question mark (?) before your search term in the Address bar, you will see search suggestions as you type (if supported by your search provider). Everything you type in the Instant Search box or in the Address bar when preceded by a question mark (?) is sent to your search provider as you type it. In addition, when you press Enter or click the Search button, all the text that is in the search box or Address bar is sent to the search provider. If you use a Microsoft search provider, the information you send is subject to the Microsoft Online Privacy Statement, which is available at go.microsoft.com/fwlink/?linkid=31493. If you use a third-party search provider, use of the information sent will be subject to the third party's privacy practices. You can turn search suggestions off at any time in Internet Explorer by using Manage Add-ons under the Tools button. For more information about the search suggestions service, see go.microsoft.com/fwlink/?linkid=128106.
- **SQL Server Reporting Services Map Report Item.** The software may include features that retrieve content such as maps, images and other data through the Bing Maps (or successor branded)

application programming interface (the "Bing Maps APIs"). The purpose of these features is to create reports displaying data on top of maps, aerial and hybrid imagery. If these features are included, you may use them to create and view dynamic or static documents. This may be done only in conjunction with and through methods and means of access integrated in the software. You may not otherwise copy, store, archive, or create a database of the content available through the Bing Maps APIs. you may not use the following for any purpose even if they are available through the Bing Maps APIs:

- Bing Maps APIs to provide sensor based guidance/routing, or
- Any Road Traffic Data or Bird's Eye Imagery (or associated metadata).

Your use of the Bing Maps APIs and associated content is also subject to the additional terms and conditions at http://www.microsoft.com/maps/product/terms.html.

- **URL Filtering**. The URL Filtering feature identifies certain types of web sites based upon predefined URL categories, and allows you to deny access to such web sites, such as known malicious sites and sites displaying inappropriate or pornographic materials. To apply URL filtering, Microsoft queries the online Microsoft Reputation Service for URL categorization. You can switch off URL filtering. For more information on this feature, see http://go.microsoft.com/fwlink/?LinkId=130980
- **Web Content Features**. Features in the software can retrieve related content from Microsoft and provide it to you. To provide the content, these features send to Microsoft the type of operating system, name and version of the software you are using, type of browser and language code of the device where you run the software. Examples of these features are clip art, templates, online training, online assistance and Appshelp. You may choose not to use these web content features.
- **Windows Media Digital Rights Management**. Content owners use Windows Media digital rights management technology (WMDRM) to protect their intellectual property, including copyrights. This software and third party software use WMDRM to play and copy WMDRM-protected content. If the software fails to protect the content, content owners may ask Microsoft to revoke the software's ability to use WMDRM to play or copy protected content. Revocation does not affect other content. When you download licenses for protected content, you agree that Microsoft may include a revocation list with the licenses. Content owners may require you to upgrade WMDRM to access their content. Microsoft software that includes WMDRM will ask for your consent prior to the upgrade. If you decline an upgrade, you will not be able to access content that requires the upgrade. You may switch off WMDRM features that access the Internet. When these features are off, you can still play content for which you have a valid license.
- **Windows Media Player**. When you use Windows Media Player, it checks with Microsoft for
    - compatible online music services in your region;
    - new versions of the player; and
    - codecs if your device does not have the correct ones for playing content.

  You can switch off this last feature. For more information, go to www.microsoft.com/windows/windowsmedia/player/11/privacy.aspx.
- **Windows Rights Management Services.** The software contains a feature that allows you to create content that cannot be printed, copied or sent to others without your permission. For more information, go to www.microsoft.com/rms. you may choose not to use this feature

- **Windows Time Service**. This service synchronizes with time.windows.com once a week to provide your computer with the correct time. You can turn this feature off or choose your preferred time source within the Date and Time Control Panel applet. The connection uses standard NTP protocol.
- **Windows Update Feature**. You may connect new hardware to the device where you run the software. Your device may not have the drivers needed to communicate with that hardware. If so, the update feature of the software can obtain the correct driver from Microsoft and run it on your device. You can switch off this update feature.

c. **Use of Information.** Microsoft may use the device information, error reports, and malware reports to improve our software and services. We may also share it with others, such as hardware and software vendors. They may use the information to improve how their products run with Microsoft software.

d. **Misuse of Internet-based Services.** You may not use any Internet-based service in any way that could harm it or impair anyone else's use of it. You may not use the service to try to gain unauthorized access to any service, data, account or network by any means.

5. **SCOPE OF LICENSE**. The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
   - install more copies of the Licensed Content on devices than the number of licenses you acquired;
   - allow more individuals to access the Licensed Content than the number of licenses you acquired;
   - publicly display, or make the Licensed Content available for others to access or use;
   - install, sell, publish, transmit, encumber, pledge, lend, copy, adapt, link to, post, rent, lease or lend, make available or distribute the Licensed Content to any third party, except as expressly permitted by this Agreement.
   - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation;
   - access or use any Licensed Content for which you are not providing a training session to End Users using the Licensed Content;
   - access or use any Licensed Content that you have not been authorized by Microsoft to access and use; or
   - transfer the Licensed Content, in whole or in part, or assign this agreement to any third party.

6. **RESERVATION OF RIGHTS AND OWNERSHIP**. Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content. You may not remove or obscure any copyright, trademark or patent notices that appear on the Licensed Content or any components thereof, as delivered to you.

7. **EXPORT RESTRICTIONS**. The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, End Users and end use. For additional information, see www.microsoft.com/exporting.

8. **LIMITATIONS ON SALE, RENTAL, ETC. AND CERTAIN ASSIGNMENTS**. You may not sell, rent, lease, lend or sublicense the Licensed Content or any portion thereof, or transfer or assign this agreement.

9. **SUPPORT SERVICES**. Because the Licensed Content is "as is", we may not provide support services for it.

10. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon any termination of this agreement, you agree to immediately stop all use of and to irretrievable delete and destroy all copies of the Licensed Content in your possession or under your control.

11. **LINKS TO THIRD PARTY SITES**. You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.

12. **ENTIRE AGREEMENT.** This agreement, and the terms for supplements, updates and support services are the entire agreement for the Licensed Content.

13. **APPLICABLE LAW.**
    a. United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

    b. Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.

14. **LEGAL EFFECT**. This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

15. **DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS," "WITH ALL FAULTS," AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT CORPORATION AND ITS RESPECTIVE AFFILIATES GIVE NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS UNDER OR IN RELATION TO THE LICENSED CONTENT. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT CORPORATION AND ITS RESPECTIVE AFFILIATES EXCLUDE ANY IMPLIED WARRANTIES OR CONDITIONS, INCLUDING THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**

16. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. TO THE EXTENT NOT PROHIBITED BY LAW, YOU CAN RECOVER FROM MICROSOFT CORPORATION AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO USD$5.00. YOU AGREE NOT TO SEEK TO RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES FROM MICROSOFT CORPORATION AND ITS RESPECTIVE SUPPLIERS.**

This limitation applies to

- anything related to the Licensed Content, services made available through the Licensed Content, or content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 $ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.
Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence , aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers ; et
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Revised March 2012

Welcome!

Thank you for taking our training! We've worked together with our Microsoft Certified Partners for Learning Solutions and our Microsoft IT Academies to bring you a world-class learning experience—whether you're a professional looking to advance your skills or a student preparing for a career in IT.

- **Microsoft Certified Trainers and Instructors**—Your instructor is a technical and instructional expert who meets ongoing certification requirements. And, if instructors are delivering training at one of our Certified Partners for Learning Solutions, they are also evaluated throughout the year by students and by Microsoft.

- **Certification Exam Benefits**—After training, consider taking a Microsoft Certification exam. Microsoft Certifications validate your skills on Microsoft technologies and can help differentiate you when finding a job or boosting your career. In fact, independent research by IDC concluded that 75% of managers believe certifications are important to team performance[1]. Ask your instructor about Microsoft Certification exam promotions and discounts that may be available to you.

- **Customer Satisfaction Guarantee**—Our Certified Partners for Learning Solutions offer a satisfaction guarantee and we hold them accountable for it. At the end of class, please complete an evaluation of today's experience. We value your feedback!

We wish you a great learning experience and ongoing success in your career!

Sincerely,

Microsoft Learning
www.microsoft.com/learning

Microsoft | Learning

[1] *IDC,* Value of Certification: Team Certification and Organizational Performance, *November 2006*

Acknowledgments

Microsoft Learning would like to acknowledge and thank the following for their contribution towards developing this title. Their effort at various stages in the development has ensured that you have a good classroom experience.

Design and Development

This course was designed and developed by SolidQ. SolidQ is a global provider of consulting, mentoring and training services for Microsoft Data Management, Business Intelligence and Collaboration platforms.

Greg Low – Lead Developer

Dr Greg Low is a SQL Server MVP, an MCT, and a Microsoft Regional Director for Australia. Greg has worked with SQL Server since version 4.2 as an active mentor, consultant and trainer. Greg describes himself as a SQL Server junkie and also describes himself as having been involved in development since dinosaurs roamed the Earth. He has been an instructor in the Microsoft SQL Server Masters certification program for several years and was one of the first two people to achieve the SQL Server 2008 Master certification. He is the author of a number whitepapers on the Microsoft MSDN and TechNet web sites and the author of a number of SQL Server related books. Greg is based in Melbourne Australia.

Herbert Albert – Course Developer

Herbert Albert started his career in 1994. He works as a trainer, consultant, and author focusing on SQL Server technologies. Herbert is a mentor and the Central European CEO for SolidQ. He is based in Vienna, Austria. He has several Microsoft certifications including MCT which he has held since 1997. Herbert is a regular speaker at conferences and co-author of the SQL Server 2012 Upgrade Technical Reference Guide and SQL Server 2005 Step-by-Step Applied Techniques. Together with Gianluca Hotz, Herbert writes a regular column at the SolidQ Journal.

Mark Hions – Technical Reviewer

Mark's passion for computing and skill as a communicator were well suited to his position as instructor at Honeywell Canada, where he started working with minicomputers, mainframes and mature students in 1984. He first met Microsoft SQL Server when it ran on OS/2, and has delivered training on every version since. An independent MCT and consultant for many years, he is a highly-rated presenter at TechEd, has designed SQL Server exams for Microsoft, and has delivered deep dive courses through the Microsoft Partner Channel. Mark is now the Principal SQL Server Instructor and Consultant at DesTech, which is the largest provider of SQL Server training in the Toronto area.

Chris Barker – Technical Reviewer

Chris Barker is an MCT working in the New Zealand market and currently employed as a staff trainer at Auldhouse, one of New Zealand's major CPLS training centers in Wellington. Chris' background includes programming from the early 1970s—his first program was written in assembly language and debugged in binary (literally)! While focusing training on programming (mostly .NET) and databases (mostly Microsoft SQL Server) Chris has also been an infrastructure trainer and has both Novell and Microsoft networking qualifications.

Contents

Module 20: Troubleshooting Common SQL Server 2012 Administrative Issues

Appendix A: Core Concepts in SQL Server High Availability and Replication

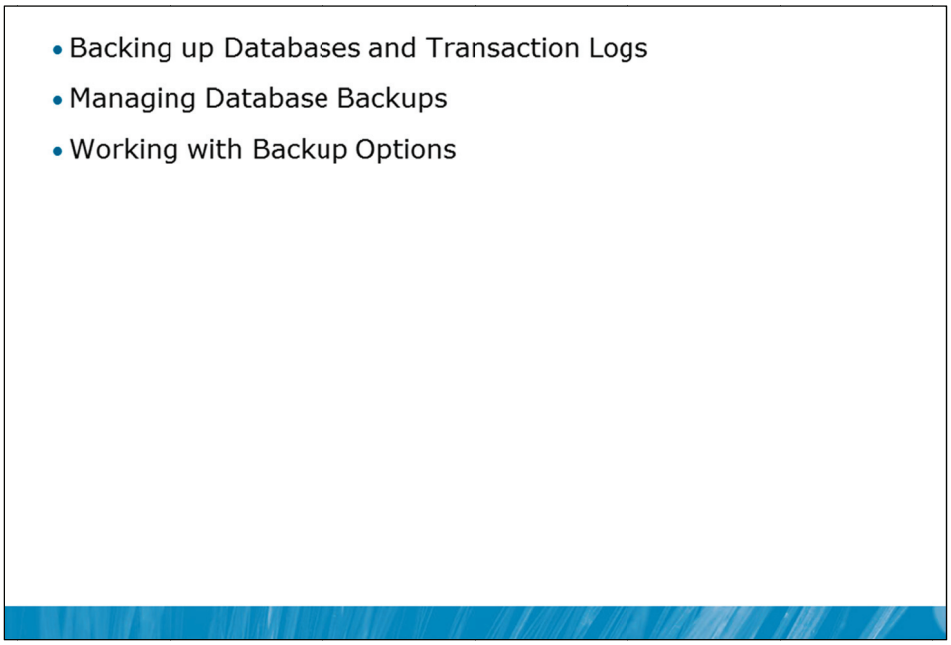
Appendix: Lab Answer Keys

Module 6

Backup of SQL Server 2012 Databases

Contents:

Module Overview



Ensuring reliable backups of corporate data is one of the most important roles for database administrators. You have seen that Microsoft® SQL Server® provides many types of backups. In this module, you will explore most of these backup types in more depth, and learn to implement the backups.

Apart from learning to perform full database backups, differential database backups, and transaction log backups, you will also see how to apply options that affect the way that the backups work. Automating and scheduling backups will be covered in later modules.

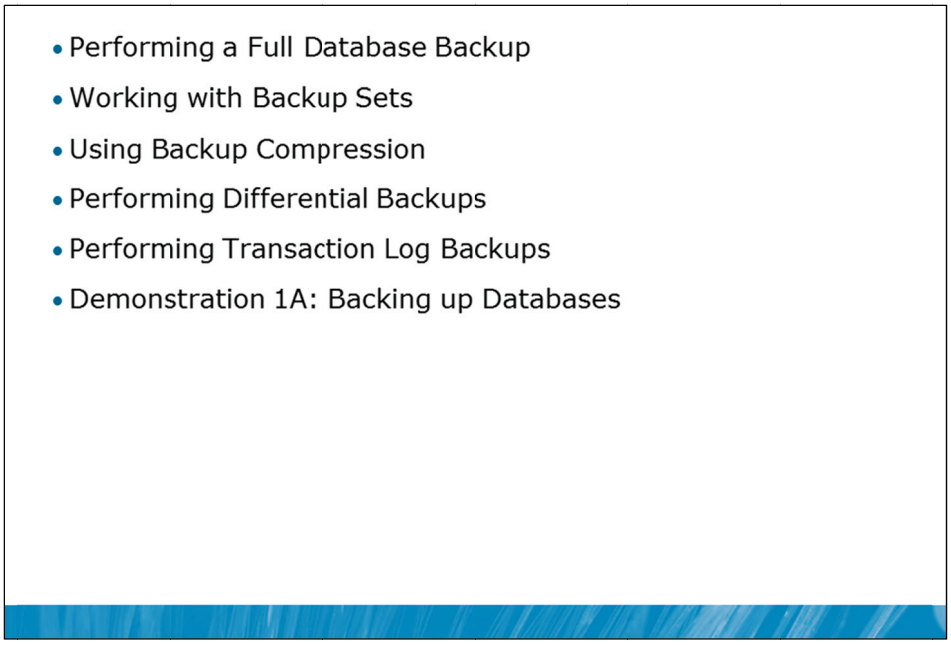
Objectives

After completing this lesson, you will be able to:

- Back up databases and transaction logs.
- Manage database backups.
- Work with more advanced backup options.

Lesson 1

Backing up Databases and Transaction Logs



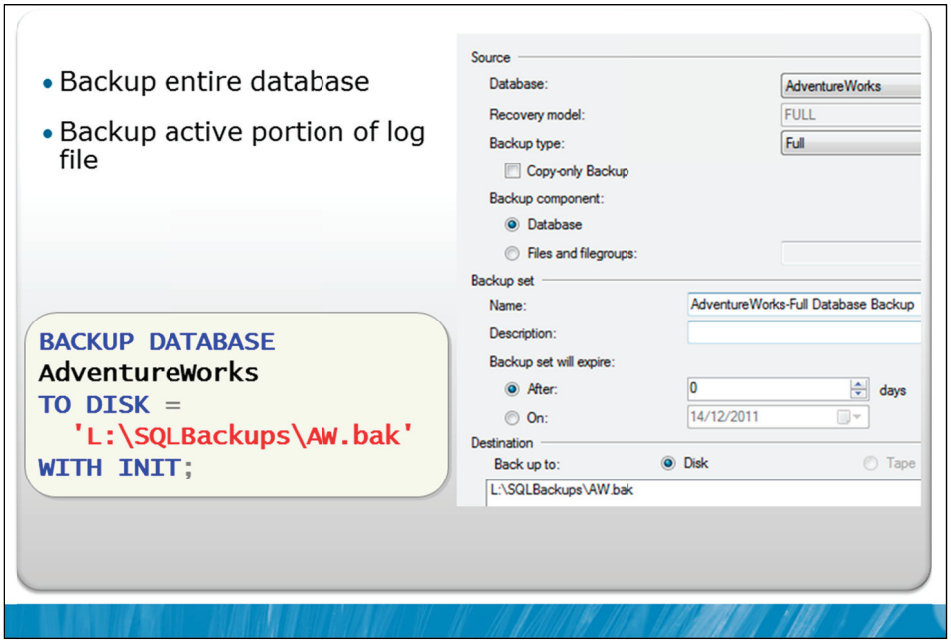
In the previous module, you saw how to plan a backup strategy for a SQL Server system. This lesson shows how to perform the most common forms of SQL Server backup: full database backups, differential database backups, and transaction log backups.

Objectives

After completing this lesson, you will be able to:

- Perform a full database backup.
- Work with backup sets.
- Use backup compression.
- Perform differential backups.
- Perform transaction log backups.

Performing a Full Database Backup



Key Points

Full database backups can be made using the BACKUP DATABASE command in T-SQL or using the GUI in SSMS. A full database backup saves all the data pages in the database, and also saves the active portion of the transaction log.

Example on the Slide

In the example on the slide, a full database backup is being made of the AdventureWorks database, to a disk file L:\SQLBackups\AW.bak. The option INIT that has been included in the command, instructs SQL Server to create the file if it does not already exist, and to overwrite the file if it does already exist.

The default initialization option, NOINIT, tells SQL Server to create the file if it does not already exist, and to append to the file if it already contains a SQL Server backup. The BACKUP DATABASE command includes many other options. The most important options will be discussed throughout this module.

Backup Timing

An important consideration when making a backup is to understand the timing associated with the contents of the backup. The database may be in use while the backup is occurring.

For example, if a backup starts at 10PM and finishes at 1AM, does the backup contain a copy of the database as it was at 10PM, a copy of the database as it was at 1AM, or a copy of the database from a time between the start and finish?

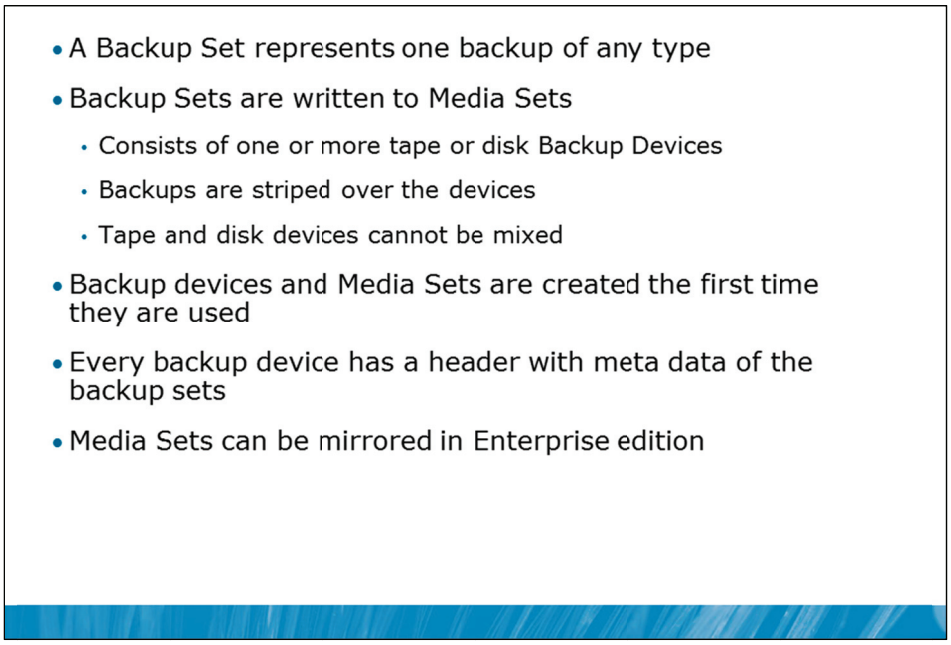
In early versions of SQL Server, the backup process wrote data pages to the backup device in sequence. However, if a user needed to modify a data page, SQL Server pushed that data page to the beginning of the backup page queue, and made the user wait for the page to be written to the backup device. In those versions of SQL Server, the backup made was a copy of the database at the time the backup was started.

To reduce the impact on users, later versions of SQL Server write all data pages to the backup device in sequence, but uses the transaction log to track any pages that are modified while the backup is occurring.

SQL Server then writes the relevant portion of the transaction log to the end of the backup. This process makes the backups slightly larger than in earlier versions, particularly if heavy update activities are happening at the same time as the backup. This altered process also means that the backup contains a copy of the database as at a time just prior to the completion of the backup, not as at the time the backup was started.

Question: What happens when you do not specify either INIT or NOINIT and a backup already exists on the backup device?

Working with Backup Sets



Key Points

Users are often confused by the fact that more than a single SQL Server backup can be contained within a single operating system file. The most common error related to this is to restore the first backup from a file, while assuming that it is the latest backup in the file.

A single backup is called a backup set, and is written to a media set, which itself can contain up to 64 backup devices. A backup device can be a disk or tape device. Tape devices must be locally attached and backups written to tape can be combined with Windows backups.

Note Tape drives that are mapped across a network cannot be used directly with SQL Server backup. Further, the ability for SQL Server to backup directly to tape is deprecated and will be removed in a future version of SQL Server.

Disk-based devices are the most commonly used. If a media set spans several backup devices, the backups will be striped across the devices.

Note No parity device is used while striping. If two backup devices are used together, each receives half the backup. Both must also be present when attempting to restore the backup.

Every backup operation to a media set must write to the same number and same types of backup devices. The Enterprise Edition of SQL Server also supports mirroring of media sets to improve the probability of being able to restore the backup. The same backup image is written to multiple locations.

Media sets and the backup devices are created the first time a backup is attempted on them. Media sets can also be named at the time of creation.

Backups created on a single non-mirrored device or a set of mirrored devices in a media set are referred to as a media family. The number of backup devices used for the media set determines the number of media families in a media set. For example, if a media set uses two non-mirrored backup devices, the media set contains two media families.

FORMAT Option

SQL Server has been designed to minimize the chance of inadvertent data loss.

As an example, consider a full database backup that has been written to two files, using the command:

```
BACKUP DATABASE AdventureWorks
TO DISK = 'D:\SQLBackups\AW_1.bak',
   DISK = 'L:\SQLBackups\AW_2.bak'
WITH INIT;
```

The two disk files that are listed make up a media set. The data from the backup is striped across the two files. Another backup could be made at a later time, to the same with a command such as the following:

```
BACKUP DATABASE AdventureWorks
TO DISK = 'D:\SQLBackups\AW_1.bak',
   DISK = 'L:\SQLBackups\AW_2.bak'
WITH NOINIT;
```

The data from the second backup would again be striped across the two files and the header of the media set updated to indicate that it now contains the two backups.

However, if a user then tries to create another backup with a command such as:

```
BACKUP DATABASE AdventureWorksDW
TO DISK = 'D:\SQLBackups\AW_1.bak';
```

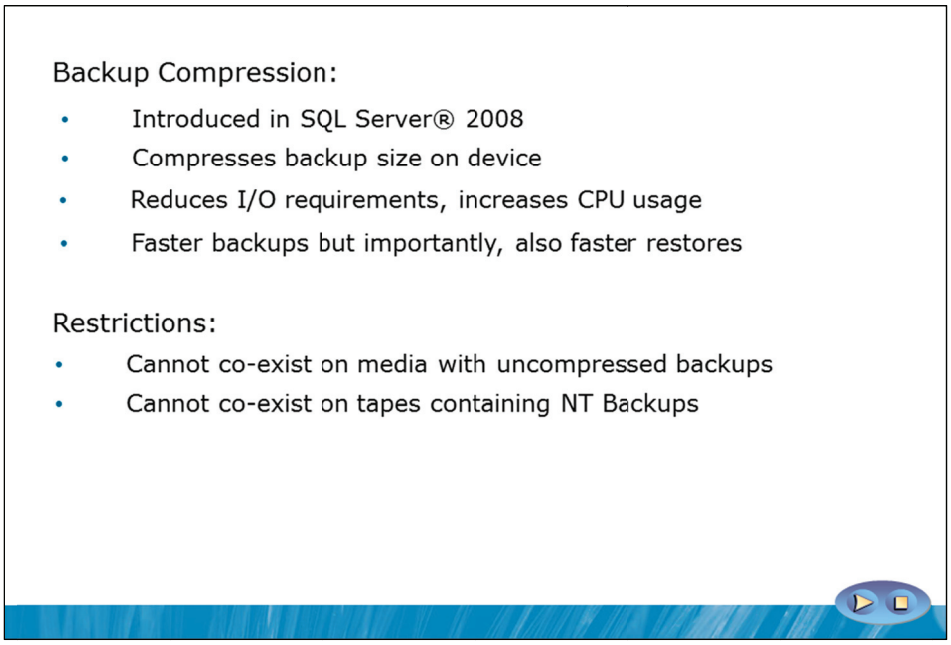
SQL Server would return an error. Before the member of the media set could be overwritten, the FORMAT option would need to be added to the WITH clause in the backup command:

```
BACKUP DATABASE AdventureWorksDW
TO DISK = 'D:\SQLBackups\AW_1.bak'
WITH FORMAT, INIT;
```

Use the FORMAT option to overwrite the contents of a backup file and split up the media set, but use the FORMAT option very carefully. Formatting one backup file of a media set renders the entire backup set unusable.

> **Question:** What advantage could striping backups to more than one backup device on a disk provide?

Using Backup Compression



Key Points

A number of compression-related technologies were introduced into SQL Server in SQL Server 2008. Backup compression trades off some CPU performance against a potentially large reduction in the size of a backup and increased backup and restore performance. Backup compression can be configured as a server option or as part of a T-SQL BACKUP command as shown:

```
BACKUP DATABASE AdventureWorksDW
TO DISK = 'D:\SQLBackups\AW_1.bak'
WITH FORMAT, COMPRESSION;
```

Performance Impact of Compressed Backups

Because a compressed backup is smaller than an uncompressed backup of the same amount of data, compressing a backup typically reduces the amount of device I/O required and decreases duration of backups significantly.

Any form of compression tends to increase CPU usage, and the additional CPU resources that are consumed by the compression process could adversely impact concurrent operations on systems that are CPU bound. Most current SQL Server systems are I/O bound, rather than CPU bound. The benefit received from the reduction in I/O usually outweighs the increase in CPU requirements by a significant factor.

Note In systems where CPU load is a concern, it is also possible to create a low-priority session for creating compressed backups by limiting CPU usage with the SQL Server Resource Governor. The use of Resource Governor is an advanced topic that is out of scope for this course.

Recovery Time

While a reduction in the time taken to perform backups is beneficial, backups are usually performed while the system is being used. However, compression benefits not only the backup process but also the restore process, and can significantly improve the ability to meet RTO requirements.

Compression Percentages

The degree of compression that is achieved depends entirely upon how compressible the data in the database is. Some data compresses well: other data does not compress well. A reduction in I/O and backup size of 30 to 50 percent is not uncommon in typical business systems.

Note While backup compression can be used on a database that has been encrypted using Transparent Database Encryption (TDE), the compression rate will be minimal. TDE is an advanced topic that is out of scope for this course.

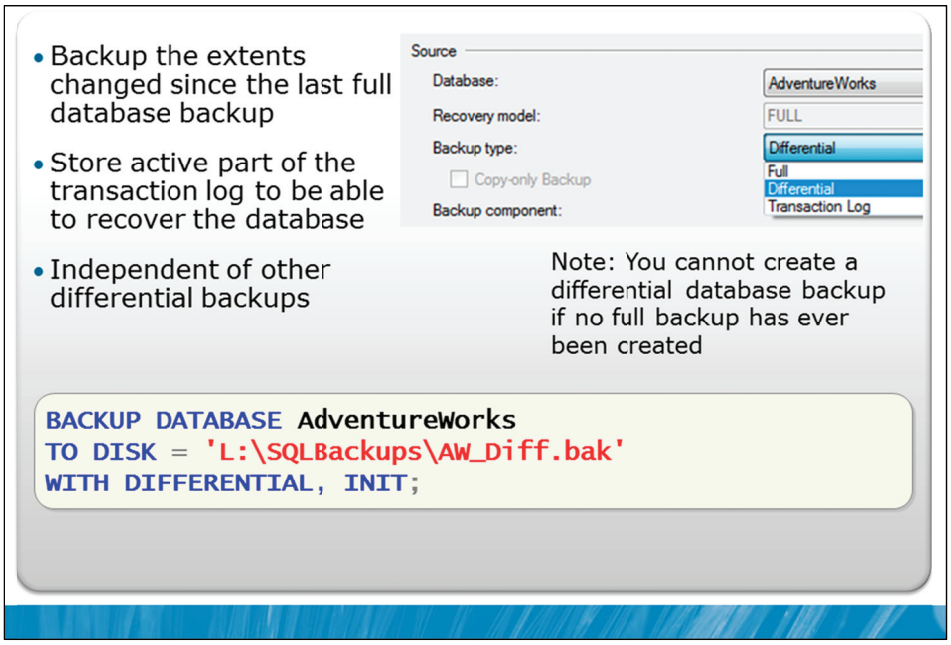
Restrictions on Backup Compression

The following restrictions apply to compressed backups:

- Compressed and uncompressed backups cannot co-exist in a media set.
- Previous versions of SQL Server cannot read compressed backups but lower editions of the product can restore compressed backups, even though the lower editions cannot create compressed backups.
- Windows-based backups cannot share a media set with compressed SQL Server backups.
- The default setting for backup compression can be set by the server configuration option 'backup compression default'.
- SQL Server 2008 R2 introduced the creation of compressed backups to the Standard Edition of SQL Server.

  Question: Why would both backup and restore time generally decrease when backup compression is used?

Performing Differential Backups



Key Points

While full database backups are ideal, there often is not enough time to perform full database backups. For situations where a relatively small percentage of the database is modified, compared to the overall database size, differential backups are a good option to consider.

Differential Backups

You can perform a differential backup using SQL Server Management Studio, or by adding the DIFFERENTIAL option to the BACKUP DATABASE T-SQL command.

SQL Server maintains a map of modified extents called the differential bitmap page. One page is maintained for every 4GB section of every data file. Each time a full database backup is created, SQL Server clears the map. As the data in the data files is modified, SQL Server updates this map. A differential backup saves all extents that have been modified since the last full database backup, not only those modified since the last differential backup.

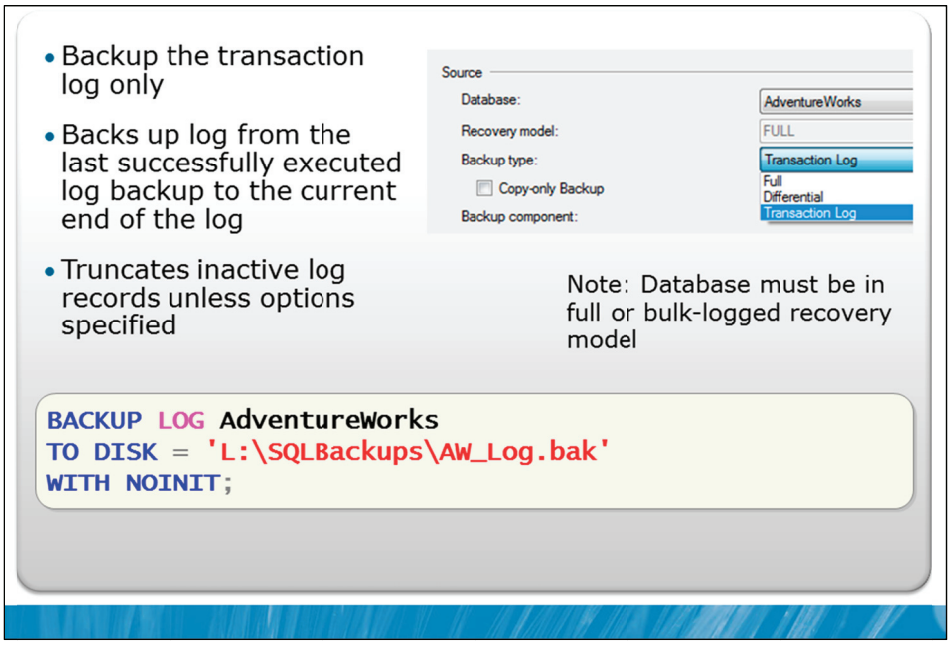
Note You cannot create a differential database backup unless a full database backup has been taken first.

A differential backup also saves the active portion of the transaction log, in exactly the same way that a full database backup does.

The syntax for the differential backup is identical to the syntax for full database backups, apart from the addition of the DIFFERENTIAL option. All other options that are available for full database backups are also available for differential backups.

Question: Does a differential backup truncate the transaction log?

Performing Transaction Log Backups



Key Points

You can perform a Transaction Log Backup using SQL Server Management Studio or by using the BACKUP LOG T-SQL statement. Before a transaction log backup can be performed, the database must be in either full or bulk-logged recovery model. In addition, a transaction log backup can only occur when a full database backup has been taken at some time prior.

A transaction log backup does not save any data pages from the database, except when the database is set to bulk logged recovery model. A transaction log backup finds the MaxLSN of the last successful transaction log backup, and saves all log entries beyond that point to the current MaxLSN. The transaction log is then truncated as far as is possible. The longest running active transaction must be retained, in case the database needs to be recovered after a failure.

Log Record Chains

Before a database can be restored using transaction log backups, an unbroken chain of log records must be available since the last full database backup to the desired point of restoration. If the chain is broken, it is only possible to restore up to the point where the backup chain was broken.

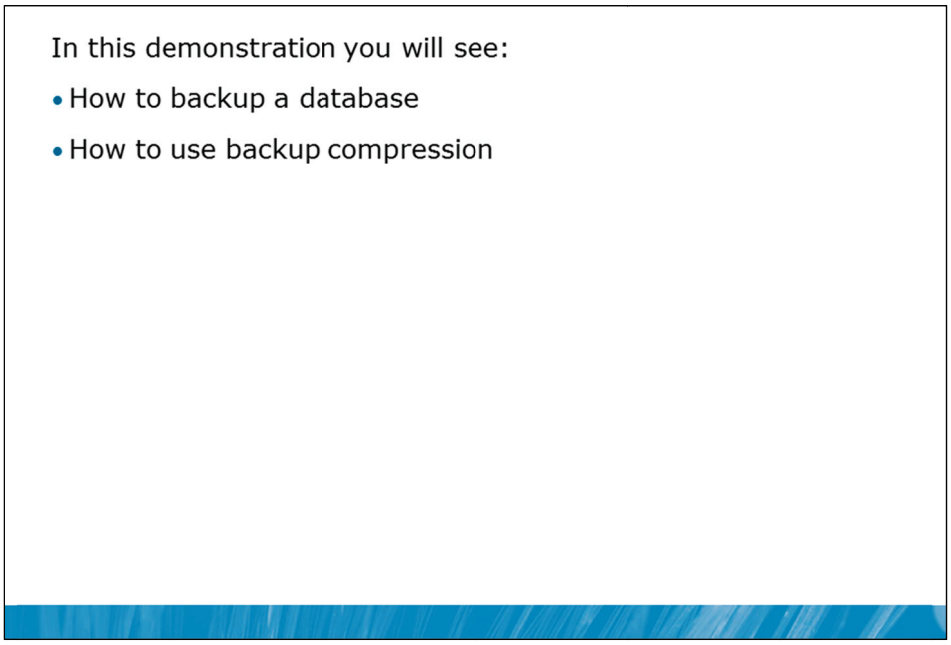
For example, imagine a scenario where a database is created, and at a later time, a full backup of the database is performed. At that point, the database could be recovered. If the recovery model of the database was then changed to simple, and subsequently changed back to full, a break in the log file chain would have occurred. Even though a previous full database backup had occurred, the database could only be recovered up to the point where the last transaction log backup was made (if any) prior to the change to simple recovery model.

After switching from simple to full recovery model, a full database backup needs to be performed to create a starting point for transaction log backups.

Log Truncation

While the default action is to truncate the transaction log as a result of a transaction log backup, this truncation is not performed if the COPY_ONLY option is used. The COPY_ONLY option is discussed in Lesson 3 of this module.

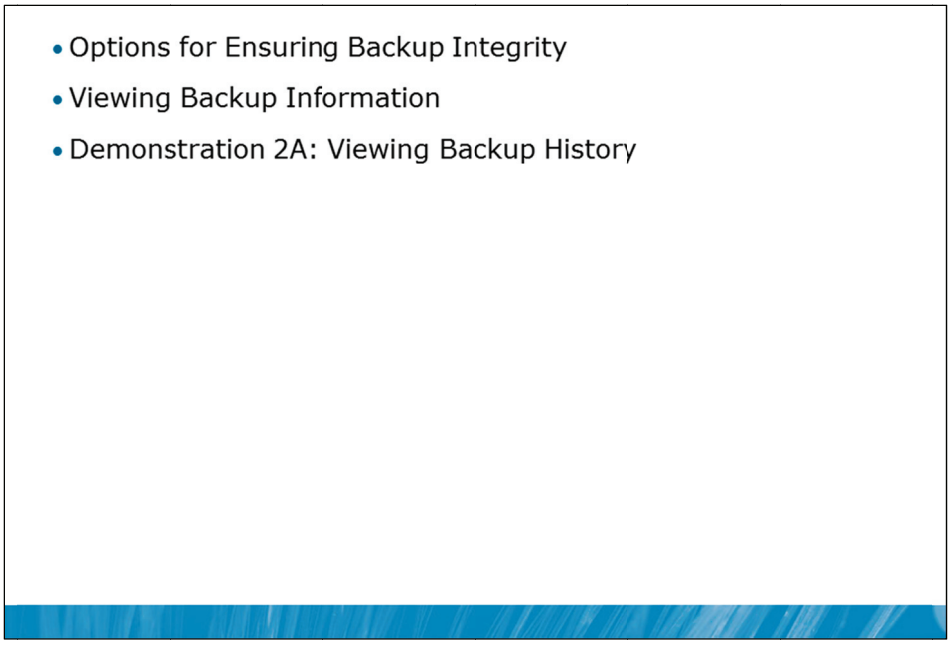
Demonstration 1A: Backing up Databases



Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_06_PRJ\10775A_06_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 2
Managing Database Backups



In Module 5, several example scenarios were presented where users performed backups but at the time the backups needed to be restored, the restore was not possible. The most basic check that should be performed on a backup is to verify that it is readable. The option to verify a backup and other options for ensuring backup integrity are discussed in this lesson.

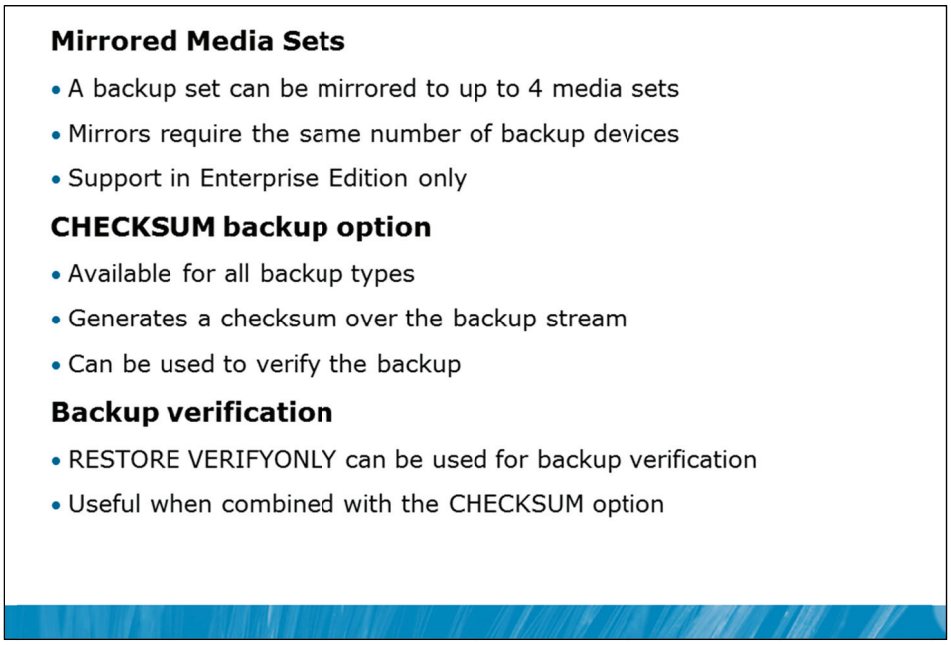
It is also important to know how to find information about backups that have been performed. SQL Server keeps a history of backup operations in the msdb database. In this lesson, you will see how to query the tables that hold this information, and also see how to retrieve the header information from backup devices.

Objectives

After completing this lesson, you will be able to:

- Describe options for ensuring backup integrity.
- View backup information.

Options for Ensuring Backup Integrity



Key Points

A great deal of effort can be expended in performing backups. This effort can be entirely wasted if the backups that are produced are not usable when the time comes to restore them. SQL Server includes several options to help avoid an inability to restore backups.

Mirrored Media Sets

A mirrored media set is a copy of the backup media set optionally created in parallel during the backup operation, on the Enterprise Edition of SQL Server.

A mirrored media set consists of two to four device mirrors; each mirror contains the entire media set. Each mirror must be configured with the same number of backup devices and the backup devices must be of the same device type.

Mirroring a media set increases availability based on the assumption that it is better to have multiple copies of a backup, rather than a single copy. However it is important to realize that mirroring a media set exposes your system to a higher level of hardware failure risk, as a failure of any of the backup devices, causes the entire backup operation to fail.

A mirrored backup set is created by using the MIRROR TO option as shown in the following command:

```
BACKUP DATABASE AdventureWorksDW
TO DISK = 'D:\SQLBackups\AW.bak'
MIRROR TO DISK = 'L:\SQLBackups\AW_M.bak'
WITH FORMAT, INIT;
```

WITH CHECKSUM Option

SQL Server 2005 introduced the option to perform a checksum operation over an entire backup stream. This option consumes slightly more CPU resources than backups without the calculation of a checksum. The WITH CHECKSUM option validates the page-level information (checksum or torn page if either is present) as well as the one checksum for the backup stream. The checksum value is written to the end of the backup and can be checked during restore operations or during backup verification operations made with the RESTORE VERIFYONLY command.

A backup checksum is enabled by using the CHECKSUM option as shown in the following command:

```
BACKUP DATABASE AdventureWorksDW
TO DISK = 'D:\SQLBackups\AW.bak'
WITH CHECKSUM;
```

Note The COMPRESSION option also enables the CHECKSUM option automatically.

Backup Verification

For backup verification, a RESTORE VERIFYONLY command exists that checks the backup for validity. It performs the following tests:

- Backup set is complete.
- All volumes are readable.
- Page identifiers are correct (to the same level as if it were about to write the data).
- Checksum is valid (if present on the media).
- Sufficient space exists on destination devices.

The checksum value can only be validated if the backup was performed with the WITH CHECKSUM option. Without the CHECKSUM option during backup, the verification options only check the metadata and not the actual backup data.

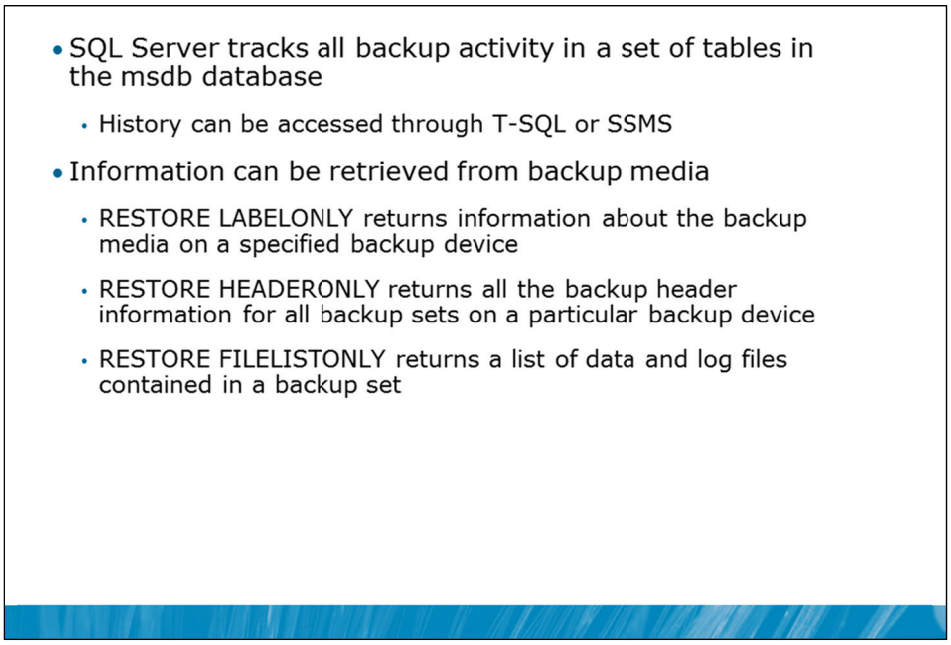
Verification can also be performed through an option in the backup database task in SSMS, and as part of SQL Server Maintenance plans.

Note Consider verifying backups on a different system to the one where the backup was performed, to eliminate the situation where a backup is only readable on the source hardware.

Note Make sure that you create your backups on different disks than the ones holding your database files. Avoid ever overwriting your most recent backup.

Question: Can you guarantee that a database could be recovered, if a backup of the database can be verified?

Viewing Backup Information



Key Points

SQL Server tracks all backup activity in a set of tables in the msdb database:

- backupfile
- backupfilegroup
- backupmediafamily
- backupmediaset
- backupset

These tables can be queried to retrieve information about backups that have been performed. In Demonstration 2A, you will see how to perform these queries.

SSMS also has options to retrieve details of backup operations on databases and logs. In Demonstration 2A, you will also see an example report that is launched from within SSMS and that shows relevant backup information.

Deleting Backup History

Backup history can be deleted using system stored procedures. Consider the following command:

```
EXEC sp_delete_backuphistory @oldest_date = '20090101';
```

This command deletes all history prior to the date provided. The date provided is the oldest date to keep.

Also consider the following command:

```
EXEC sp_delete_database_backuphistory @database_name = 'Market';
```

This command deletes the history for a database named Market.

If a database is restored onto another server, the backup information is not restored with the database, as it is held in the msdb database of the original system.

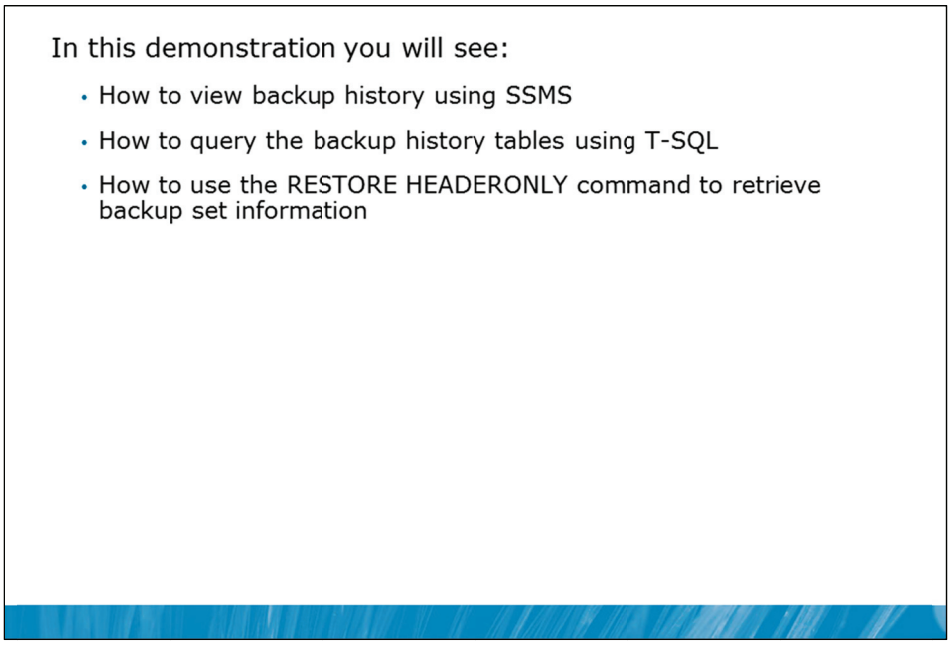
Retrieving Backup Metadata

Information about a specific media set is available by executing the RESTORE command with the following options:

Command	Description
RESTORE LABELONLY	Returns information about the backup media on a specified backup device
RESTORE HEADERONLY	Returns all the backup header information for all backup sets on a particular backup device
RESTORE FILELISTONLY	Returns a list of data and log files contained in a backup set

Question: In what situations would SQL Server not have complete information on the backups of a database stored in msdb?

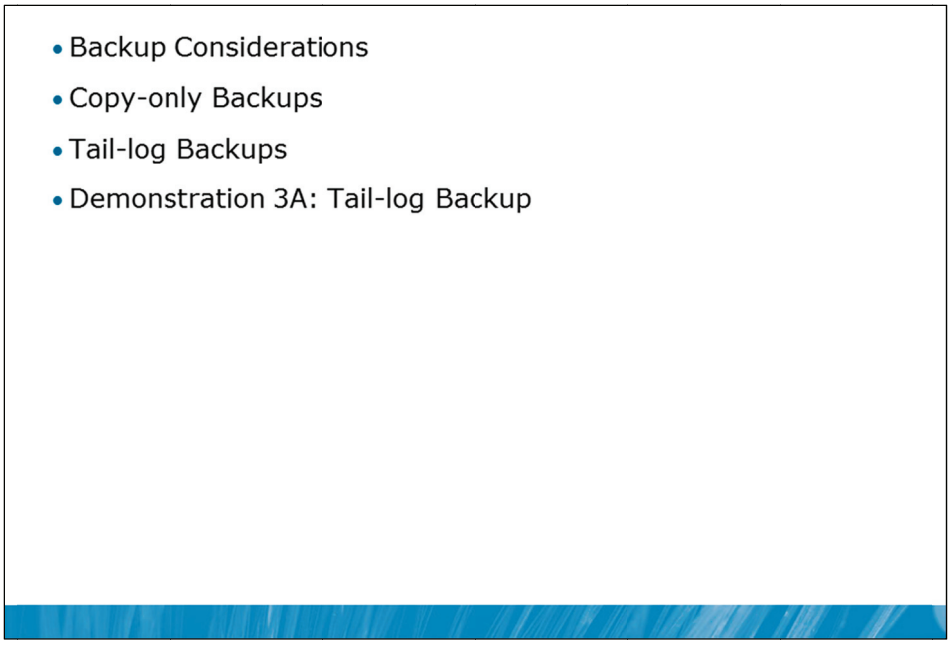
Demonstration 2A: Viewing Backup History



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_06_PRJ\10775A_06_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **21 – Demonstration 2A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lesson 3
Working with Backup Options



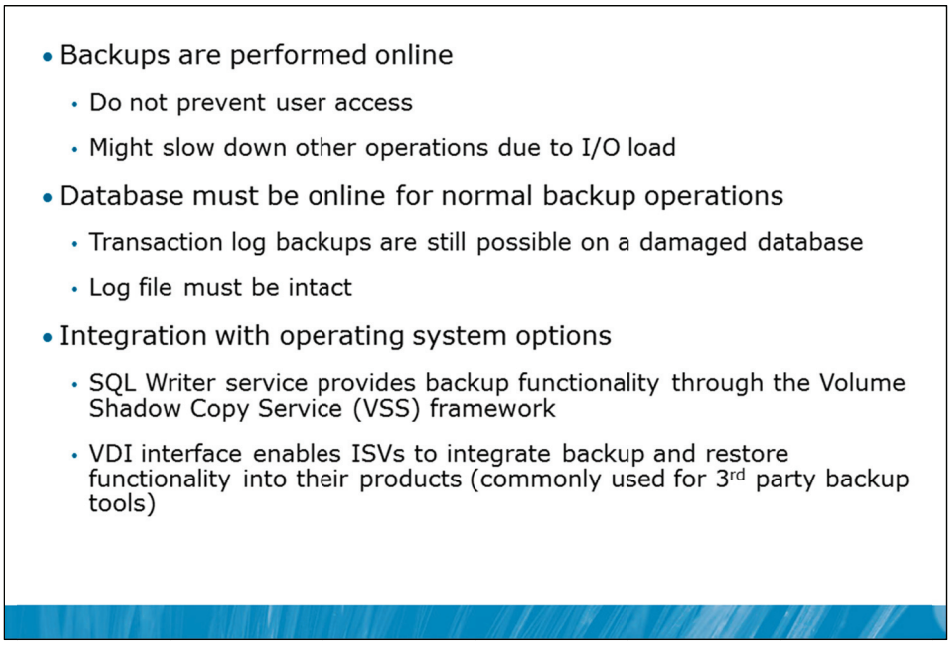
Now that you have seen the most common options related to backing up databases and transactions logs, it is important to consider general considerations surrounding the creation of backups, along with some of the less commonly used, but useful, options related to backups.

Objectives

After completing this lesson, you will be able to:

- Highlight backup considerations.
- Perform copy-only backups.
- Perform tail-log backups.

Backup Considerations



Key Points

SQL Server backups can be created while other users are working with the system. Other users might, however, be impacted by the I/O load placed on the system by the backup operation. SQL Server also places some limitations on the types of commands that can be executed while a backup is being performed. For example, the ALTER DATABASE command cannot be used with ADD FILE or REMOVE FILE options and shrinking a database is not permitted during a backup.

The BACKUP command cannot be included in either an explicit or an implicit transaction. You cannot ROLLBACK a BACKUP.

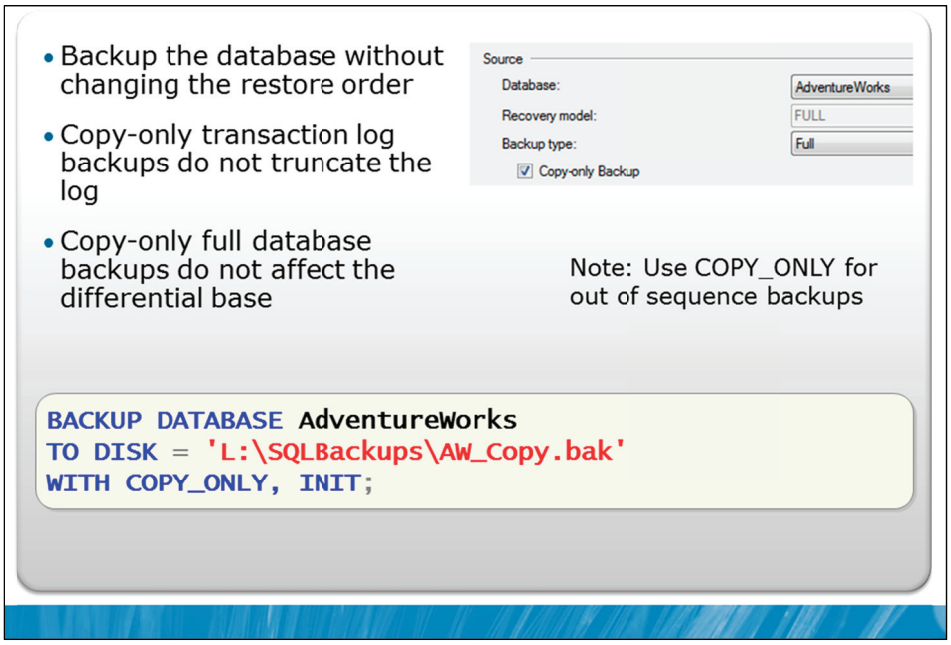
Databases can only be backed up when they are online but it is still possible to perform a backup of the transaction log when a database is damaged, assuming the log file itself is still intact. This is a key reason that it is important to separate data and log files onto separate physical media.

VSS and VDI

The Windows Volume Shadow Copy Service (VSS) and the Virtual Device Interface (VDI) programming interface are available for use with SQL Server. The main use for these interfaces is so that third party backup tools can work with SQL Server.

In very large systems, it is common to need to perform disk to disk imaging while the system is in operation, as standard SQL Server backups might take too long to be effective. The VDI programming interface allows an application to freeze SQL Server operations momentarily while a consistent snapshot of the database files is created. This form of snapshot is commonly used in geographically distributed SAN replication systems.

Copy-only Backups



Key Points

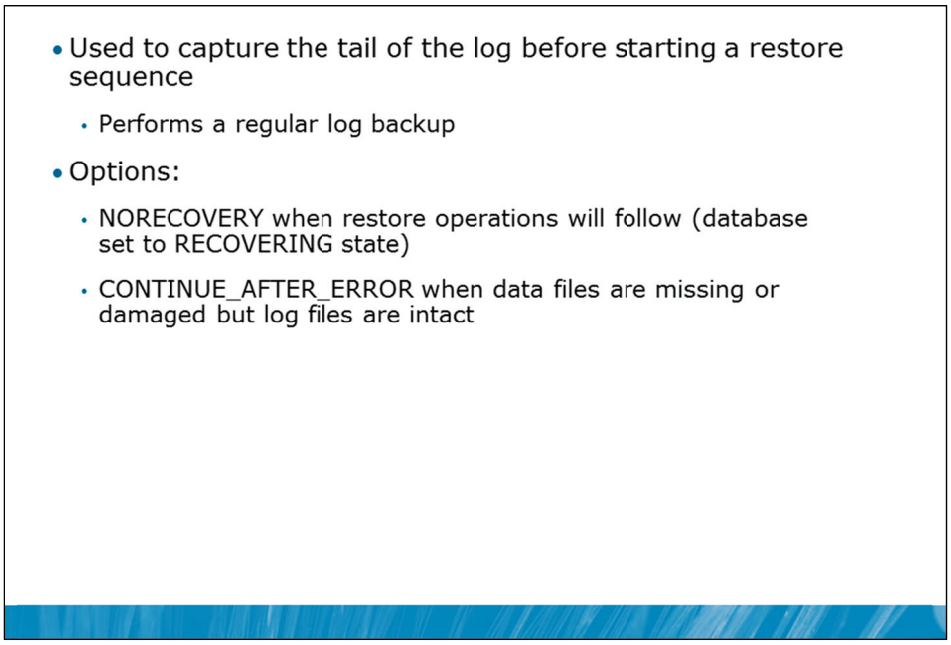
A copy-only backup is a SQL Server backup that is independent of the sequence of conventional SQL Server backups. Usually, taking a backup changes the database and affects how later backups are restored.

There may, however, be a need to take a backup for a special purpose without affecting the overall backup and restore procedures for the database.

Copy-only backups can be made of either the database or of the transaction logs. Restoring a copy-only full backup is the same as restoring any full backup.

Question: Can you suggest a scenario where you might use a Copy-only backup?

Tail-log Backups



Key Points

SQL Server 2005 and later versions require that you take a tail-log backup before you start a restore over an existing database. This requirement is intended to avoid a common data loss scenario where a database has not been fully backed up before a restore of the database occurs. The requirement ensures that, by default, all transactions must have been written to at least one backup, before they can be overwritten. The tail-log backup prevents work loss and keeps the log chain intact.

WITH NORECOVERY

The WITH NORECOVERY option is normally applied to restore operations and is discussed in detail in Module 7. However, users are often surprised to see that the command to create a tail-log backup also has a WITH NORECOVERY option. This option backs up the transaction log and then immediately changes the database into a recovering state. The most common reason for the use of this option is in conjunction with log shipping, when a server is changing roles from a primary server to a secondary server.

Note Log Shipping is an advanced option that is out of scope for this course.

Tail-log Backups

When you are recovering a database to the point of a failure, the tail-log backup is often the last backup of interest in the recovery plan. It is a standard type of transaction log backup. If you cannot back up the tail of the log, you can only recover a database to the end of the last backup that was created before the failure.

Not all restore scenarios require a tail-log backup. You do not need to have a tail-log backup if the recovery point is contained in an earlier log backup, or if you are moving or replacing (overwriting) the database and do not need to restore it to a point of time after the most recent backup.

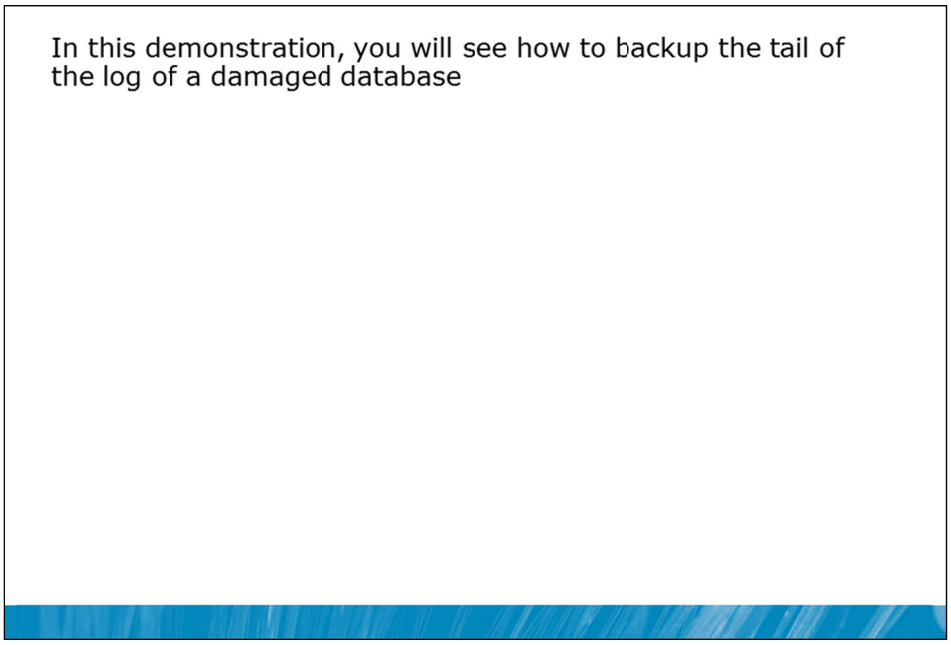
Use the CONTINUE_AFTER_ERROR option if you are backing up the tail of a damaged database.

If you are unable to back up the tail of the log using the NO_TRUNCATE option when the database is damaged, you can attempt a tail-log log backup by specifying CONTINUE_AFTER_ERROR instead of NO_TRUNCATE.

The NO_TRUNCATE option is available and is equivalent to the use of the COPY_ONLY and CONTINUE_AFTER_ERROR options together. It should be attempted for damaged databases. It causes the database engine to attempt the backup regardless of the state of the database. This means that a backup taken while using the NO_TRUNCATE option might have incomplete metadata. Without the NO_TRUNCATE database option, the database must be online.

Question: What is the biggest advantage of being able to perform tail-log backups even when data files are damaged?

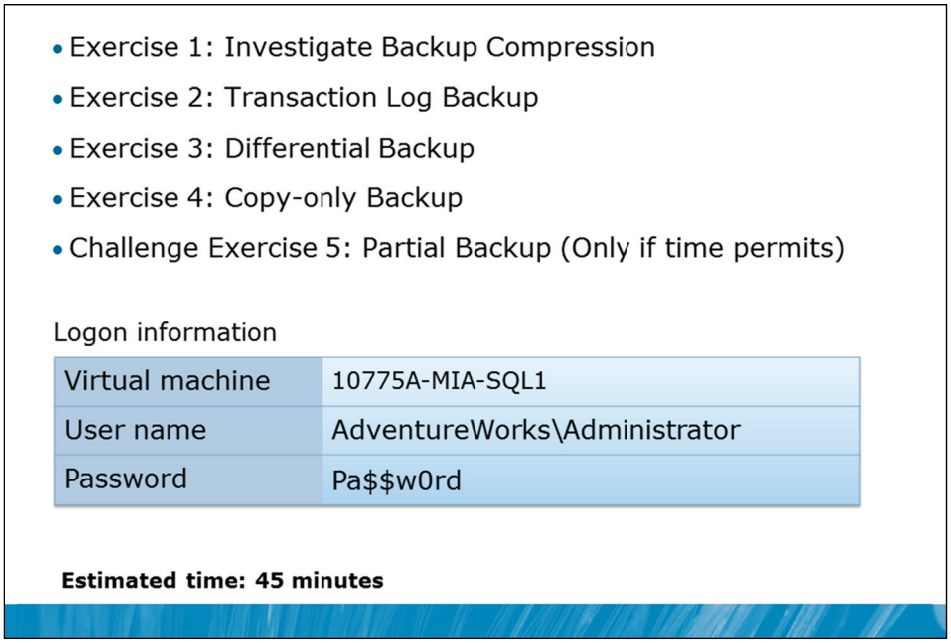
Demonstration 3A: Tail-log Backup



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_06_PRJ\10775A_06_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **31 – Demonstration 3A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lab 6: Backup of SQL Server Databases



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_06_PRJ\10775A_06_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

You have reviewed and updated the recovery models. As the database administrator, you need to implement a database backup strategy. You have been provided with details of the required backup strategy for a number of databases on a SQL Server instance. You need to complete the required backups.

Exercise 1: Investigate Backup Compression

Scenario

The size of the database backups has been increasing with the number of orders being placed. You want to investigate the amount of space that can be saved by using backup compression. In this exercise, you need to investigate the effectiveness of backup compression when backing up the MarketDev database on the Proseware instance. You will perform a full database backup with backup compression disabled. You will then perform another full database backup with backup compression enabled. You will compare the backup files produced to identify the possible space savings.

The main tasks for this exercise are as follows:

1. Create a database backup without compression.
2. Create a database backup with compression.
3. Compare the file sizes created.

▶ Task 1: Create a database backup without compression

- Using Windows Explorer, create a new folder L:\SQLBackups.
- Perform a full backup of the MarketDev database with compression disabled to the file L:\SQLBackups\MarketDev_Full_Uncompressed.BAK.

▶ Task 2: Create a database backup with compression

- Perform a full backup of the MarketDev database with compression enabled to the file L:\SQLBackups\MarketDev_Full_Compressed.BAK.

▶ Task 3: Compare the file sizes created

- Calculate the space savings provided by compression as:

```
SpaceSavings=(Uncompressed size-Compressed size)*100/Uncompressed size
```

Results: After this exercise, you have calculated the space saved by using backup compression on the MarketDev database.

Exercise 2: Transaction Log Backup

Scenario

Part of the ongoing management of the MarketDev database is a series of transaction log backups to provide point in time recovery. In this exercise, you need to back up the transaction log.

The main tasks for this exercise are as follows:

1. Execute a script to introduce workload to the MarketDev database.
2. Backup the transaction log on the MarketDev database.

▶ Task 1: Execute a script to introduce workload to the MarketDev database

- Open and execute the script file 61 – Workload File.sql from Solution Explorer.

▶ Task 2: Backup the transaction log on the MarketDev database

- Backup the transaction log of the MarketDev database to the file L:\SQLBackups\MarketDev_Log_Compressed.BAK. Use backup compression during the backup.

Results: After this exercise, you should have completed a transaction log backup.

Exercise 3: Differential Backup

Scenario

There is a concern that the data volumes in the MarketDev database will be so large that daily full backups will not be possible. In this exercise, you need to perform a differential backup to assist to manage the size of the database backups.

The main tasks for this exercise are as follows:

1. Execute a script to introduce workload to the MarketDev database.
2. Create a differential backup of the MarketDev database.

▶ Task 1: Execute a script to introduce workload to the MarketDev database

- Open and execute the script file 61 – Workload File.sql from Solution Explorer.

▶ Task 2: Create a differential backup of the MarketDev database

- Create a differential backup of the MarketDev database to the file L:\SQLBackups\MarketDev_Differential_Compressed.BAK. Use backup compression during the backup.
- Using Windows Explorer, note the size of the Differential backup compared to the Full backup.

▶ Task 3: Execute a script to introduce workload to the MarketDev database

- Open and execute the script file 61 – Workload File.sql from Solution Explorer.

▶ Task 4: Append a differential backup to the previous differential backup file

- Append a differential backup of the MarketDev database to the file L:\SQLBackups\MarketDev_Differential_Compressed.BAK.
- Using Windows Explorer, note that the size of the Differential backup has increased. The file now contains two backups.

Results: After this exercise, you should have completed two differential backups.

Exercise 4: Copy-only Backup

Scenario

Another team periodically needs a temporary copy of the MarketDev database. It is important that these copies do not interfere with the backup strategy that is being used. In this exercise, you need to perform a copy-only backup and verify the backup.

The main task for this exercise is as follows:

1. Create a copy-only backup of the MarketDev database, ensuring to choose to verify the backup.

▶ Task 1: Create a copy-only backup of the MarketDev database, ensuring to choose to verify the backup

- Create a copy-only backup of the MarketDev database to the file L:\SQLBackups\MarketDev_Copy_Compressed.BAK. Make sure you choose to:
  - Verify the backup while creating it.
  - Use backup compression.
  - Choose to create a new media set called MarketDev Copy Backup.
  - For the media set description use MarketDev Copy Backup for Integration Team.

Results: After this exercise, you should have completed a copy-only backup.

Challenge Exercise 5: Partial Backup (Only if time permits)

Scenario

On the Proseware instance, there is a database called RateTracking that has two filegroups. The ARCHIVE filegroup is set to read-only and both the default filegroup USERDATA and the PRIMARY filegroup are read-write. In this exercise, you need to back up the read-write filegroups only, using T-SQL commands.

The main task for this exercise is as follows:

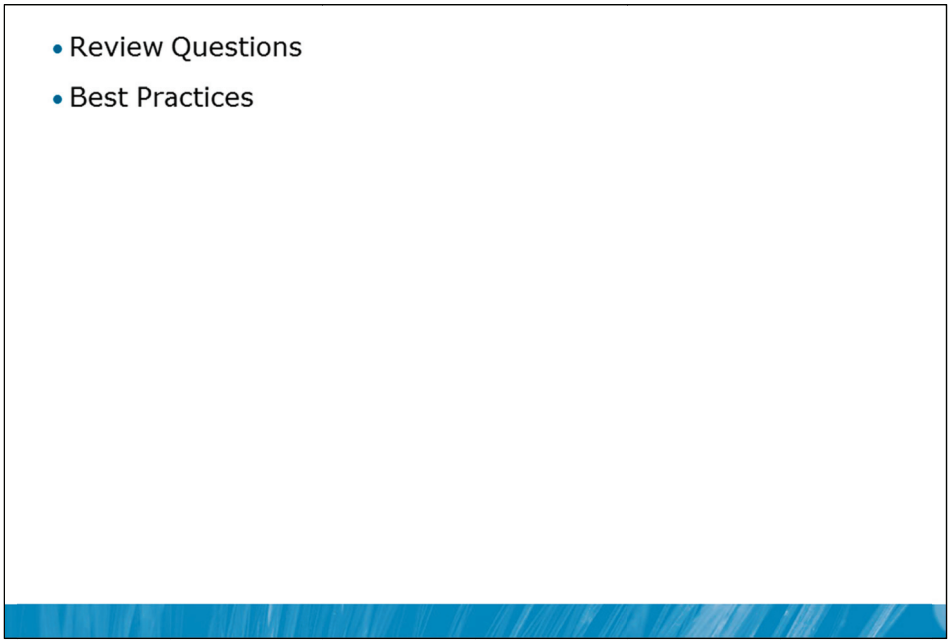
1. Perform a backup of the read-write filegroups on the RateTracking database.

▶ Task 1: Perform a backup of the read-write filegroups on the RateTracking database

- Perform a backup of the read-write filegroups (USERDATA and PRIMARY) on the RateTracking database. Write the backup to the file L:\SQLBackups\RateTracking_ReadWrite.BAK. Use the CHECKSUM and INIT options.

Results: After this exercise, you should have completed a partial backup.

Module Review and Takeaways



Review Questions

1. Which backup types can be performed in simple recovery model?
2. How can backup information be read?

Best Practices

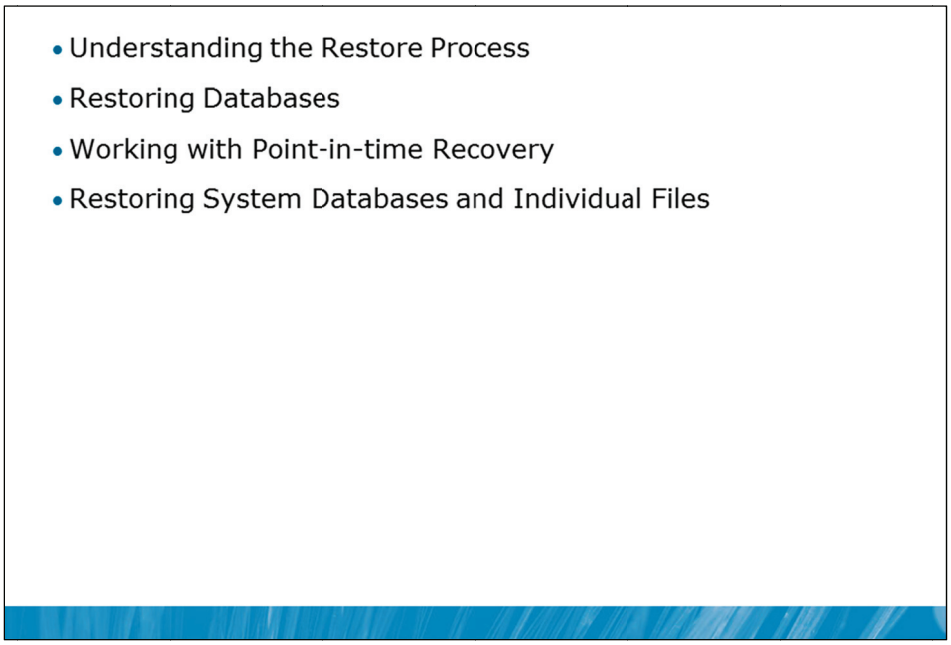
1. Consider using CHECKSUM to create a checksum over your backup files.
2. Use backup compression to increase backup and restore performance and safe storage space.
3. Consider mirroring your backups to increase safety.
4. Check if differential backup can speed up your restore process in full recovery mode.
5. Use COPY_ONLY for out of sequence backups.

Module 7

Restoring SQL Server 2012 Databases

Contents:

Module Overview



In the previous module, you saw how to create backups of Microsoft® SQL Server® 2012 databases. A backup strategy might involve many different types of backup. This means that it is important for you to understand the process required when restoring databases. Often when database restores are required, an urgent situation exists. Unfortunately, database administrators often make errors of judgment when placed into urgent recovery situations. The first rule when dealing with a bad situation should always be to "do no further harm". In urgent situations, it is more important than ever to have a clear plan for how to proceed. A good understanding of the process required and a good plan can help avoid making the situation worse.

Not all database restores are related to system failures. With most system failure situations, there is a need to return the system to as close as possible to the state that it was in prior to the failure. Some failures are related to human errors. In those cases, you may wish to recover the system to a point prior to the failure. The point-in-time recovery features of SQL Server 2012 can help you to achieve this.

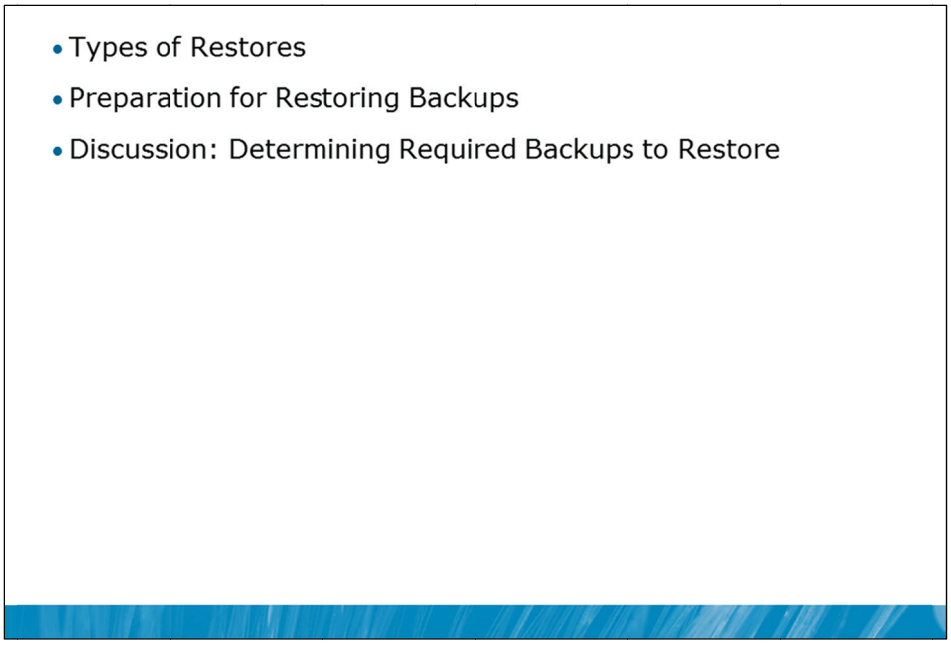
User databases are more likely to be affected by system failures than system databases as user databases are typically much larger than system databases. However, system databases can be affected by failures, and special care needs to be taken when recovering system databases. In particular, you need to understand how each system database should be recovered as not all system databases can be recovered using the same process.

Objectives

After completing this lesson, you will be able to:

- Understand the restore process.
- Restore databases.
- Work with Point-in-time Recovery.
- Restore system databases and individual files.

Lesson 1
Understanding the Restore Process



You have seen that there are many types of backup that can be created with SQL Server 2012. Similarly, there are different types of restore processes that can be required.

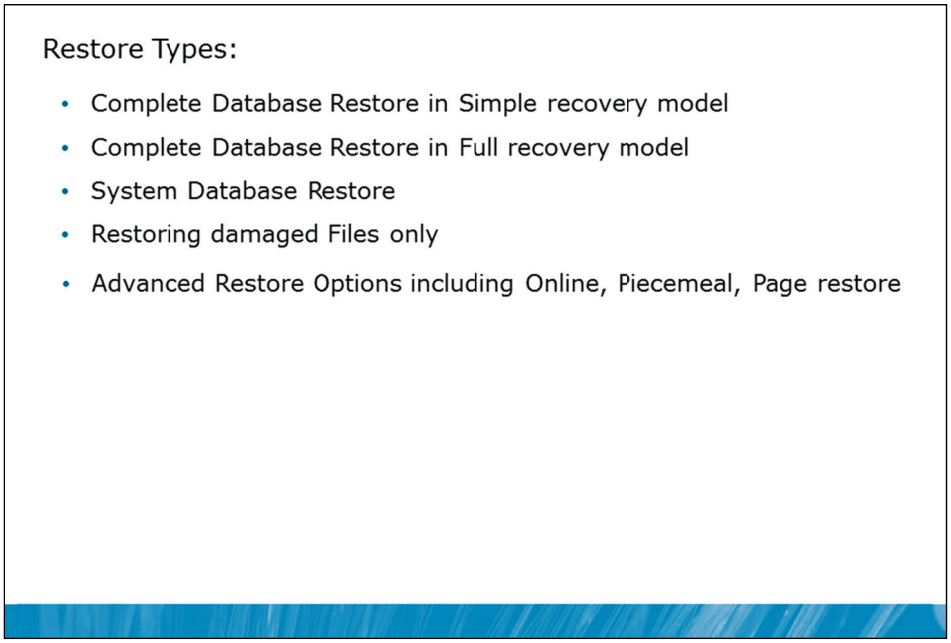
It was mentioned earlier that when it is time to recover a database, a good plan is required, to avoid causing further damage. Once the preliminary step of attempting to create a tail-log backup has been carried out, the most important decision that needs to be taken is the determination of which database backups need to be restored, and in which order.

Objectives

After completing this lesson, you will be able to:

- Describe the different types of restores.
- Decide which backups to restore and in which order.

Types of Restores



Key Points

Restoring a database in SQL Server 2012 is a two-step process. The first step involves restoring data pages from one or more backups. Once the data pages have been restored, the database is potentially in an inconsistent state. To correct this situation, in the second step of the restore process, the available details from the transaction log are used to recover the database. The restore scenarios available for a database depend on the recovery model of the database and the edition of SQL Server.

Complete Database Restore in Simple Recovery model

The most basic restore strategy for SQL Server databases is to restore and recover a full database backup. If a differential backup of the database is available, the latest differential backup could be restored after the restore of the full database backup but before the recovery process for the database.

In most cases that use simple recovery model, no differential backups are performed, in which case only the last full database backup will be restored and the database would be returned to the state it was in at the time just prior to the full database backup being completed.

Complete Database Restore in Full Recovery model

The most common restore strategy requires full or bulk-logged recovery model and involves restoring full, differential (if present), and log backups.

To restore to the last available point in time:

1. Try to perform a tail-log backup. This step might or might not be possible, depending upon the type of failure that has occurred.
2. Restore the last full database backup.

3. Restore the most recent differential backup if a differential backup had been created.
4. Restore all transaction log backups performed after the point of the most recent differential backup (or full backup if no differential backup was created) in the same sequence as they were created. If a tail-log backup was successfully created, it would be restored as the most recent transaction log backup.

While the aim of the restore would normally be to recover the database to the latest point in time possible, options do exist to restore the database to earlier points in time. These options will be discussed in Lesson 3 of this module.

Note Even if differential backups and transaction log backups were created, you can choose not to apply them, if you wish to return the database state to an earlier point in time.

System Database Restore

Restoring system databases is possible but requires special processes to avoid further issues from occurring. For example, if a master database is left in an inconsistent state, SQL Server will refuse to start until the master database is recovered. The recovery of system databases will be discussed in Lesson 4 of this module.

File Restore

If individual files in a database have become corrupted or have been lost, the ability to restore individual files has the potential to substantially reduce the overall time to recover the database. The recovery of individual files is only supported for read-only files when operating in simple recovery model, but can be used for read-write files when using the bulk-logged or full recovery models. The recovery of individual files uses a process that is similar to the complete database restore process and will be discussed in Lesson 4 of this module.

Online Restore

Online restore involves restoring data while the database is online. This is the default option for File, Page, and Piecemeal restores. In SQL Server 2012, online restore is only available in the Enterprise edition.

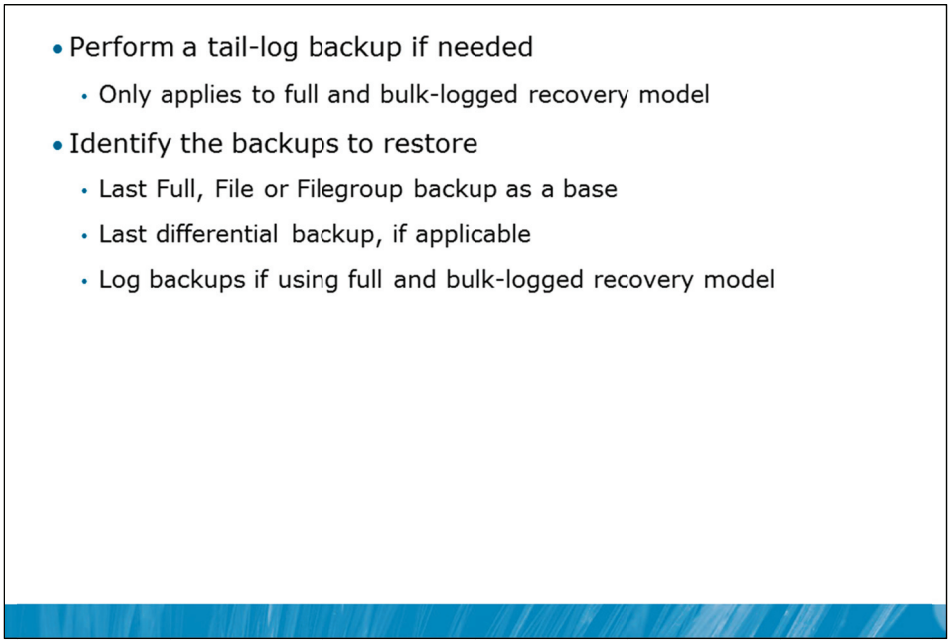
Piecemeal Restore

A piecemeal restore is used to restore and recover the database in stages, based on filegroups, rather than restoring the entire database at a single time. The first filegroup that must be restored is the primary filegroup. In SQL Server 2012, piecemeal restore is only available in the Enterprise edition.

Page Restore

Another advanced option is the ability to restore an individual data page. If an individual data page is corrupt, users will usually see either an 823 error or an 824 error when they execute a query that tries to access the page. An online page restore could be used to try to recover the page. Once the restore has commenced, if a user query tries to access the page, the error that the user would see is error 829, which indicates "page is restoring". If the page restore is successful, user queries that access the page would again return results as expected. Page restores are supported under full and bulk-logged recovery models but are not supported under simple recovery model. In SQL Server 2012, online page restore is only available in the Enterprise edition. Offline page restore is available in lower editions.

Preparations for Restoring Backups



Key Points

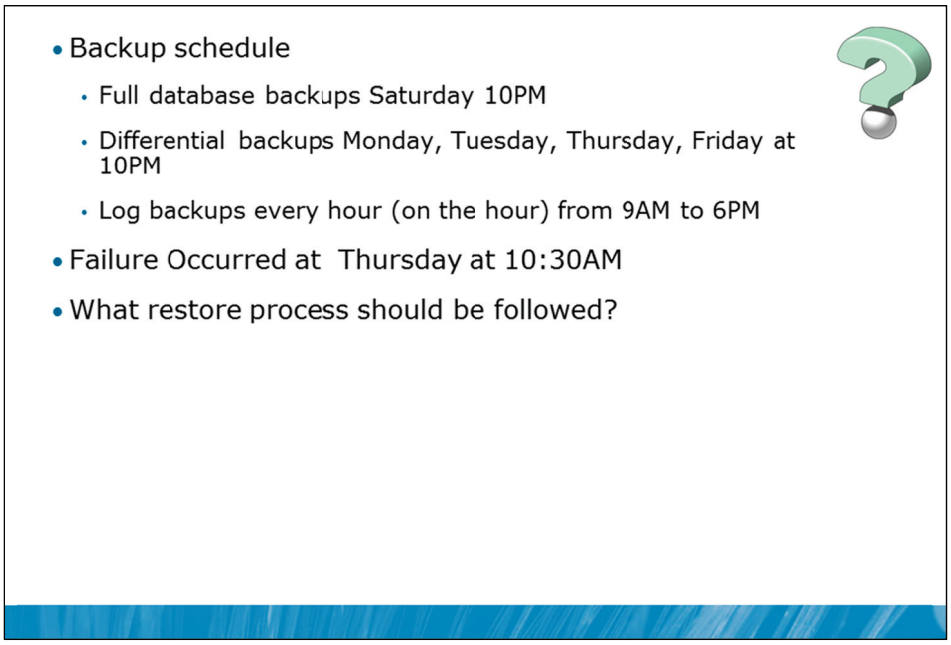
In critical situations, users often make inappropriate choices about the actions that should be taken. It is important to avoid any action that will make the situation worse than necessary. Before restoring any database, it is important to attempt a tail-log backup, unless you are intending to replace the current state of the database. The tail-log backup can often be performed, even when damage has occurred to the data files of the database. The tail-log backup is critical when you need to restore the database to the latest point in time possible.

Identifying Backups to Restore

The recovery of any database depends upon restoring the correct backups in the correct order. The normal process for restoring a database is:

1. Restore the latest full database backup as a base to work from. (If only individual files are damaged or missing, you may be able to restore only those files).
2. If differential backups have been created, only the latest differential backup is needed. (Differential backups save all database extents that have been modified since the last full database backup. Differential backups are not incremental in nature.
3. If transaction log backups have been created, all transaction log backups since the last differential backup are required. You also need to include the tail-log backup created at the start of the restore process, if the tail-log backup was successful. (This step does not apply to databases in simple recovery model).

Discussion: Determining Required Backups to Restore

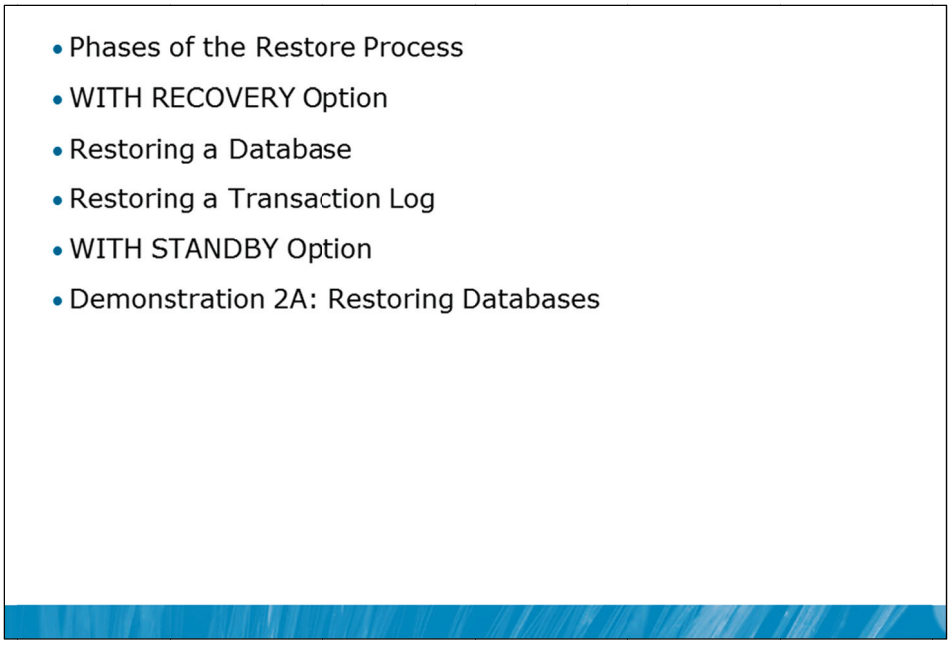


Discussion

The example on the slide describes the backup schedule for an organization.

Question: If a failure occurs at Thursday at 10:30AM, what is the restore process that should be undertaken?

Lesson 2
Restoring Databases



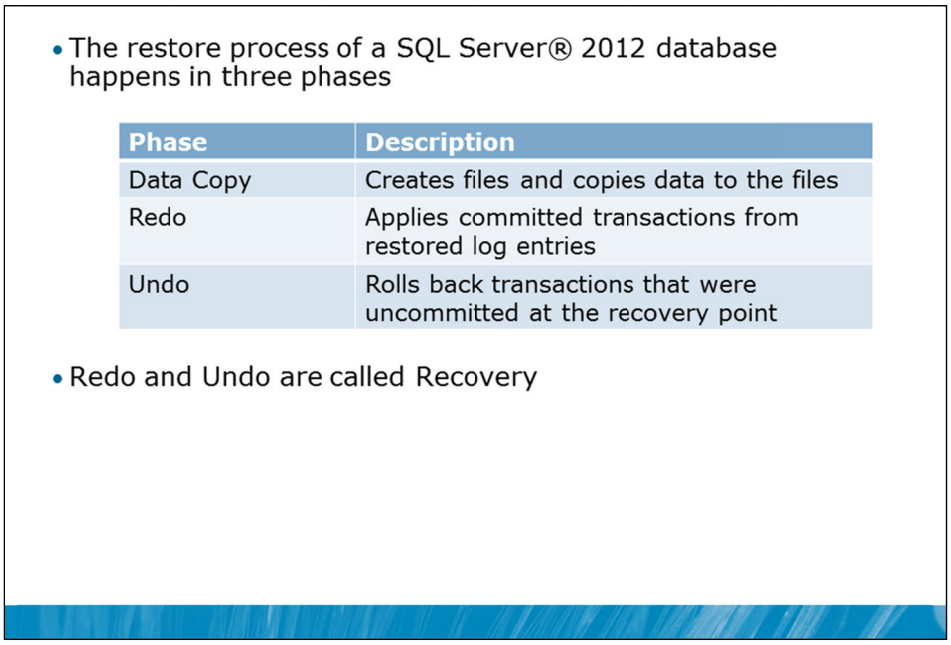
It is important to understand the phases that SQL Server 2012 uses when restoring a database. Once a decision has been made to restore a database, you need to know how to implement the restore process. The restore process might involve both database and transaction log backups. When multiple backups need to be restored in a single process, you need to control the point at which the recovery of the database occurs. If database recovery occurs too early in the process, you will not be able to complete the entire restore process.

Objectives

After completing this lesson, you will be able to:

- Describe the different phases of the restore process.
- Restore a database from a full database backup or a differential backup.
- Restore a transaction log backup.
- Control database recovery by using the WITH RECOVERY option.
- Allow read-only access to a recovering database by using the WITH STANDBY option.

Phases of the Restore Process



Key Points

The restore of a SQL Server 2012 database passes through three phases: Data Copy, Redo, and Undo. The combination of the Redo and the Undo phases is commonly referred to as the recovery of a database.

Data Copy

The data copy phase is typically the longest phase in a database restore. The data files from the database need to be recovered from the backups. Before any data pages are restored, the header of the backup is read and SQL Server recreates the required data and log files. If instant file initialization (IFI) has not been enabled by granting rights to the SQL Server service account, the rewriting of the data files can take a substantial amount of time.

Once the data and log files are recreated, the data files are restored from the full database backup. Data pages are retrieved from the backup in order and written to the data files.

The log files need to be zeroed out before they can be used. IFI is not used for log files. This process can also take a substantial time if the log files are large.

If a differential backup is also being restored, SQL Server overwrites the extents in the data files with the ones that are contained in the differential backup.

Redo Phase

Details from the transaction log are then retrieved. In simple recovery model, these details would only be retrieved from either the full database backup or the differential backup, if a differential backup is also being restored. In full or bulk-logged recovery model, these log file details will be supplemented by the contents of any transaction log backups that were taken after the full and differential database backups.

In the redo phase, SQL Server rolls into the database pages all changes that are contained within the transaction log details, up to the recovery point. The recovery point is typically the latest time for which transactions are contained in the log.

Undo Phase

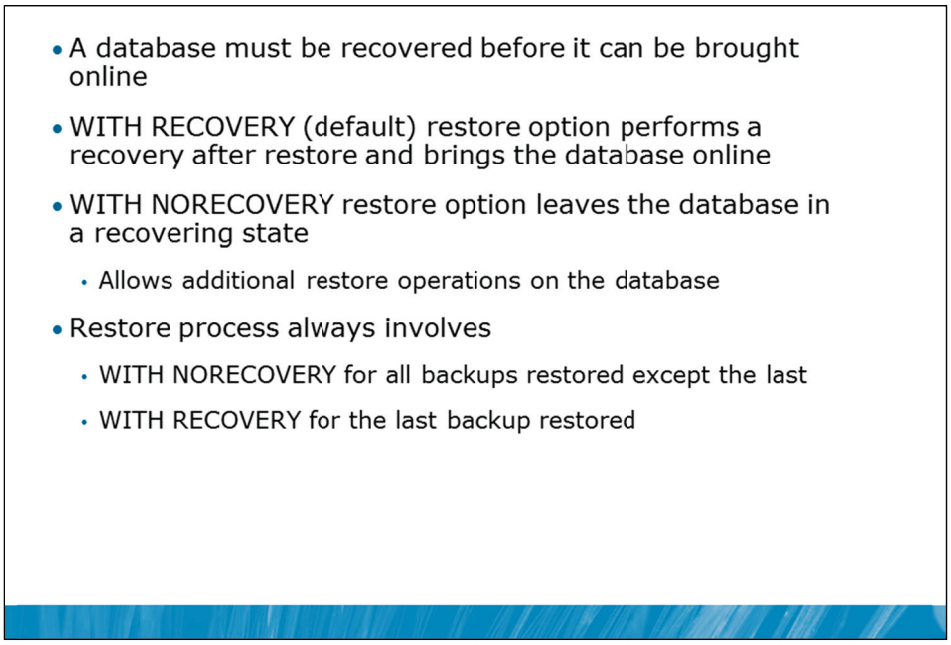
Note that the transaction log details will likely include details of transactions that were not committed at the recovery point, which is typically the time of the failure. In the undo phase, SQL Server rolls back any uncommitted transactions.

Because the action of the undo phase involves rolling back uncommitted transactions and placing the database online, subsequent backups cannot be restored.

During the undo phase, the Enterprise edition of SQL Server 2012 will allow the database to come online and will allow users to begin to access the database. This capability is referred to as the fast recovery feature. Queries that attempt to access data that is still being undone are blocked until the undo phase is complete. This could potentially cause transactions to time out.

Question: Why does SQL Server need to redo and undo transactions when only restoring a full database backup?

WITH RECOVERY Option



Key Points

In general, a database cannot be brought online until it has been recovered. The one exception to this is the fast recovery option that was mentioned in the last topic. The fast recovery option allows users to access the database while the undo phase is continuing.

Recovery Events

Note that recovery does not only occur during the execution of RESTORE commands. If a database is taken offline and then placed back into an ONLINE state, recovery of the database will occur. The same recovery process occurs when SQL Server 2012 restarts.

Note Other events that lead to database recovery include clustering or database mirroring failovers. Failover clustering and database mirroring are advanced topics that are out of scope for this course. These events are listed here for completeness.

The recovery process in SQL Server is critical to the maintenance of transactional integrity that requires that all transactions that had committed are recorded in the database and that all transactions that had not committed are rolled back.

Recovery Options

Each RESTORE command includes an option to specify WITH RECOVERY or WITH NORECOVERY. The WITH RECOVERY option is the default action and does not need to be specified.

It is important to choose the correct option (WITH RECOVERY or WITH NORECOVERY) when executing a RESTORE command. The process is straightforward in most cases. All restores must be performed WITH NORECOVERY except the last restore which has to be performed WITH RECOVERY.

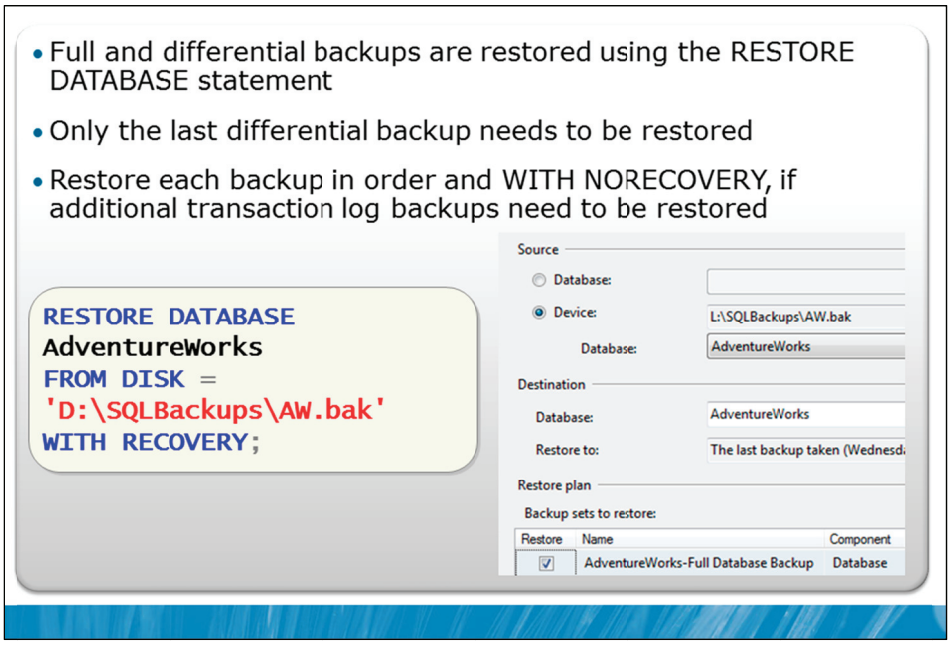
There is no way to restore additional backups after a restore WITH RECOVERY. If a backup has been performed WITH RECOVERY by accident the restore sequence has to be restarted.

If the last backup of a set was inadvertently also restored WITH NORECOVERY, the database can be forced to recover by executing the following command:

```
RESTORE LOG databasename WITH RECOVERY;
```

Question: Why is it not possible to restore additional backups to a recovered database?

Restoring a Database



Key Points

You can restore a database using either the GUI in SSMS or by using the RESTORE DATABASE command in T-SQL.

If a single database backup is being restored, the WITH RECOVERY option can be used, as no later backups need to be restored. You can also omit the WITH RECOVERY option as it is the default for the RESTORE DATABASE command.

Differential Restore

The command for restoring a differential backup is identical to the command for restoring a full database backup. Differential backups might be appended to the same file as the full database backup. In that case, you need to specify the file from the media set that you need to restore.

Consider the following command:

```
RESTORE DATABASE AdventureWorks
   FROM DISK = 'D:\SQLBackups\AW.bak'
   WITH FILE = 1, NORECOVERY;

RESTORE DATABASE AdventureWorks
   FROM DISK = 'D:\SQLBackups\AW.bak'
   WITH FILE = 3,  RECOVERY;
```

In this example, the database AdventureWorks is restored from the first file in the media set. The media set is stored in the operating system file D:\SQLBackups\AW.bak. In this case, the second file that was contained in the media set was the first differential backup that was performed on the database. The third file in the media set was the second differential backup that was performed on the database. Because the second differential backup was the latest differential backup that was performed, the second RESTORE command in the example shows how to restore the latest differential backup from that media set.

WITH REPLACE

SQL Server 2012 will not allow you to restore a database backup over an existing database if you have not performed a tail-log backup on the database. If you attempt to do this using SQL Server Management Studio, SQL Server 2012 will provide a warning and will automatically attempt to create a tail-log backup for you first. If you need to perform the restore operation and you do not have a tail-log backup, you must specify the WITH REPLACE option.

Note The WITH REPLACE option needs to be used with caution as it can lead to data loss.

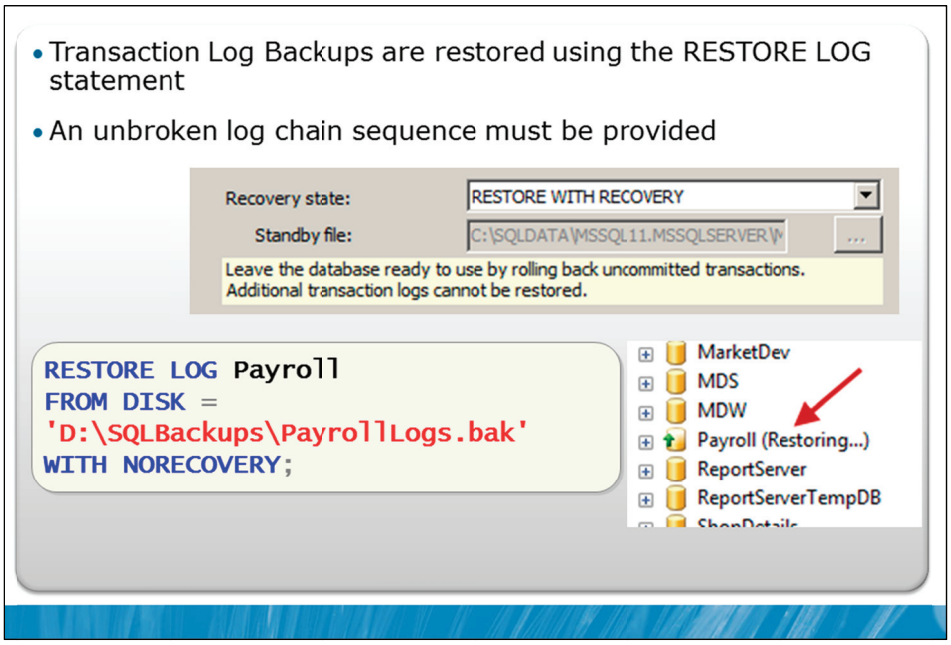
WITH MOVE

When you restore a database from another server, you might need to place the database files in different locations than those that are recorded in the backup from the original server. You might also need to do this if you are copying a database by a process of backup and restore. The WITH MOVE option allows you to specify new file locations. Consider the following command:

```
RESTORE DATABASE Spatial
FROM DISK = 'D:\SQLBackups\Spatial.bak'
WITH MOVE 'Spatial_Data' TO 'D:\MKTG\Spatial.mdf',
     MOVE 'Spatial_Log' TO 'L:\MKTG\Spatial.ldf';
```

In the example shown, the database named Spatial is being restored from another server. As well as specifying the source location for the media set, in the command, new locations for each database file have been specified. Note that the MOVE option requires the specification of the logical file name, rather than the original physical file path.

Restoring a Transaction Log



Key Points

You can restore the transaction logs for a database using either the GUI in SSMS or by using the RESTORE LOG command in T-SQL. All log files apart from the last log should be restored WITH NORECOVERY. The last log file (which is often the tail-log backup) is then restored WITH RECOVERY.

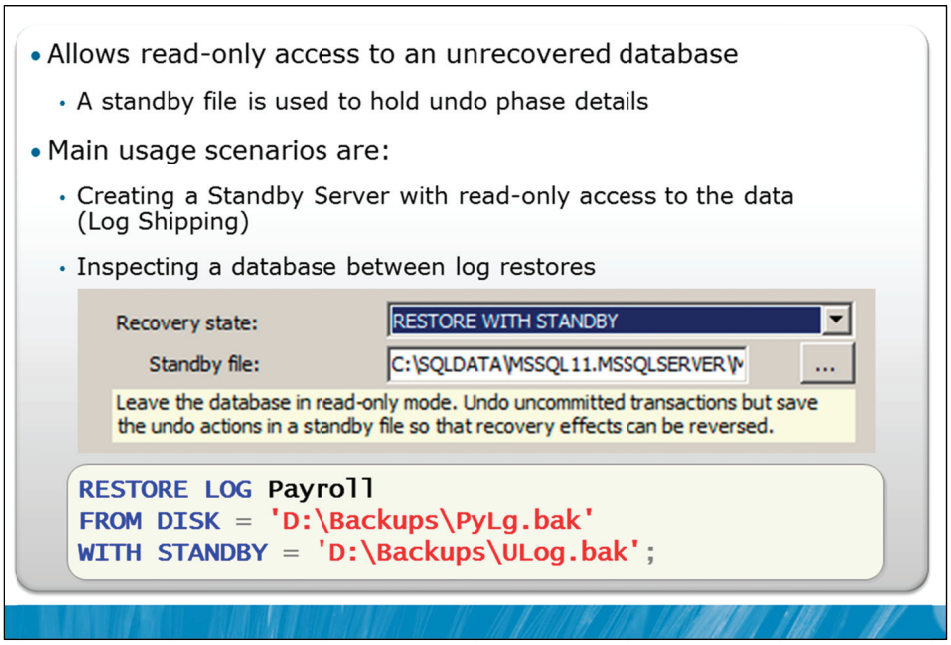
Transaction Log Restores

All transaction logs created after the last full or differential backup must be restored in chronological order with no break in the chain of backups. A break in the chain of transaction logs will cause the restore process to fail. The restore process cannot be continued after a failure and would need to be restarted.

While the database is in recovering mode, Object Explorer shows it with the words "(Restoring…)" after the name of the database, as shown in the slide example.

Question: Why is it faster to restore differential and log backups instead of restoring all log backups since the last full database backup?

WITH STANDBY Option



Key Points

SQL Server 2012 provides the ability to view the contents of a database that has not been recovered, by using the option WITH STANDBY, instead of the WITH NORECOVERY option.

Further transaction log backups can be applied to a database that has been restored WITH STANDBY. There are two common reasons for using the WITH STANDBY option.

WITH STANDBY and Log Shipping

One widely used high availability feature maintains a standby server that can be brought online quickly. This feature is called Log Shipping. The basic operation of Log Shipping is to automate the process of backing up log files on one computer, copying the log files to another computer, and restoring the log files on that other computer. The database on the second server would be nearly complete yet unusable if the restores were performed using WITH NORECOVERY. The database cannot be recovered to allow for read-only use as more transaction logs would need to be restored at a later time.

Note Log shipping is an advanced topic beyond the scope of this course but an introduction to log shipping is provided in Appendix A.

WITH STANDBY was designed to help in this situation, by performing a modified version of recovery that copies the transactions that would have been deleted by the undo phase, to an operating system file. When the next transaction log restore operation is required, SQL Server automatically reapplies the transactions from that file to the log before continuing with the log restore.

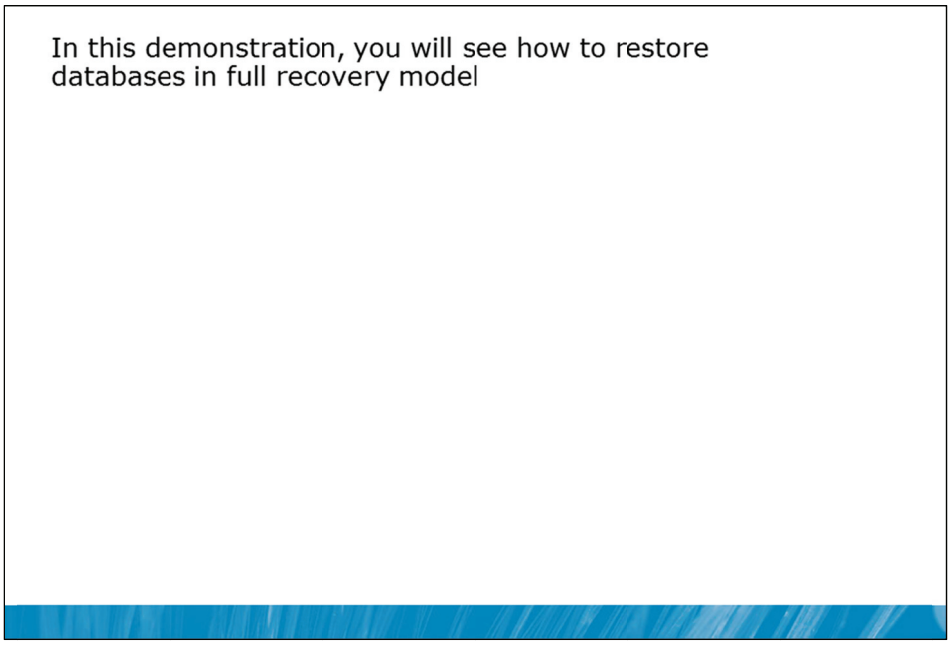
WITH STANDBY for Inspection

Imagine that a user error has caused the inadvertent deletion of some data. You may not be aware of when the error occurred. When restoring the database, you may not then know which log file contains the deletion. You can use the WITH STANDBY option on each log file restore and inspect the state of the database after each restore operation.

Question: What would be a reason to provide read-only access to a database?

Question: What would be a limitation of the WITH STANDBY option when used to permit reporting on the second standby database in a log shipping environment?

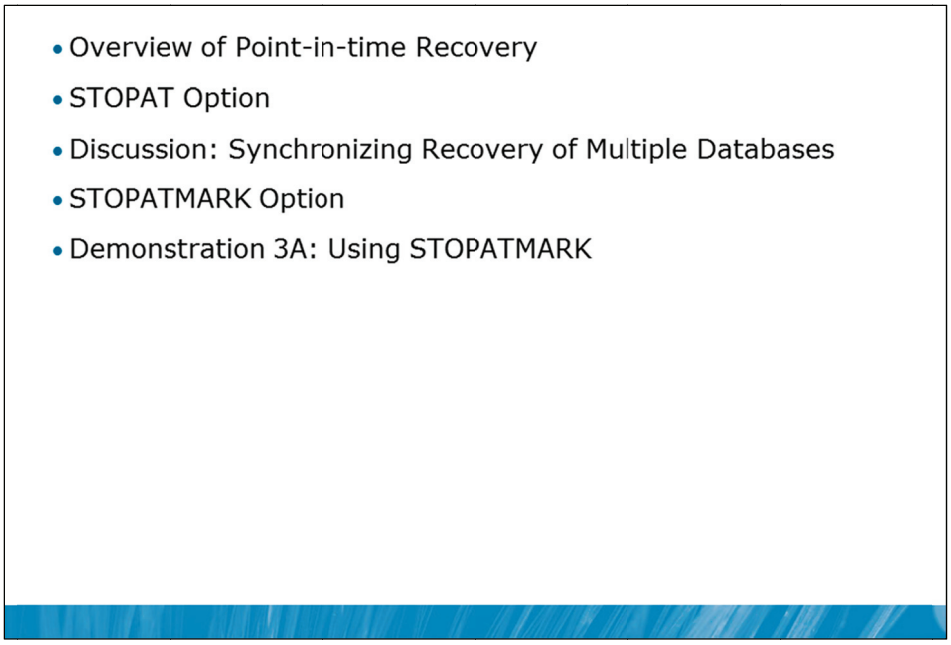
Demonstration 2A: Restoring Databases



Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_07_PRJ\10775A_07_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer. Note: The setup script for this module is intended to throw an error regarding missing files; this is normal.
4. Open the **21 – Demonstration 2A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 3
Working with Point-in-time Recovery



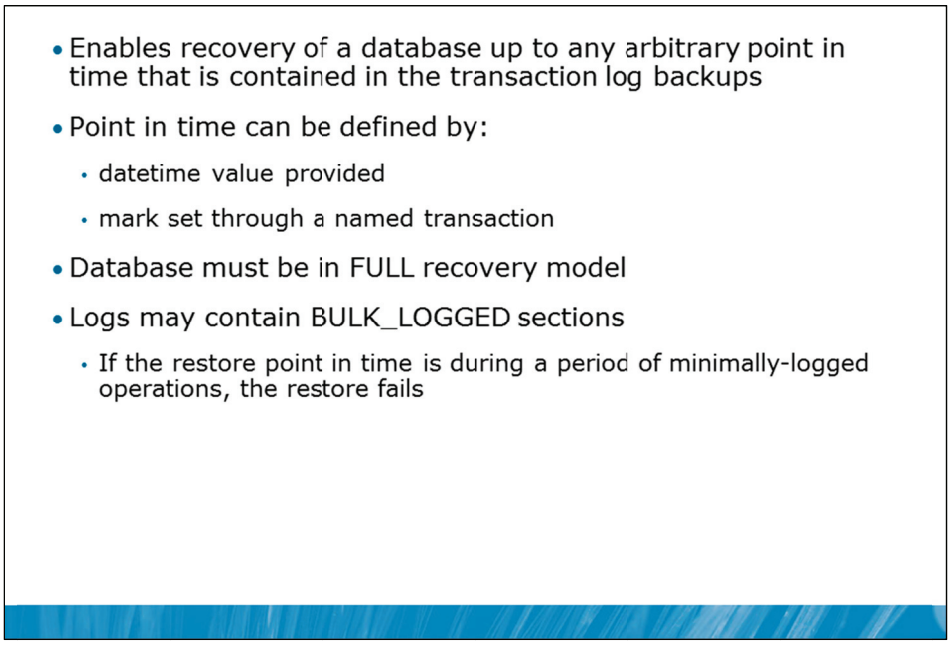
In the previous lesson, you have seen how to recover a database to the latest point in time possible. However, there are occasions when there is a need to recover the database to an earlier point in time. You have seen that you could stop the restore process after any of the backups are restored, and initiate the recovery of the database. While stopping the restore process after a restore provides a coarse level of control over the recovery point, SQL Server 2012 provides additional options that allow for more fine grained control over the recovery point.

Objectives

After completing this lesson, you will be able to:

- Describe Point-in-time Recovery.
- Implement the stop at time restore option.
- Explain the complexities surrounding the synchronization of multiple databases.
- Implement the stop at mark restore option.

Overview of Point-in-time Recovery



Key Points

Recovery of a database to the most recent point in time is the most commonly requested option. However, there can also be a need to restore a database to an earlier point in time.

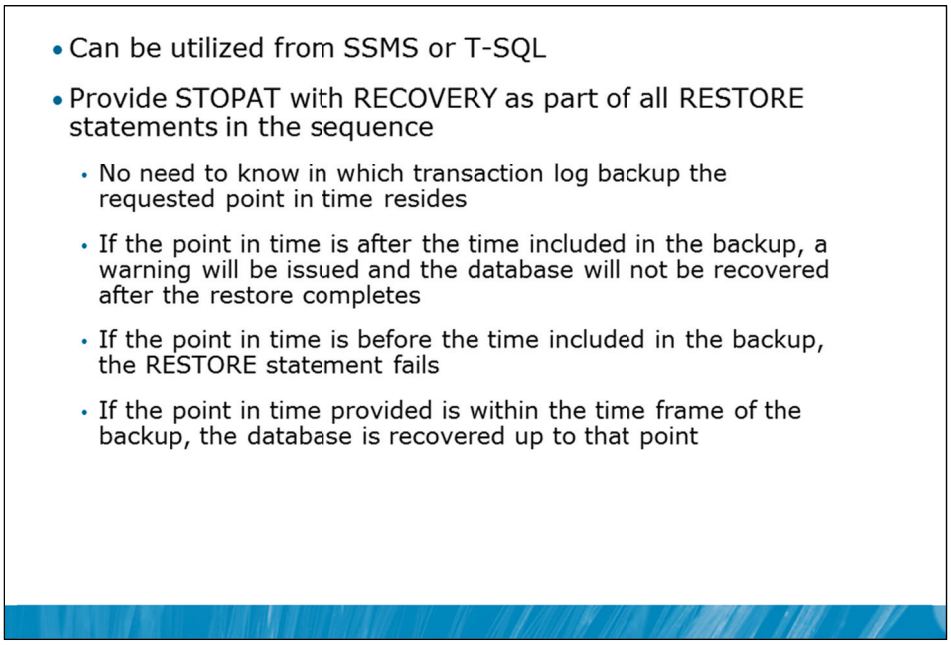
SQL Server 2012 allows the restore of a database to stop at a specified point in time and to then commence recovery. The point in time can be specified in two different ways. A datetime value can specify the exact time for the recovery point. Keep in mind that computer times tend to be approximate and that computer systems can perform a large amount of work in a short period of time. A datetime value might then not be precise enough for specifying a recovery point.

The other option provided by SQL Server 2012 is to specify a named transaction (referred to as a transaction log mark) as the recovery point.

For either of these options to work, the database needs to be in full or bulk-logged recovery model. SQL Server can only stop at points in the transaction log chain when the database is in full recovery model. If a database changes from full recovery model to bulk-logged recovery model to process bulk transactions, and is then changed back to full recovery model, the recovery point cannot be in the time that the database was in bulk-logged recovery model. If you attempt to specify a recovery point during which the database was in bulk-logged recovery model, the restore will fail and an error will be returned.

Question: What other restore option might be useful if the point in time is not known exactly and no mark was set?

STOPAT Option



Key Points

The STOPAT option is used to specify a recovery point that is based on a datetime value. Because the DBA might not know in advance which transaction log backup contains the time where the recovery needs to occur, the syntax of the RESTORE LOG command allows the RECOVERY option to be specified for each log restore command in the sequence.

Consider the following restore sequence:

```
RESTORE DATABASE database_name FROM full_backup
WITH NORECOVERY;

RESTORE DATABASE database_name FROM differential_backup
WITH NORECOVERY;

RESTORE LOG database_name FROM first_log_backup
WITH STOPAT = time, RECOVERY;

… (additional log backups could be restored here)

RESTORE LOG database_name FROM final_log_backup
WITH STOPAT = time, RECOVERY;
```

Note that the RECOVERY option is specified on each of the RESTORE LOG commands, not just on the last command.

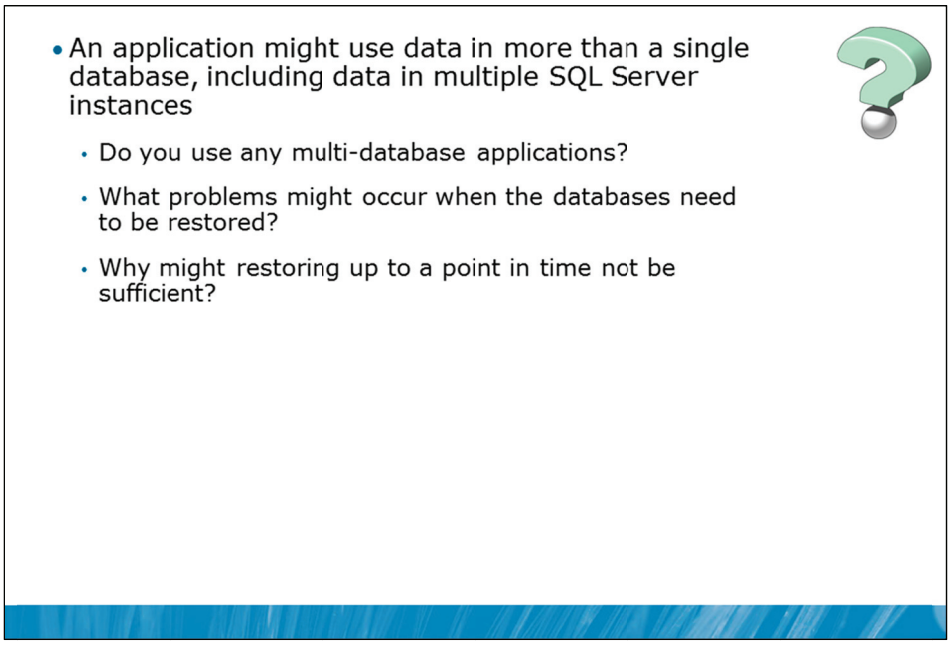
The behavior of the STOPAT and RECOVERY options is as follows:

- If the specified time is earlier than the first time in the transaction log backup, the restore command fails and returns an error.
- If the specified time is contained within the period covered by the transaction log backup, the restore command recovers the database at that time.
- If the specified time is later than the last time contained in the transaction log backup, the restore command restores the logs, sends a warning message, and the database is not recovered so that additional transaction log backups can be applied.

With the behavior described above, the database is recovered up to the requested point, even when STOPAT and RECOVERY are both specified with every restore as long as the requested point is not before the restore sequence.

Question: Why might you need to recover a database to a specific point in time?

Discussion: Synchronizing Recovery of Multiple Databases



Discussion

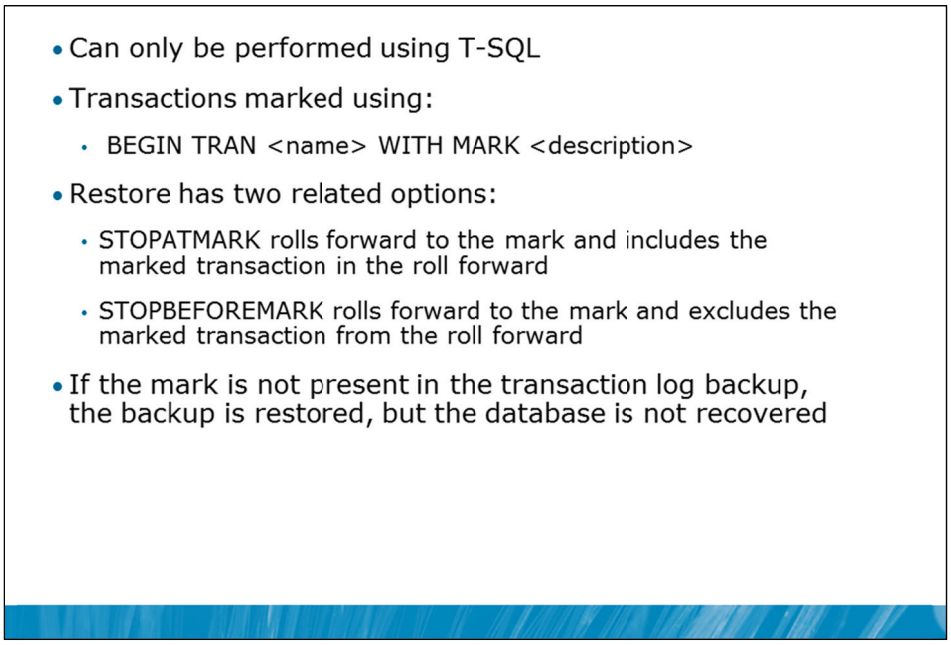
An application might use data in more than a single database, including data in multiple SQL Server instances.

Question: Do you use any multi-database applications?

Question: What problems might occur when the databases need to be restored?

Question: Why might restoring up to a point in time not be sufficient?

STOPATMARK Option



Key Points

If more precise control over the recovery point is required, the STOPATMARK option can be useful. The GUI in SSMS has no options for working with transaction log marks during restore operations. This process must be carried out using T-SQL.

Marking a Transaction

If you know in advance that you might need to recover to the point of a specific operation, you can place a mark in the transaction log to record that precise location. Consider the following command:

```
BEGIN TRAN UpdPrc WITH MARK 'Start of nightly update process';
```

In this command, a transaction is commenced but the transaction is also given the name UpdPrc, and a transaction mark with the same name as the transaction is created. The value after the WITH MARK clause is only a description and is not used in the processing of the transaction mark.

If you do not know the name of a transaction that was marked, you can query the dbo.logmarkhistory table in the msdb database.

Mark-Related Options

The STOPATMARK option is similar to the STOPAT option for the RESTORE command. SQL Server will stop at the named transaction mark and include the named transaction in the redo phase.

If you wish to exclude the transaction (that is restore everything up to the beginning of the named transaction), you can specify the STOPBEFOREMARK option instead.

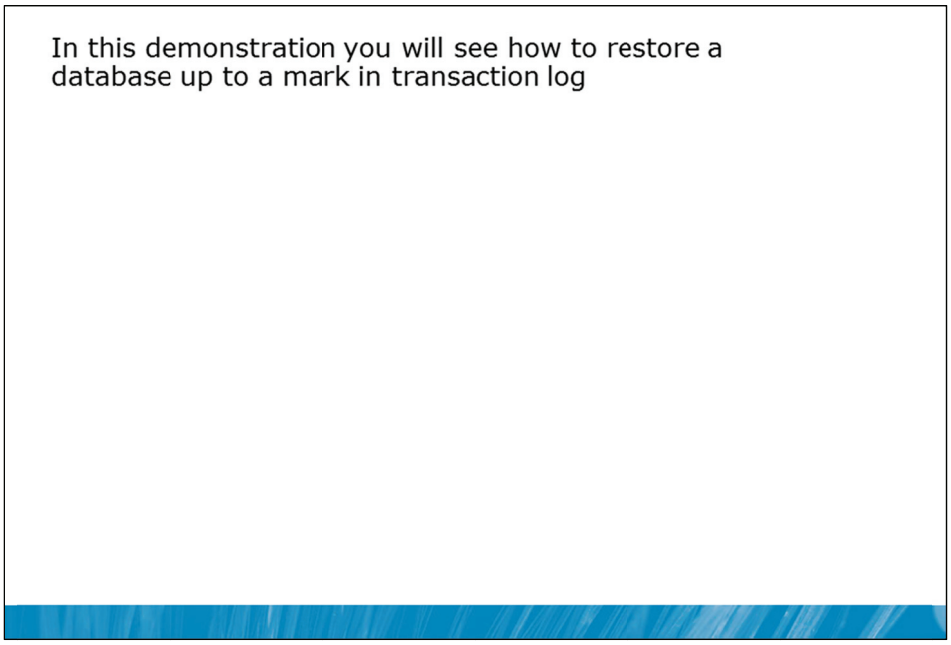
If the transaction mark is not found in the transaction log backup that is being restored, the restore completes and the database is not recovered so that other transaction log backups can be restored.

Multi-database Applications

The main use for the stop at mark feature is to restore an entire set of databases to a mutually consistent state, at some earlier point in time. If you need to perform a backup of multiple databases so that they can be recovered to a consistent point, consider marking all the transaction logs before commencing the backups.

Question: Why might STOPAT not be a good choice for synchronizing the restore of several databases and that STOPATMARK might be preferred?

Demonstration 3A: Using STOPATMARK

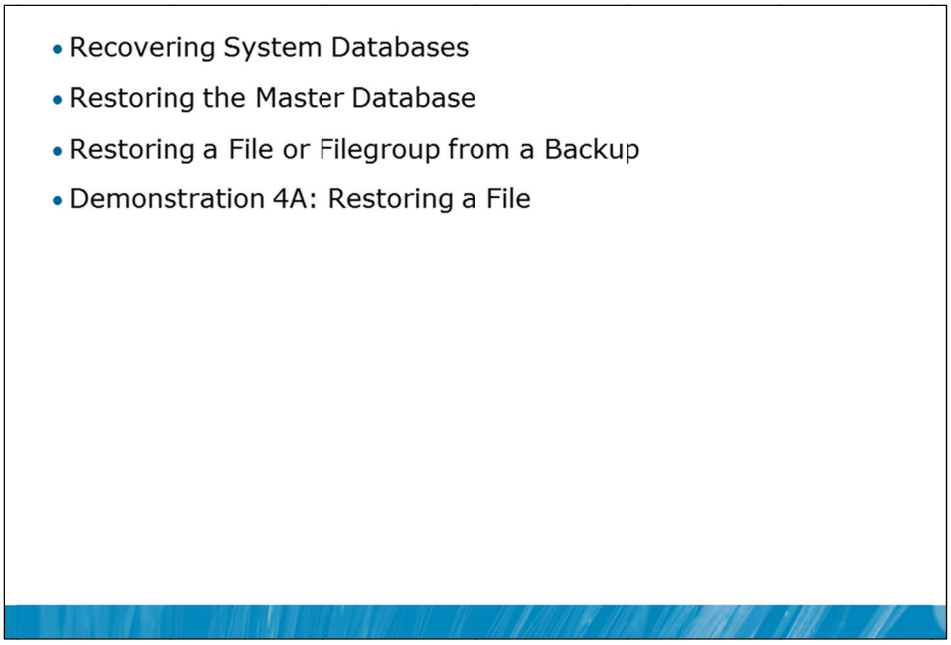


Demonstration Steps

1. If Demonstration 2A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_07_PRJ\10775A_07_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer. Note: The setup script for this module is intended to throw an error regarding missing files; this is normal.
2. Open and execute the **31 – Demonstration 3A.sql** script file from within Solution Explorer.
3. Follow the instructions contained within the comments of the script file

Lesson 4

Restoring System Databases and Individual Files



User databases are restored much more regularly than system databases, as the user databases are typically much larger, and thus more exposed to failure as they are often spread across many devices. However, failures can affect system databases and they may need to be restored. Restoring a system database is not identical to restoring user databases.

The master database is the most critical database for a SQL Server system and recovery of the master database involves more steps than the recovery of other databases.

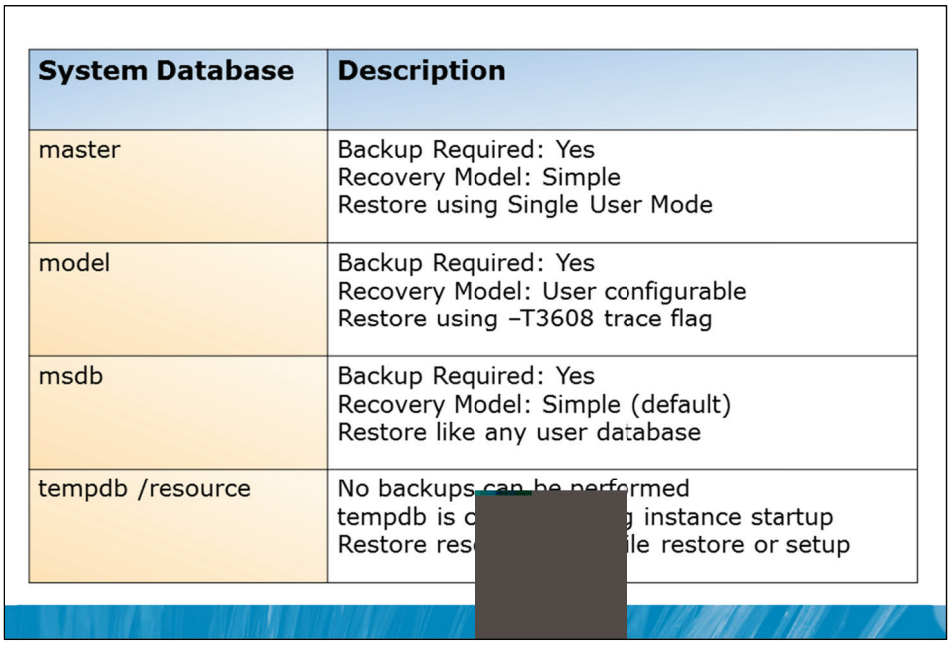
Rather than restoring entire databases, situations can arise where only a single file or filegroup needs to be restored. This can speed up the overall restore process.

Objectives

After completing this lesson, you will be able to:

- Recover System databases.
- Restore the master database.
- Restore a file or filegroup from a backup.

Recovering System Databases



Key Points

The recovery process for all system databases is not identical. Each system database has specific recovery requirements:

master

The master database holds all system level configurations. SQL Server requires the master database before a SQL Server instance can run at all. A special procedure needs to be followed to restore the master database. The process for restoring the master database is discussed in the next topic.

model

The model database is the template for all databases that are created on the instance of SQL Server. When the model database is corrupt, the instance of SQL Server cannot start. This means that a normal restore command cannot be used to recover the model database if it becomes corrupted. In the case of a corrupt model database, the instance must be started with the -T3608 trace flag as a command-line parameter. This trace flag only starts the master database. Once SQL Server is running, the model database can be restored using the normal RESTORE DATABASE command.

msdb

The msdb database is used by SQL Server Agent for scheduling alerts and jobs, and for recording details of operators. The msdb database also contains history tables, such as the history tables that record details of backup and restore operations. If the msdb database becomes corrupt, SQL Server Agent will not start. msdb can be restored like user databases, using the RESTORE DATABASE command and then the SQL Server Agent service can be restarted.

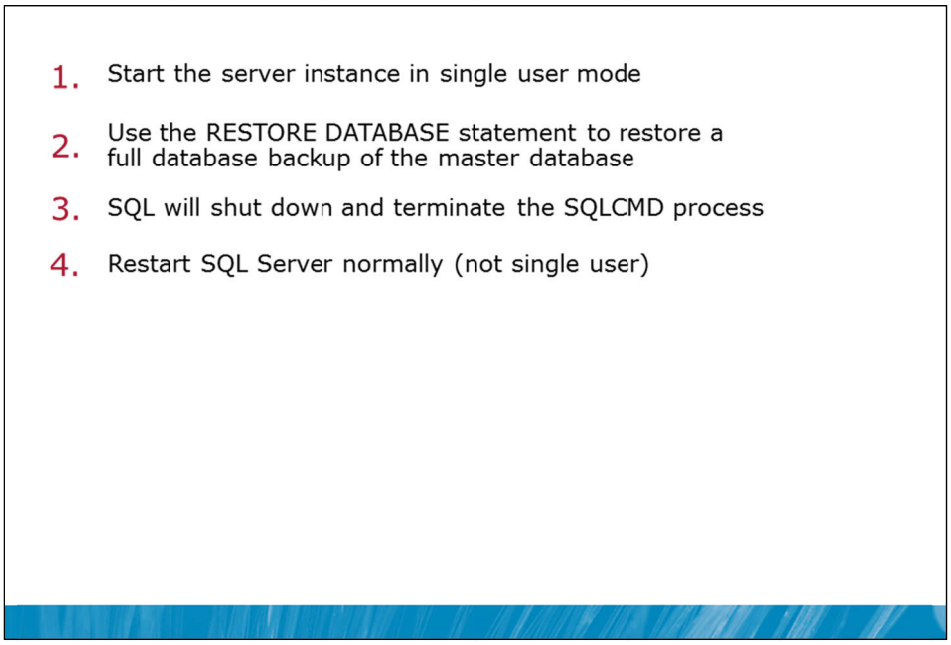
resource

The resource database is a read-only database that contains copies of all system objects that ship with Microsoft SQL Server 2012. No backup operations can be performed on this database, and it is a hidden database. It can, however, be corrupted by failures in areas such as I/O subsystems or memory. If the resource database is corrupt, it can be restored by a file-level restore in Windows or by running the setup program for SQL Server.

tempdb

The tempdb database is a workspace for holding temporary or intermediate result sets. This database is re-created every time an instance of SQL Server is started. When the server instance is shut down, any data in tempdb is deleted permanently. No backup operations can be performed on the tempdb database but as it is recreated with every restart of the instance, a restart of the instance is enough to recover the tempdb in case of corruption.

Restoring the master Database



Key Points

The master database is integral to the operation of SQL Server. As SQL Server will not start if the master database is missing or corrupt, you cannot execute a standard RESTORE DATABASE command to restore the master database in the case of a missing or corrupt database.

Recovering the master Database

Before starting this process, some version of a master database must exist so that the SQL Server instance will start at all. If your master database becomes corrupted, a temporary master database must be created first. This temporary master database doesn't need to have the correct configuration as it will be only used to start up the instance. The correct master database will be restored afterwards using the process described on the slide.

There are three ways to obtain a temporary master database. You can use the SQL Server setup program to rebuild the system databases, either from the location that you installed SQL Server from, or by running the setup program found at: Microsoft SQL Server\110\Setup\Bootstrap\SQL11\setup.exe. (This path is approximate and may change in future).

Note The setup program will overwrite all system databases and all will need to be restored later.

Another way to obtain a temporary master database is to use file-level backup of the master database files to restore the master database. This file-level backup must have been taken when the master database was not in use, that is, when SQL Server was not running, or by using the VSS service.

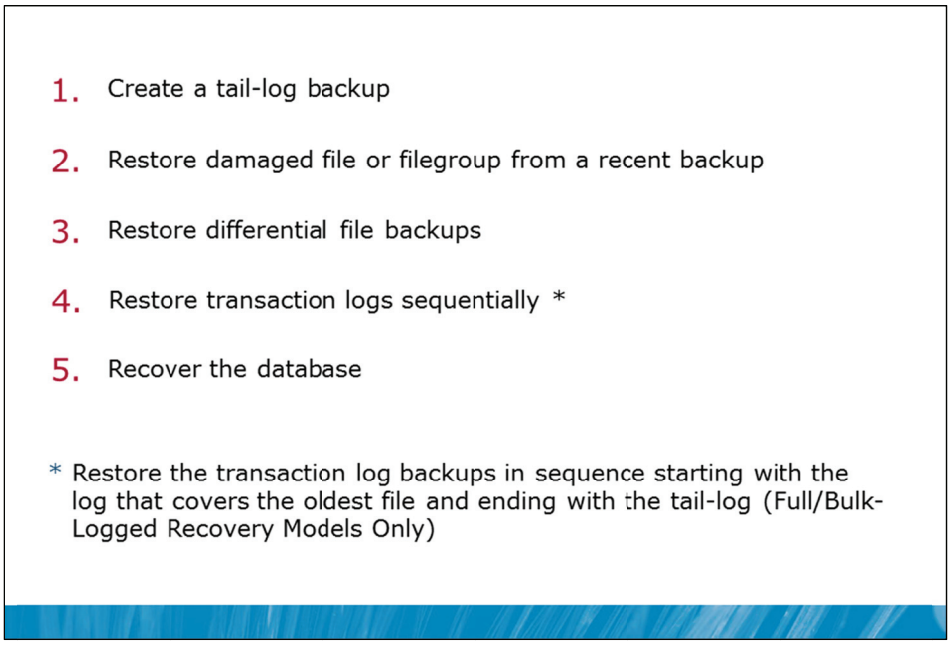
Note Copying the master database from another instance is not supported. The VSS service is out of scope for this course.

The final option is to locate a master.mdf database from the Templates folder located in the MSSQL\Binn folder for each instance.

Once a temporary master database has been put in place, use the following procedure to recover the correct master database:

- Start the server instance in single-user mode. (SQL Server configuration manager can be used to start SQL Server in single user mode by using the –m startup option).
- Use a RESTORE DATABASE statement to restore a full database backup of master. (In single-user mode, it is recommended that you enter the RESTORE DATABASE statement using the sqlcmd utility).
- After the master database is restored, the instance of SQL Server will shut down and terminate your sqlcmd connection.
- Remove the single-user startup parameter.
- Restart SQL Server.

Restoring a File or Filegroup from a Backup



Key Points

Restoring individual files or filegroups from backups, instead of restoring entire databases, can often be a much faster option when corruption has occurred or when files or filegroups are missing.

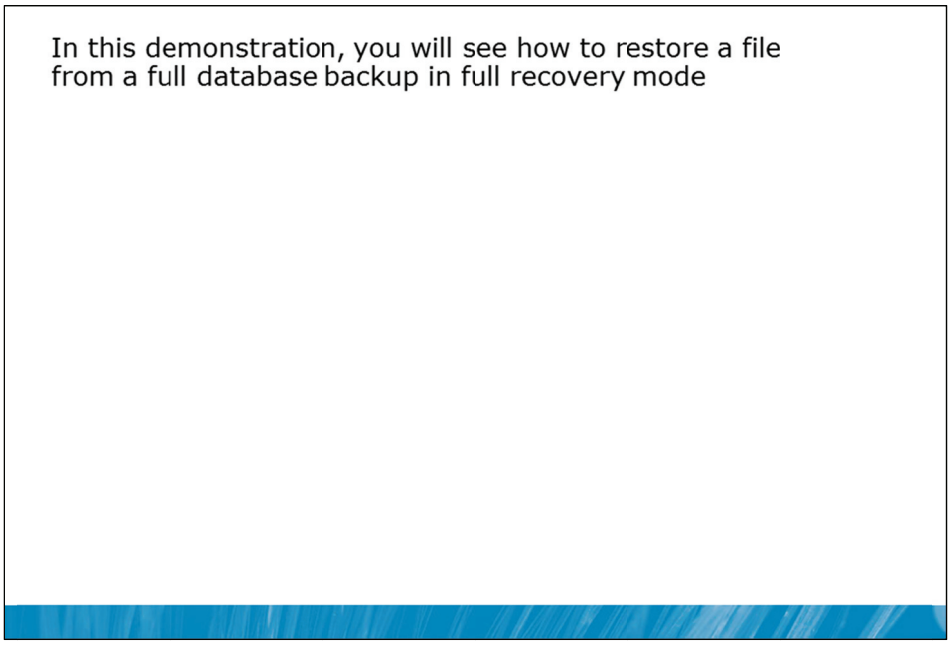
Note While it is possible to backup only files or filegroups, there is no need to have performed file or filegroup backups before restoring individual files or filegroups. SQL Server can extract specific database files out of a full backup or a differential backup.

Restore Process

1. Create a tail-log backup of the active transaction log. (If you cannot do this because the log has been damaged, you must restore the whole database or restore to an earlier point-in-time).
2. Restore each damaged file from the most recent file backup of that file.
3. Restore the most recent differential file backup, if any, for each restored file.
4. Restore transaction log backups in sequence, starting with the backup that covers the oldest of the restored files and ending with the tail-log backup created in step 1.
5. Recover the database.

You must restore the transaction log backups that were created after the file backups to bring the database back to a consistent state. The transaction log backups can be rolled forward quickly, because only the changes that apply to the restored files or filegroups are applied. Undamaged files are not copied and then rolled forward. However, the whole chain of log backups still needs to be processed.

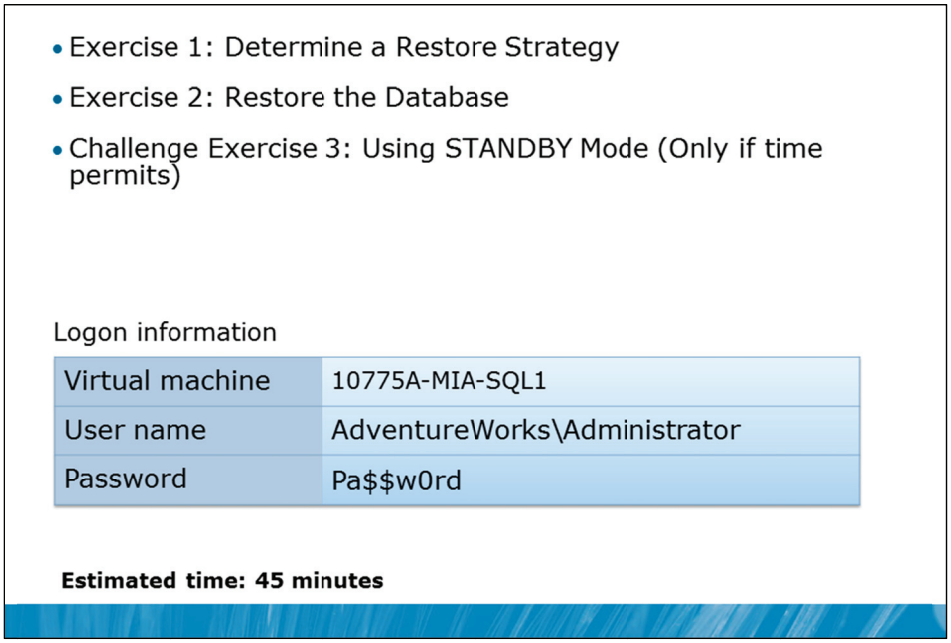
Demonstration 4A: Restoring a File



Demonstration Steps

1. If Demonstration 2A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_07_PRJ\10775A_07_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer. Note: The setup script for this module is intended to throw an error regarding missing files; this is normal.
2. Open and execute the **41 – Demonstration 4A.sql** script file from within Solution Explorer.
3. Follow the instructions contained within the comments of the script file.

Lab 7: Restoring SQL Server 2012 Databases



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_07_PRJ\10775A_07_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Note The setup script for this module is intended to throw an error regarding missing files. This is normal.

Lab Scenario

You have been provided with a series of backups taken from a database on another server that you need to restore to the Proseware, Inc. server with the database name MarketYields. The backup file includes a number of full, differential, and log backups. You need to identify backups contained within the file, determine which backups need to be restored, and perform the restore operations. When you restore the database, you need to ensure that it is left as a warm standby, as additional log backups may be applied at a later date.

If you have time, you should test the standby operation.

Exercise 1: Determine a Restore Strategy

Scenario

You need to restore a database backup from another instance to the Proseware instance. You have been provided with a backup file containing multiple full, differential, and log backups. In this exercise you need to determine which backups are contained within the file and determine which backups need to be restored and in which order.

The main tasks for this exercise are as follows:

1. Review the backups contained within the backup file.
2. Determine how the restore should be performed.

▶ Task 1: Review the backups contained within the backup file

- Use the HEADERONLY option of the RESTORE command to identify the backups that are contained within the file D:\MSSQLSERVER\MarketYields.bak.

▶ Task 2: Determine how the restore should be performed

- Determine which backup need to be restored and in which order.

Results: After this exercise, you should have identified the backups that need to be restored.

Exercise 2: Restore the Database

Scenario

You have determined which backups need to be restored. You now need to restore the database MarketYields to the Proseware instance from the backups that you have decided upon. You will leave the database in STANDBY mode.

The main task for this exercise is as follows:

1. Restore the database.

▶ Task 1: Restore the database

- Using SSMS, restore the MarketYields database using the backups that you determined were needed in Exercise 1. Make sure you use the STANDBY option with a STANDBY filename of L:\MKTG\Log_Standby.bak. You will need to move the mdf file to the folder D:\MKTG and the ldf file to the folder L:\MKTG.

- In Object Browser refresh the list of databases and check the status of the MarketYields database on the Proseware instance.

Results: After this exercise, you should have restored the database in STANDBY mode.

Challenge Exercise 3: Using STANDBY Mode (Only if time permits)

Scenario

In this exercise, you will ensure that the STANDBY mode works as expected. You will access the database and then restore another log file to make sure the database can continue to be restored.

The main tasks for this exercise are as follows:

1. Execute queries against the STANDBY database to ensure it is accessible.
2. Restore another log file, leaving the database in STANDBY mode.

▶ Task 1: Execute queries against the STANDBY database to ensure it is accessible

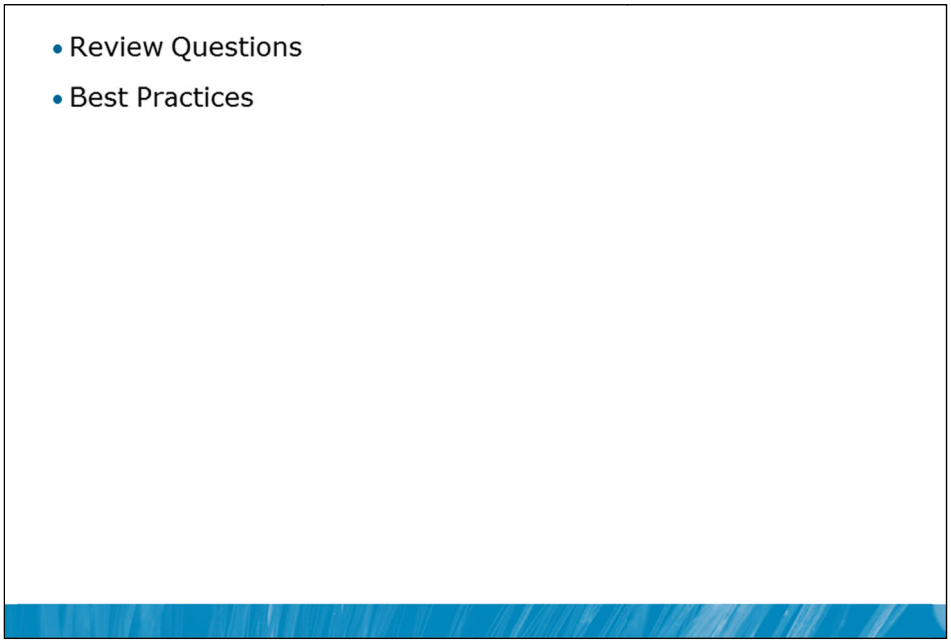
- Open a query window against the MarketYields database on the Proseware instance.
- Select a count of the rows in the LogData table.
- Close the query window.

▶ Task 2: Restore another log file, leaving the database in STANDBY mode

- Restore the log file D:\MSSQLSERVER\MarketYields_log.bak. Ensure you leave the database in STANDBY mode.
- In Object Browser refresh the list of databases and check the status of the MarketYields database on the Proseware instance.

Results: After this exercise, you should have tested the STANDBY capability.

Module Review and Takeaways



Review Questions

1. What are the three phases of the restore process?
2. What is always performed before a database starts up and goes ONLINE?

Best Practices related to a particular technology area in this module

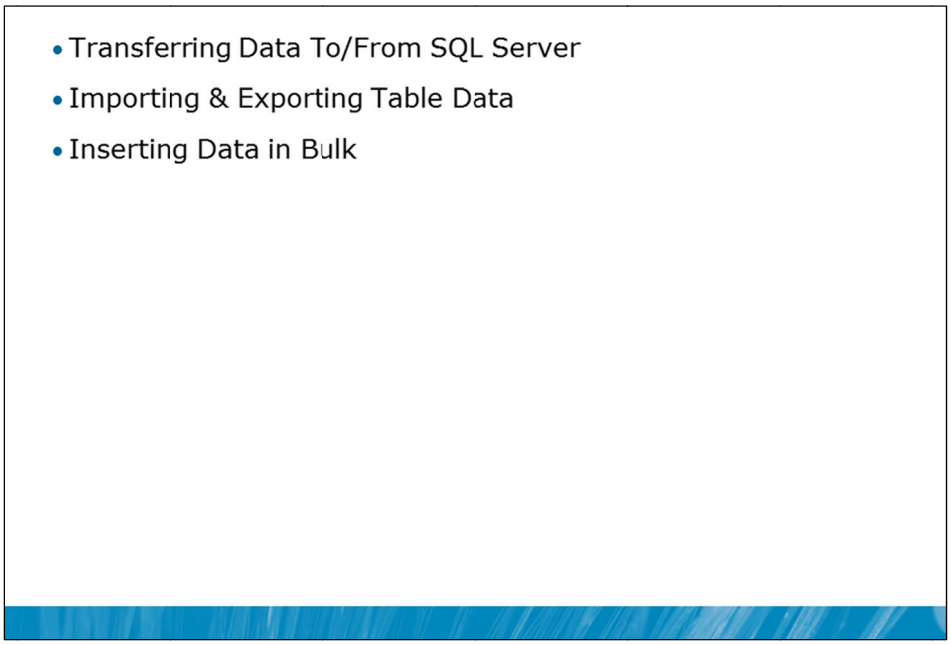
1. Don't forget to backup the tail of the log before starting a restore sequence.
2. Use differential restore to speed up the restore process if available.
3. Use file level restore to speed up restores when not all database files are corrupt.
4. Perform regular database backups of master, msdb and model system databases.
5. Create a disaster recovery plan for your SQL Server and test restoring databases regularly.

Module 8

Importing and Exporting Data

Contents:

Module Overview



While a great deal of data that resides in a Microsoft® SQL Server® system may be entered directly by users that are running application programs, there is also a need to move data in other locations to and from SQL Server.

SQL Server provides a set of tools that can be used to transfer data in and out of SQL Server. Some of these tools, such as the bcp utility and SQL Server Integration Services are external to the database engine, and other tools such as the BULK INSERT statement, and the OPENROWSET function, are implemented within the database engine. In this module, you will briefly explore each of these tools.

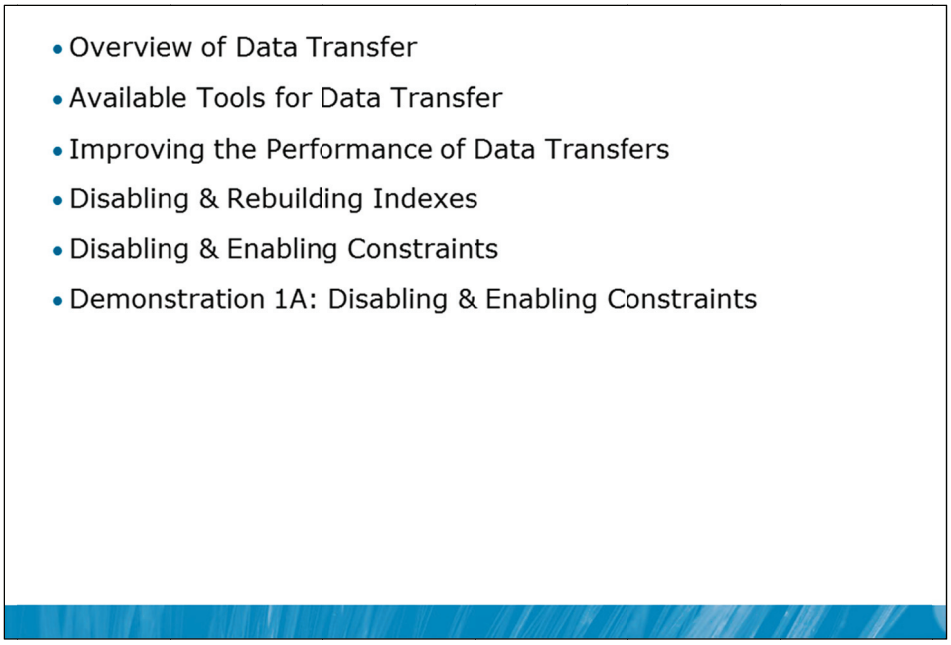
On occasions, large amounts of data need to be imported into SQL Server. While the default settings in SQL Server can be used while importing the data, higher performance can be achieved by exerting control over how constraints, triggers, and indexes are used during the import process.

Objectives

After completing this lesson, you will be able to:

- Transfer data to and from SQL Server.
- Import and export table data.
- Insert data in bulk and optimize the bulk insert process.

Lesson 1
Transferring Data To/From SQL Server



The first step in learning to transfer data in and out of SQL Server is to become familiar with the processes involved and with the tools that SQL Server provides to implement data transfer.

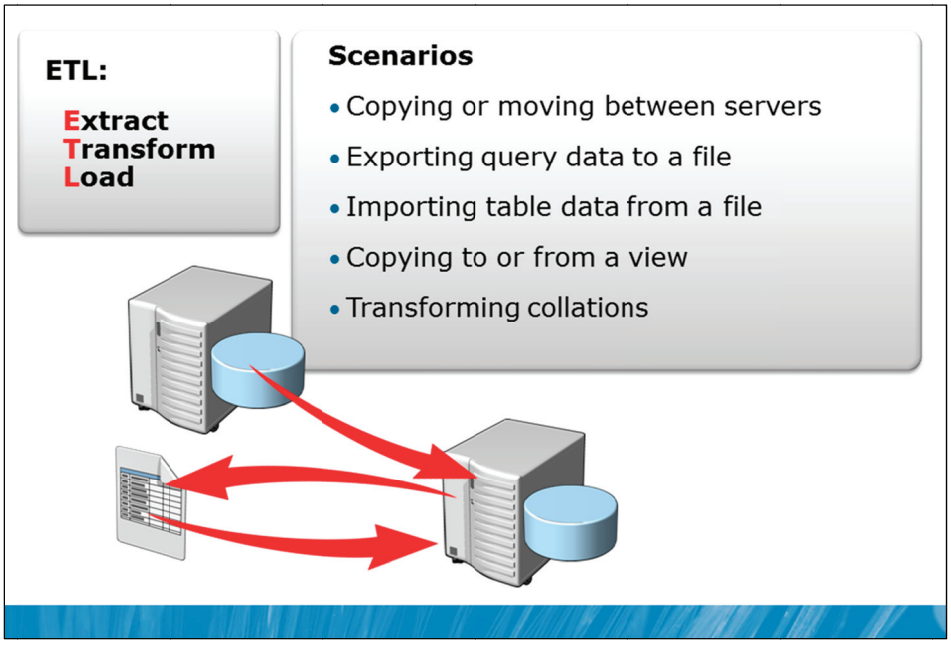
When large amounts of data need to be inserted into SQL Server tables, the default settings for constraints, triggers, and indexes are not likely to provide the best performance possible. You may achieve higher performance by controlling when the checks that are made by constraints are carried out and by controlling when the index pages for a table are updated.

Objectives

After completing this lesson, you will be able to:

- Explain core data transfer concepts.
- Describe the tools that SQL Server provides for data transfer.
- Improve the performance of data transfers.
- Disable and rebuild indexes.
- Disable and re-enable constraints.

Overview of Data Transfer



Key Points

Not all data can be entered row by row by database users. Often data needs to be imported from external data sources such as other database servers or from operating system files. Users often also request that data from tables in databases is exported to operating system files. In earlier modules, you have seen how collations can cause issues when misconfigured. Correcting the collation of a database also often requires the export and re-import of the data from the database.

Data Transfer Steps

Although not all data transfer requirements are identical, there is a standard process that most data transfer tasks follow. The main steps of this process are:

- Extracting data from a given data source.
- Transforming the data in some way to make it suitable for the target system.
- Loading the data into the target system.

Together, these three steps are commonly referred to as an Extract, Transform, Load (ETL) process. Tools that implement these processes are commonly referred to as ETL tools.

Note In some situations, an Extract, Load, Transform (ELT) process might be more appropriate. For example, you may decide to perform data transformations once the data has been loaded into the database engine rather than before it is loaded.

Extracting Data

While there are other options, extracting data typically involves the execution of queries on a source system to retrieve the data, or opening and reading operating system files. Another option would involve the querying of views provided by the source system.

During the extraction process, there are two common aims:

- Avoid excessive impact on the source system. For example, do not read entire tables of data when you only need to read selected rows. Also, do not continually re-read the same data, and avoid the execution of statements that block users of the source system in any way.
- Ensure consistency of the data extraction. For example, do not include one row from the source system more than once in the output of the extraction.

Transforming Data

The transformation phase of an ELT process will generally involve several steps, such as the following:

- Data might need to be cleansed. For example, you might need to remove erroneous data or provide default values for missing columns.
- Lookups might need to be performed. For example, the input data might include the name of a customer, but the database might need an ID for the customer.
- Data might need to be aggregated. For example, the input data might include every transaction that occurred on a given day, but the database might need only daily summary values.
- Data might need to be de-aggregated. This is often referred to as data allocation. For example, the input data might include quarterly budgets, but the database might need daily budgets.

In addition to these common operations, data might need to be restructured in some way. One common requirement is that the data might need to be pivoted or unpivoted. For example, the input data might reside in a table such as the following:

```
Object     Attribute      Value
1          FirstName      David
1          Age            64
1          Gender         M
1          LastName       Pelton
2          FirstName      Erin
2          Age            47
2          Gender         F
2          LastName       Hagens
```

The database might, however, require the same data in this format:

```
PersonID FirstName  Age  Gender  LastName
1        David      64   M       Pelton
2        Erin       47   F       Hagens
```

This transformation is an example of pivoting data, where rows become columns or columns become rows.

Loading Data

Once data is in an appropriate format, it can be loaded into the target system. Instead of performing row by row insert operations for the data, special options for loading data in bulk might be used. In addition, temporary configuration changes may be made to improve the performance of the load operation.

Question: What other types of aggregation might need to be performed on data during the transformation phase?

Available Tools for Data Transfer

Key Points

SQL Server provides a set of tools for performing data transfer tasks. It is important to understand where to use each of the tools.

Bulk Copy Program (bcp)

The Bulk Copy Program (bcp) can be used to import large numbers of new rows from an operating system data file into a SQL Server table, or to export data from a SQL Server table to an operating system file. Although the bcp utility can be used with a T-SQL queryout option, which specifies the rows to be exported, the normal use of bcp does not require T-SQL knowledge.

BULK INSERT

The BULK INSERT statement is a T-SQL command that is used to import data directly from an operating system data file into a database table. The BULK INSERT statement differs from bcp in a number of ways. First, the BULK INSERT statement is executed from within T-SQL whereas the bcp utility is a command line utility. Also, while the bcp utility can be used for both import and output, the BULK INSERT statement can only be used for data import.

OPENROWSET (BULK)

OPENROWSET is a table-valued function that is used to connect to and retrieve data from OLE-DB data sources. Full details of how to connect to the data source need to be provided as parameters to the OPENROWSET function. OPENROWSET can be used to connect to other types of database engine.

SQL Server provides a special OLE-DB provider called BULK that can be used with the OPENROWSET function. The BULK provider allows the import of entire documents from the file system.

Import/Export Wizard

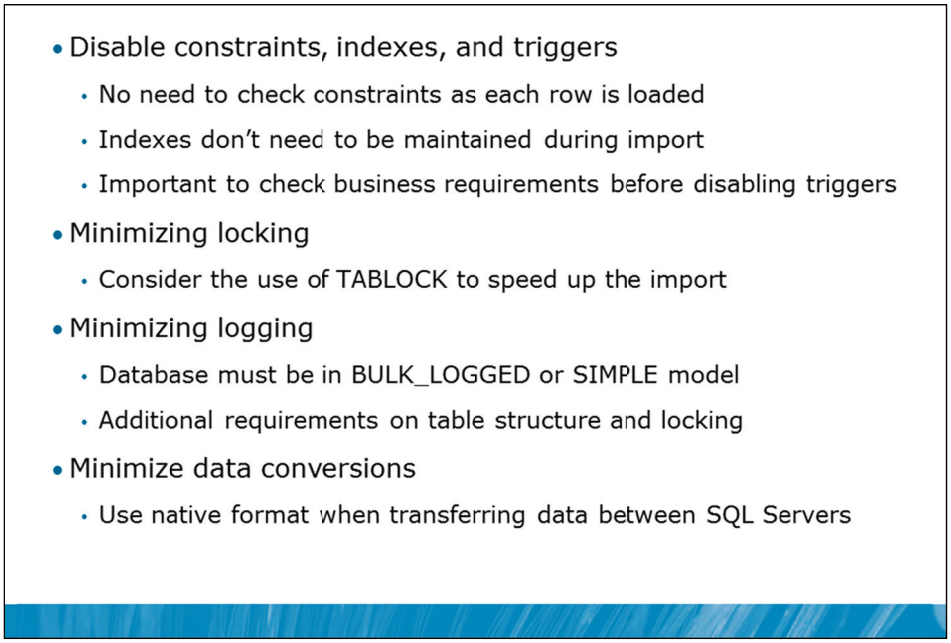
SQL Server Integration Services (SSIS) is an ETL tool that is supplied with SQL Server. SSIS is capable of connecting to a wide variety of data sources and destinations and is capable of performing complex transformations on data. SSIS provides many tasks and transformations out of the box and can also be extended by the use of custom .NET components and scripts. SQL Server also provides the Import/Export Wizard, which is a simple method of creating SSIS packages, without the need to use the SSIS design tools.

XML Bulk Load

The XML Bulk Load provider can be used to import XML data as a binary stream within a T-SQL statement. The data can be inserted directly into a column in an existing row of a database table.

Question: When would you choose SSIS over bcp?

Improving the Performance of Data Transfers



Key Points

If constraints, indexes, and triggers are enabled on the tables that are the targets of data transfers, the data values need to be checked for every row that is imported. This constant checking can substantially slow down SQL Server data transfers.

Disabling Constraints, Indexes, and Triggers

Rather than checking each value that is imported, or updating each index for every row that is inserted, higher overall performance can often be achieved by disabling the process of checking or index updating, until all the data is loaded, and then performing that work one time at the end of the import process.

For example, consider a FOREIGN KEY constraint that is used to ensure that the relevant customer does in fact exist, whenever a customer order is inserted into the database. While this reference could be checked for each customer order, consider that a customer might have thousands of customer orders. Instead of checking each value as it is inserted, the customer reference could be checked as a single lookup after the overall import process, to cover all customer orders that refer to that customer.

Only CHECK and FOREIGN KEY constraints can be disabled. The process for disabling and re-enabling constraints will be discussed later in this lesson.

Similar to the way that avoiding lookups for FOREIGN KEY constraints during data import can improve performance, avoiding constant updating of indexes can also improve performance. In many cases, rebuilding the indexes after the import process is complete will be much faster than updating the indexes as the rows are imported. The exception to this situation is when there is a much larger number of rows already in the table than are being imported.

Triggers are commands that are executed when data is modified. It is important to decide if the processing that the triggers perform would also be better processed in bulk after the import, rather than as each insert occurs.

Control the Locking Behavior

By default, SQL Server manages the granularity of the locks it acquires during the execution of commands. SQL Server starts with row level locking and only tries to escalate when a significant number of rows are locked within a table. Managing large numbers of locks occupies resources which could be used to minimize the execution time for queries. As the data in tables that are the target of bulk-import operations are normally only accessed by the process that is importing the data, the advantage of row-level locking is often not present. For this reason, it may be advisable to lock the entire table by using a TABLOCK query hint during the import process.

Use Minimal Logging Whenever Possible:

The operation of the transaction log was discussed in Module 5. Minimal logging is a special operation that can provide substantial performance improvements in some operations such as bulk-imports. As well as making the operations faster, minimal logging helps avoid excessive log growth during large import operations.

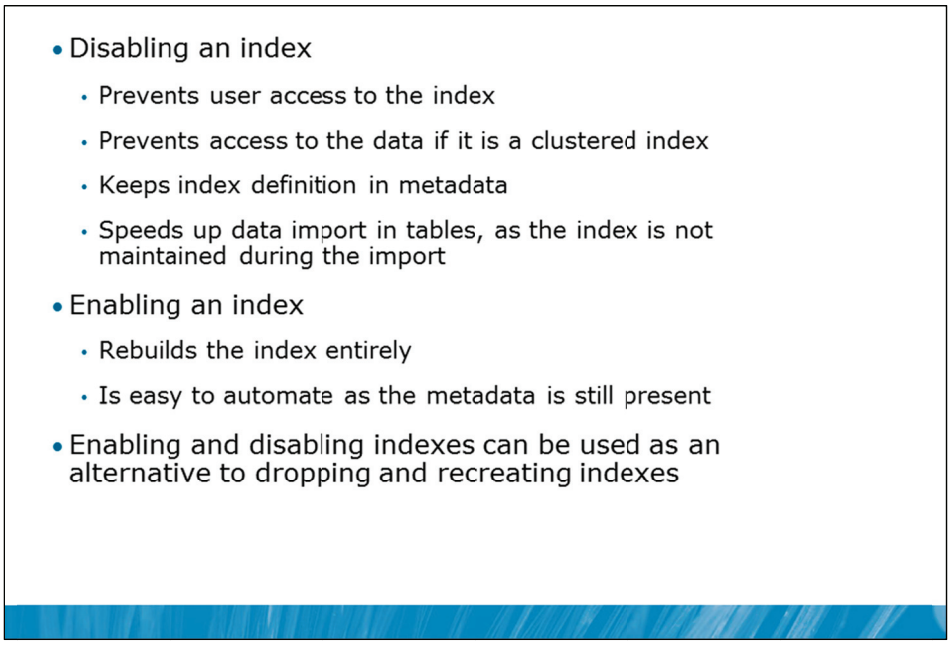
Not all commands can use minimal logging. While not an exhaustive list, the items below indicate the types of restrictions that must be met for minimal logging to be applied:

- The table is not being replicated.
- Table locking is specified (using TABLOCK).
- If the table has no clustered index but has one or more nonclustered indexes, data pages are always minimally logged. How index pages are logged, however, depends on whether the table is empty.
- If the table is empty, index pages are minimally logged.
- If table is non-empty, index pages are fully logged.
- If the table has a clustered index and is empty, both data and index pages are minimally logged.
- If a table has a clustered index and is non-empty, data pages and index pages are both fully logged regardless of the recovery model.

Note Index types including clustered and nonclustered indexes are discussed in course 10776A: Developing Microsoft SQL Server 2012 Databases.

Question: What would the main problem with the transaction log be, if full logging occurs during a bulk-import operation?

Disabling & Rebuilding Indexes



Key Points

Prior to SQL Server 2005, indexes needed to be dropped to prevent them from being updated as the data in the table was updated. The problem with dropping the index, is that when you need to put the index back in place by recreating it, you would need to know exactly how the index was configured.

Disabling an Index

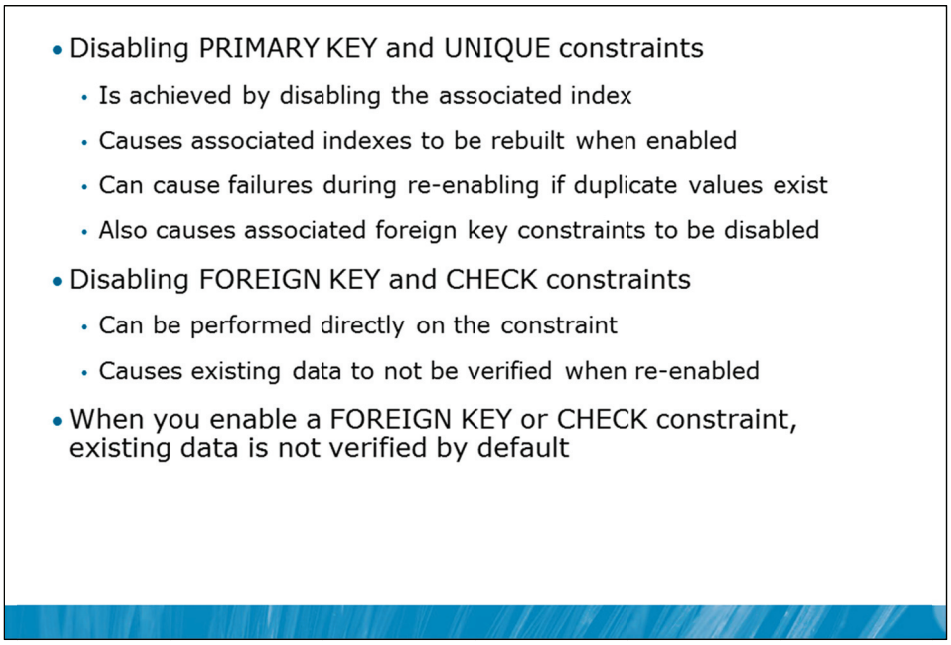
Since SQL Server 2005, an option exists to disable an index. Rather than totally dropping the index details from the database, disabling an index leaves the metadata about the index in place, and stops the index from being updated. Queries that are executed by users will not use disabled indexes.

The major advantage of disabling an index instead of dropping it is that the index can be put back into operation by a rebuild operation. When you rebuild an index, you do not need to know details of how it is configured. This makes it much easier to create administrative scripts that stop indexes being updated while large import or update operations are taking place, and that put the indexes back into operation after those operations have completed.

Note One special type of index known as a clustered index relates to how the table is structured, rather than to a separate index that speeds up the location of rows within the table. If a clustered index is disabled, the table becomes unusable until the index is rebuilt.

Question: What is the main advantage of disabling and enabling indexes compared to dropping and recreating an index during bulk-imports?

Disabling & Enabling Constraints



Key Points

PRIMARY KEY constraints define the column or columns that uniquely identify each row in a table. UNIQUE constraints ensure that a column or columns do not contain duplicate values. SQL Server creates indexes to help it to enforce these constraints.

Disabling PRIMARY KEY or UNIQUE Constraints

To disable a PRIMARY KEY or UNIQUE constraint, you need to disable the index that is associated with the constraint. This is typically only useful with nonclustered PRIMARY KEY constraints. When the constraint is re-enabled, the associated indexes are rebuilt automatically. If duplicate values are found during the rebuild, the re-enabling of the constraint will fail. For this reason, if you disable these constraints while importing data, you need to be sure that the data that is being imported will not violate the rules that the constraints enforce.

Note If a table has a primary key enforced with a clustered index, disabling the index associated with the constraint would prevent access to any data in the table.

FOREIGN KEY constraints are used to make sure that those entities in one table that are referred to by entities in another table, actually exist. For example, a supplier must exist before a purchase order could be entered for the supplier. FOREIGN KEY constraints use PRIMARY KEY or UNIQUE constraints while checking the references. If you disable the PRIMARY KEY or UNIQUE constraint that a FOREIGN KEY reference points to, the FOREIGN KEY constraint will also automatically be disabled. However, when you re-enable the PRIMARY KEY or UNIQUE constraint, FOREIGN KEY references that use these constraints will not also be automatically re-enabled.

Disabling FOREIGN KEY and CHECK Constraints

CHECK constraints are used to limit the values that can be contained in a column or the relationship between the values in multiple columns in a table.

Note Constraints are described in course 10776A: Developing Microsoft SQL Server 2012 Databases.

Both FOREIGN KEY and CHECK constraints can be disabled and enabled using the CHECK and NOCHECK options of the ALTER TABLE statement. For example, consider the following code sample that disables a CHECK constraint named SalaryCap on a table called Person.Salary:

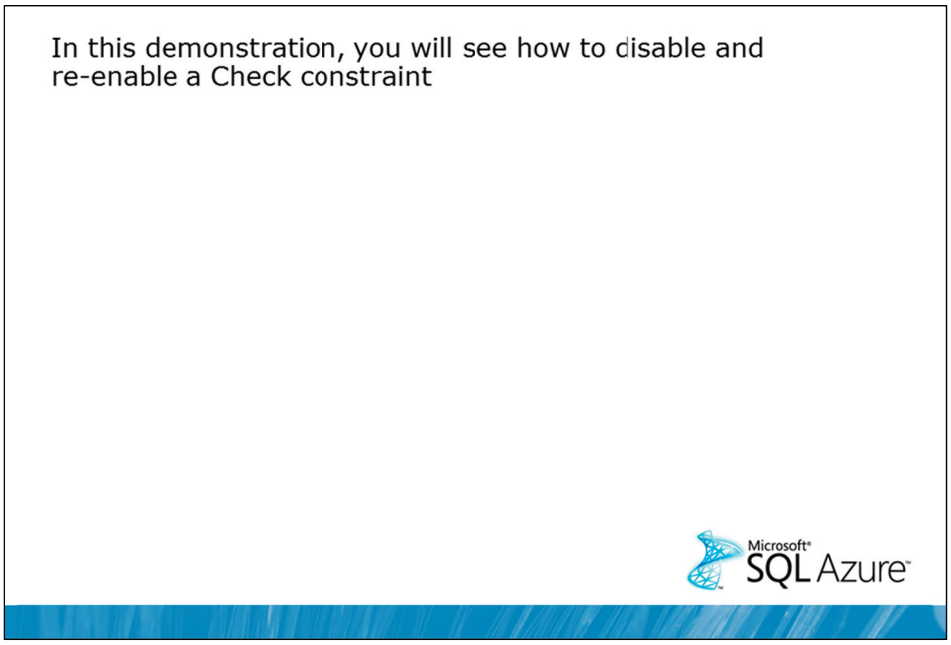
```
ALTER TABLE Person.Salary NOCHECK CONSTRAINT SalaryCap;
```

The following code is used to re-enable the constraint:

```
ALTER TABLE Person.Salary CHECK CONSTRAINT SalaryCap;
```

Question: Why do referencing foreign key constraints get disabled when the referenced PRIMAY KEY or UNIQUE constraints get disabled?

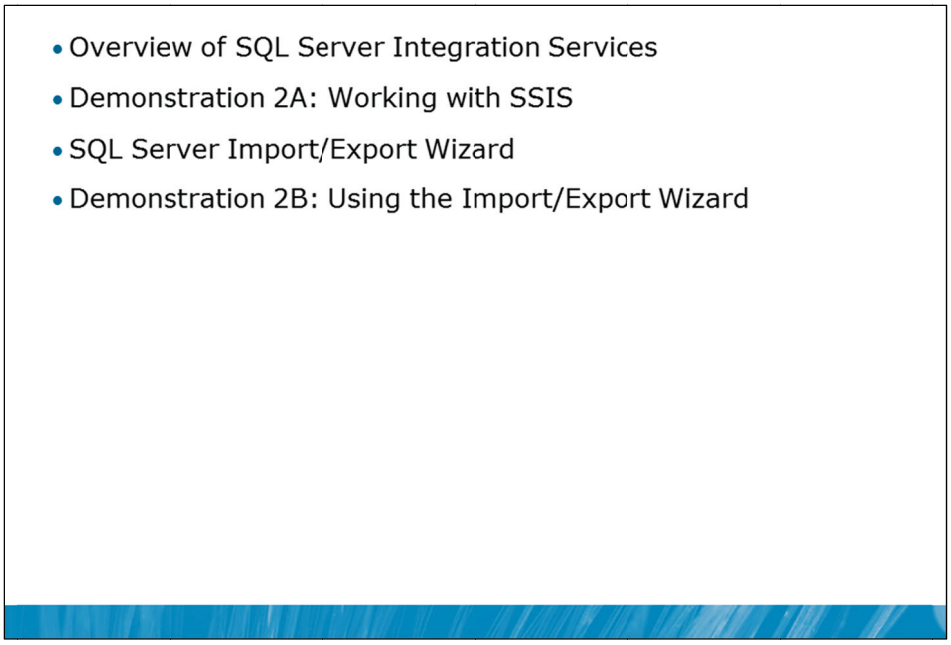
Demonstration 1A: Disabling & Enabling Constraints



Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 2
Importing & Exporting Table Data



SQL Server Integration Services (SSIS) is a powerful ETL tool that can be used in conjunction with SQL Server. SSIS is capable of performing complex transformations on data from one or many sources and loading that data into one or many destinations.

While SSIS is a very capable and complex tool, configuring SSIS for simple import and export processes has been made much easier because SQL Server also provides the Import/Export Wizard. This wizard presents simple configuration dialogs to users and creates an SSIS package based on the selections that the user has made.

Objectives

After completing this lesson, you will be able to:

- Describe SQL Server Integration Services.
- Use SQL Server Import/Export wizard.

Overview of SQL Server Integration Services

Key Points

SQL Server Integration Services (SSIS) is an ETL tool that is provided with SQL Server, in Standard and higher editions. SSIS allows the definition of complex data flows and transformations. The main purpose of SSIS is to create reusable and easily deployable packages that perform data transfers.

Packages

The result of building a set of tasks and transformations in SSIS is referred to as a package. Packages are also the unit of deployment for SSIS and contain a number of objects: Data Sources, Data Destinations, Control Flow, and Data Flows.

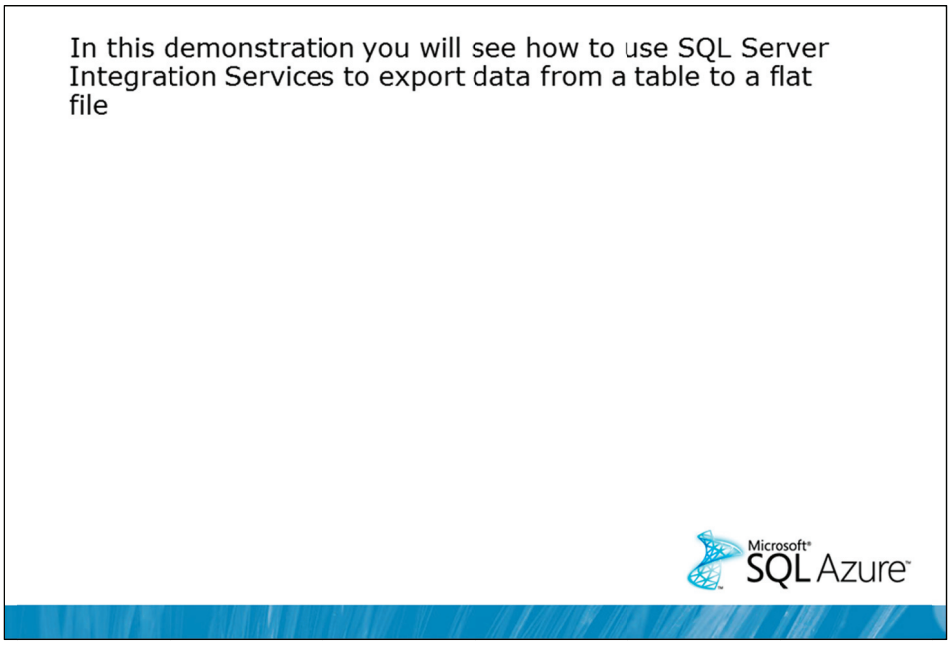
Each package has a single Control Flow that contains a set of tasks that need to be executed. The workflow that needs to be followed when executing the tasks is defined by a set of precedence constraints. If data needs to be moved within an SSIS package, a Data Flow task is added to the set of tasks in the Control Flow. Each Data Flow task is individually configured to specify where data should come from, how it should be transformed, and where it should be sent to.

One goal of SSIS is to perform all data transformation steps of the ETL process in a single operation without the need to stage data before transforming it.

Packages are built using SQL Server Data Tools (SSDT). The SQL Server Import/Export Wizard also creates SSIS packages without the need for users to work with SSDT.

Question: When it will useful to use SSIS instead of other data transfer options?

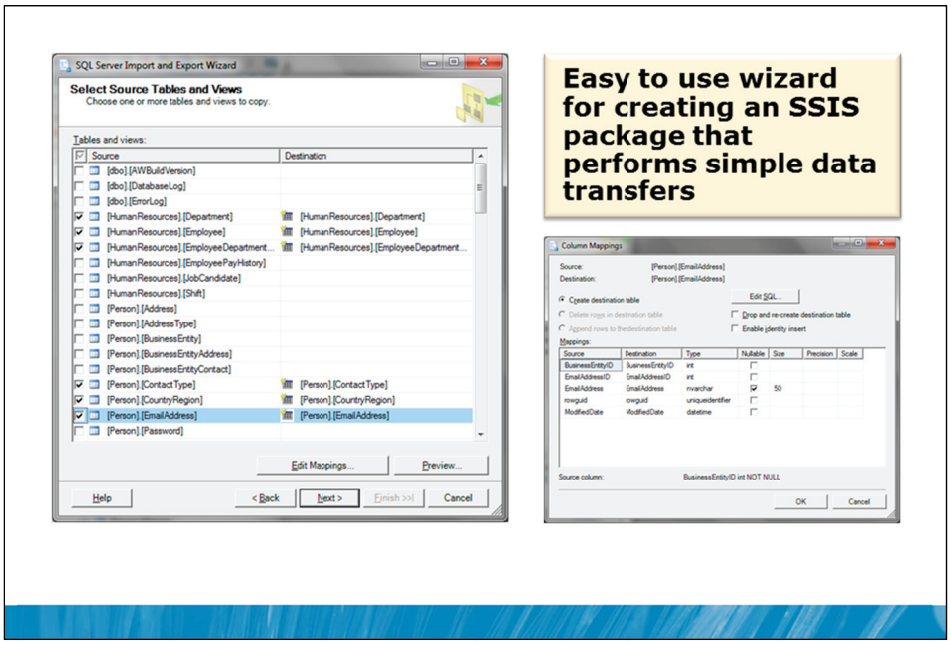
Demonstration 2A: Working with SSIS



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, open the **10775A_08_PRJ SQL** Server script project within SQL Server Management Studio.
   - Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **21 – Demonstration 2A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

SQL Server Import/Export Wizard



Key Points

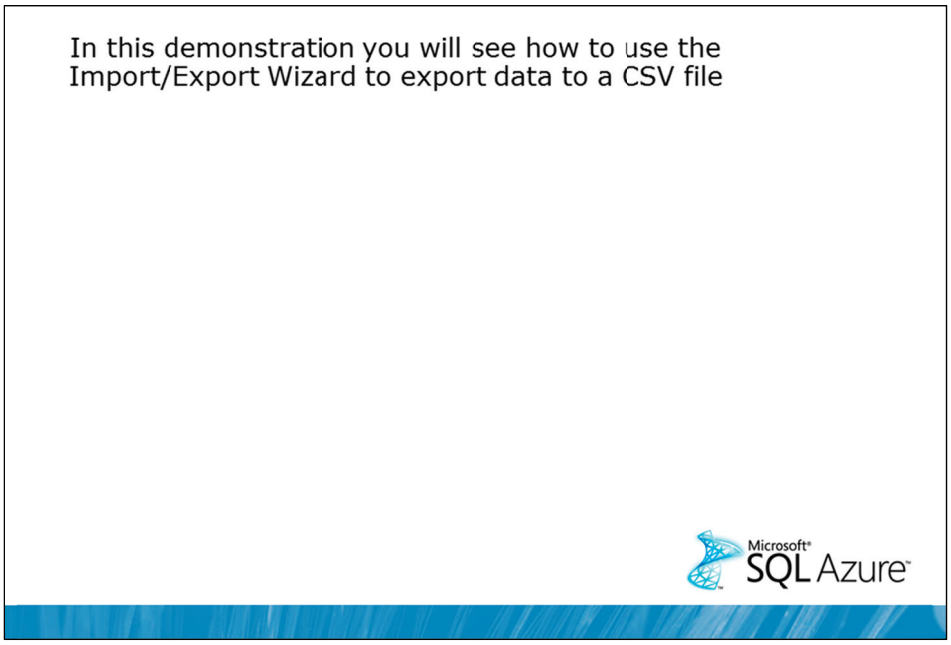
The SQL Server Import and Export Wizard can copy data to and from any data source for which a managed .NET Framework data provider or a native OLE-DB provider is available.

The wizard has some limitations, but can be used with SQL Server, flat files, Microsoft Office Access®, Microsoft Office Excel®, and a wide variety of other database engines.

Although it leverages SQL Server Integration Services, the SQL Server Import and Export Wizard provides minimal transformation capabilities. Except for setting the name, the data type, and the data type properties of columns in new destination tables and files, the SQL Server Import and Export Wizard supports no column-level transformations.

Question: If additional transformations are needed above what is provided with the Import/Export Wizard, how could these be created?

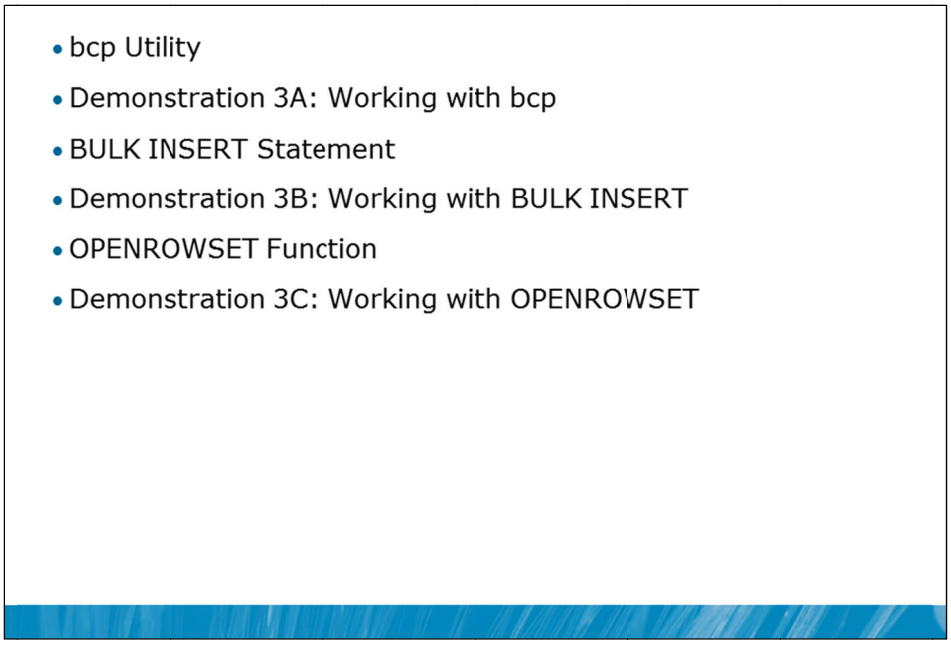
Demonstration 2B: Using the Import/Export Wizard



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **22 – Demonstration 2B.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lesson 3
Inserting Data in Bulk



There are three common options that are used to process the insertion of data in bulk with SQL Server. The bcp utility can be used from the command line to perform both input and output operations.

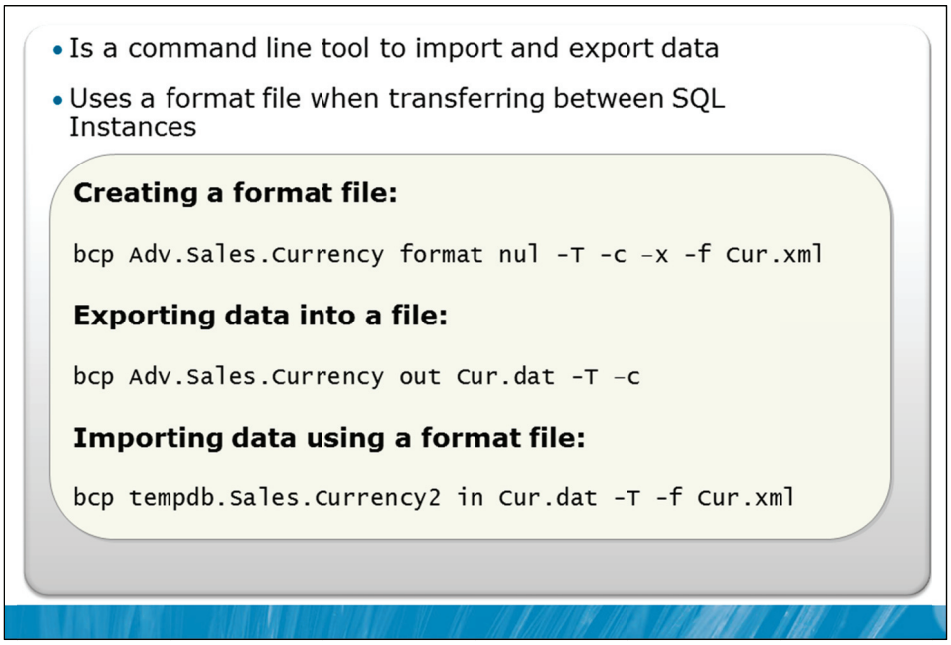
BULK INSERT is a T-SQL command that can be used to import data in bulk and OPENROWSET is a table-valued function that retrieves data from a variety of sources and returns it as a table that can be queried. OPENROWSET can be used in conjunction with an INSERT…SELECT statement to insert data in bulk.

Objectives

After completing this lesson, you will be able to:

- Use the bcp utility.
- Use the BULK INSERT statement.
- Use the OPENROWSET function.

bcp Utility



Key Points

The bcp utility is used to bulk copy data between an instance of Microsoft SQL Server 2012 and a data file in a user-specified format. The bcp utility can be used to import large numbers of new rows into SQL Server tables or to export data out of tables into data files. Except when used with the queryout option, the utility requires no knowledge of T-SQL.

Format Files

To import data into a table, you must either use a format file created for that table or provide that information to bcp interactively. Two types of format files are supported. Current versions of SQL Server use XML-based format files but are able to work with the older text-based format files.

You can use the bcp utility to create a format file that can then be consumed by the bcp utility. In the first example on the slide, the bcp utility is being used to create a format file, based on the column layout of the Adv.Sales.Currency table.

The main parameters that have been specified have the following meanings:

Parameter	Description
format	Create a format file
-T	Integrated security is used to connect to the server
-c	Character data type is used for the export. Character data type provides the highest compatibility between different types of system. An alternative option –n would use the SQL Server native format, which is a more compact format but which can only be used for import/export to other SQL Server systems.
-f	The name of the format file
-x	The format file should be created as XML file

In this example, bcp would connect to the default instance on the local server. If it is necessary to connect to another instance or another server, the –S parameter can be used to supply a server name or a server name and an instance name.

Exporting Data

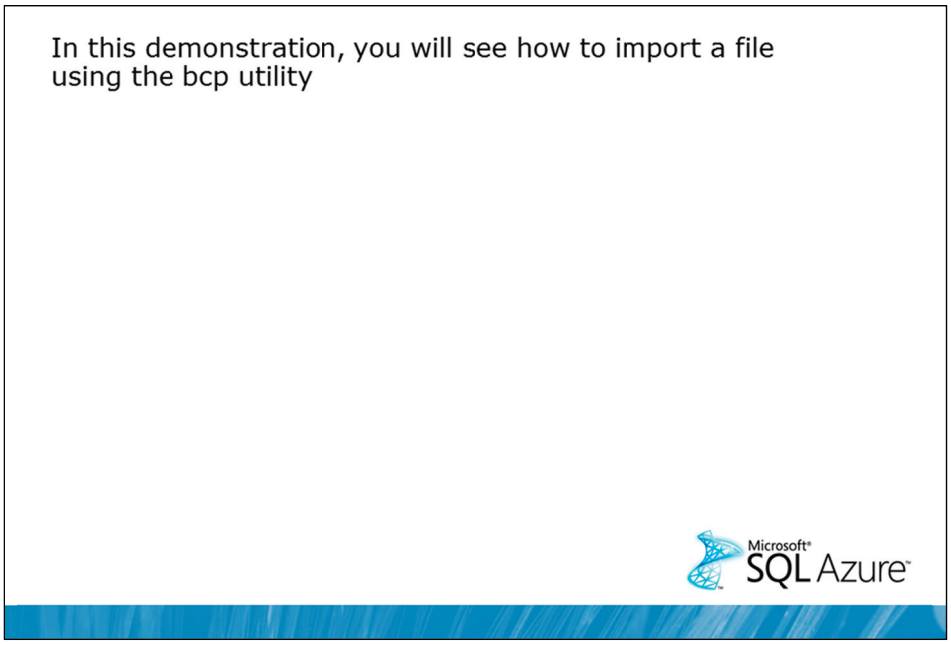
In the second example on the slide, bcp is being used to export the current contents of the Adv.Sales.Currency table to the file Cur.dat. The one parameter that is different to the previous example is the "out" parameter that is used to specify the output file name.

Importing Data

In the third example on the slide, bcp is being used to import the contents of a file Cur.dat into the tempdb.Sales.Currency2 table. Two further uses of parameters are of interest in this example. The "in" parameter is being used to name the file that should be read. The parameter "-f" is being used to specify the format file Cur.xml so that SQL Server understands the format of the input file.

Question: How could you improve the import speed of a bcp operation?

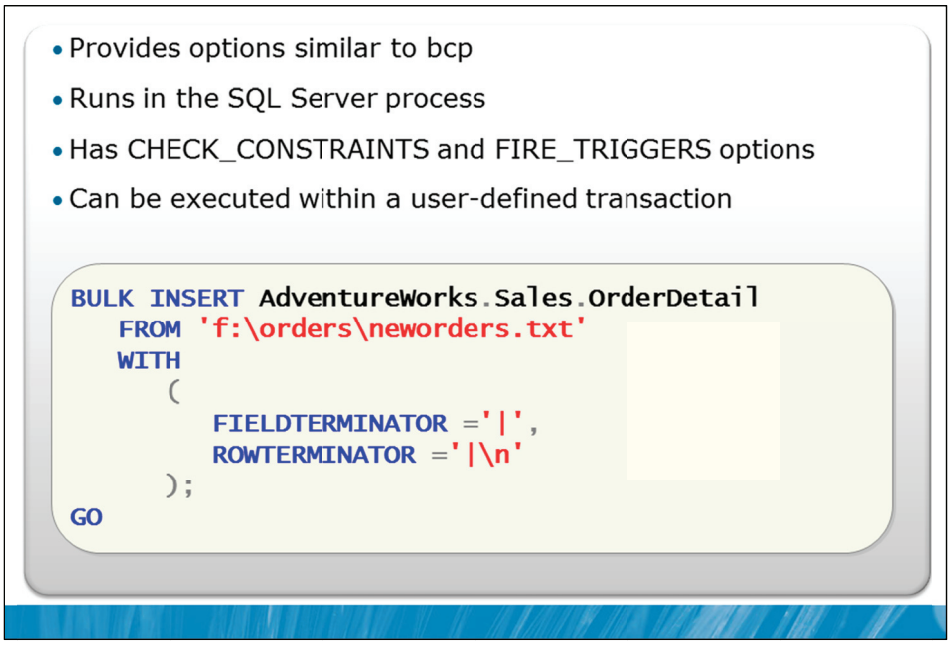
Demonstration 3A: Working with bcp



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **31 – Demonstration 3A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

BULK INSERT Statement



Key Points

BULK INSERT loads data from a data file into a table. This functionality is similar to that provided by the "in" option of the bcp command; however, the data file is read by the SQL Server process, not by an external utility. The BULK INSERT statement executes within a T-SQL batch. Because the data files are opened by a SQL Server process, data is not copied between client process and SQL Server processes. By comparison, the bcp utility runs in a separate process which produces a higher load on the server when run on the same system.

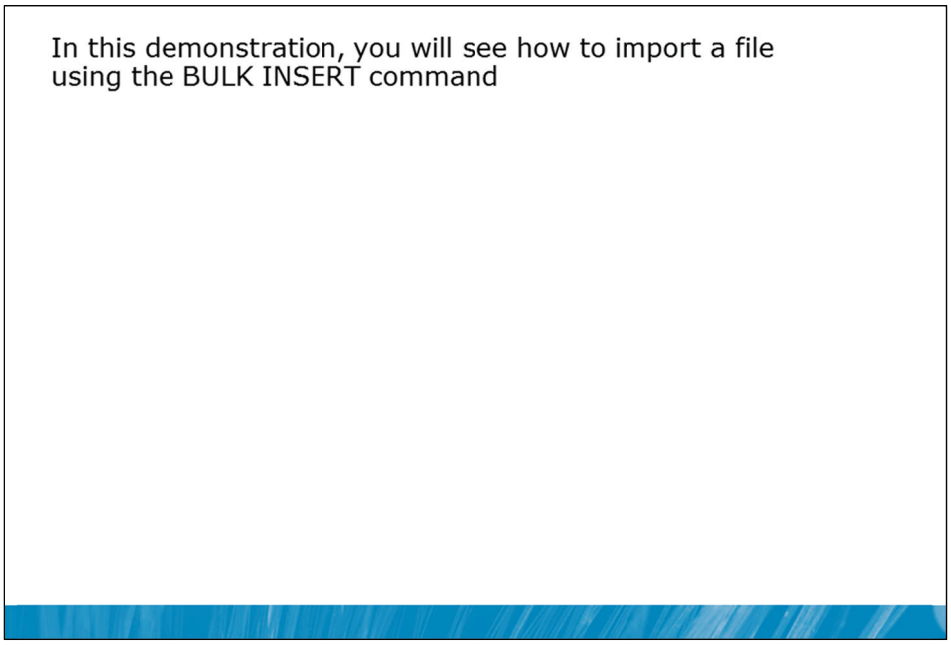
Constraints and Triggers

The BULK INSERT statement offers the CHECK_CONSTRAINTS and FIRE_TRIGGERS options that can be used to tell SQL Server to check constraints and triggers. Unlike the bcp utility, the default operation of the BULK INSERT statement is to not check CHECK and FOREIGN KEY constraints, or to fire triggers on the target table during import operations.

Also unlike bcp, the BULK INSERT can be executed from within a user-defined transaction, which gives the ability to group BULK INSERT with other operations in a single transaction. Care must be taken however, to ensure that the size of the data batches that are imported within a single transaction are not excessive, or significant log file growth might occur, even when the database is in simple recovery model.

Question: How does the BULK INSERT statement differ from bcp?

Demonstration 3B: Working with BULK INSERT

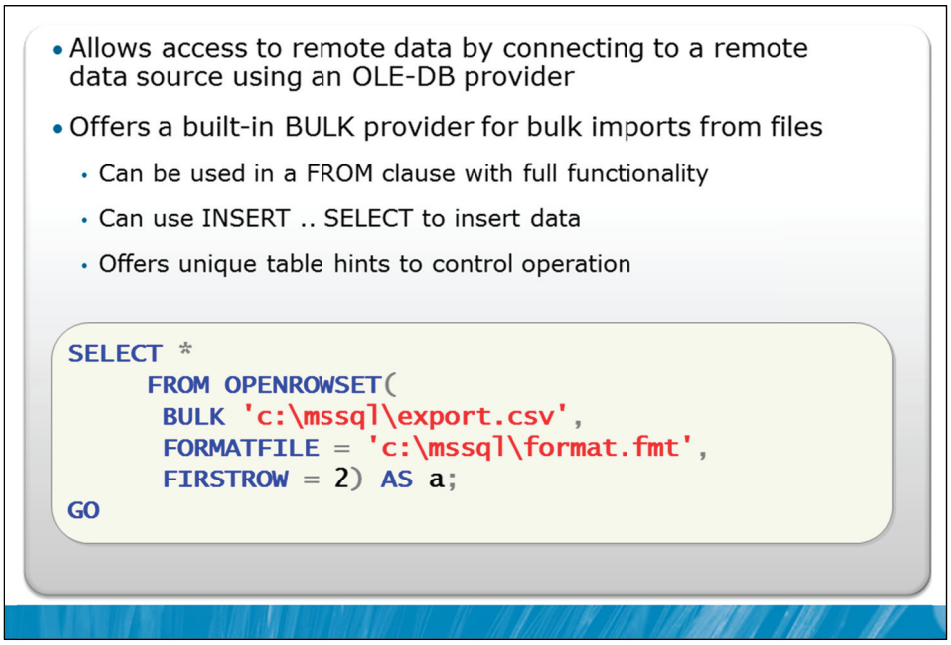


Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **32 – Demonstration 3B.sql** script file.
3. Follow the instructions contained within the comments of the script file.

   Question: Why does the first message show 199 and messages after that show 200?

OPENROWSET Function



Key Points

OPENROWSET can be used to access data using an OLE-DB provider. For OLE-DB providers to be usable in OPENROWSET, a system configuration option "Ad Hoc Distributed Queries" must be enabled and a registry entry for the OLE-DB provider called DisallowAdhocAccess must be explicitly set to 0. This registry key is typically located here: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSDASQL.

When these options are not set, the default behavior does not allow for the ad hoc access that is required by the OPENROWSET function when working with external OLE-DB providers.

BULK Provider

Since SQL Server 2005, a special OLE-DB provider called BULK has been provided which allows the specification of operating system files that will be returned. The same format files that are used with bcp and BULK INSERT can be used with this provider.

In addition to the import of typical data rows, the BULK provider offers three special options that allow the BULK provider to read entire file contents into a single column of a table. These special options are:

Option	Description
SINGLE_CLOB	Reads an entire single-byte character-based file as a single value of data type varchar(max).
SINGLE_NCLOB	Reads an entire double-byte character-based file as a single value of data type nvarchar(max).
SINGLE_BLOB	Reads an entire binary file as a single value of data type varbinary(max).

For example, consider the following code sample:

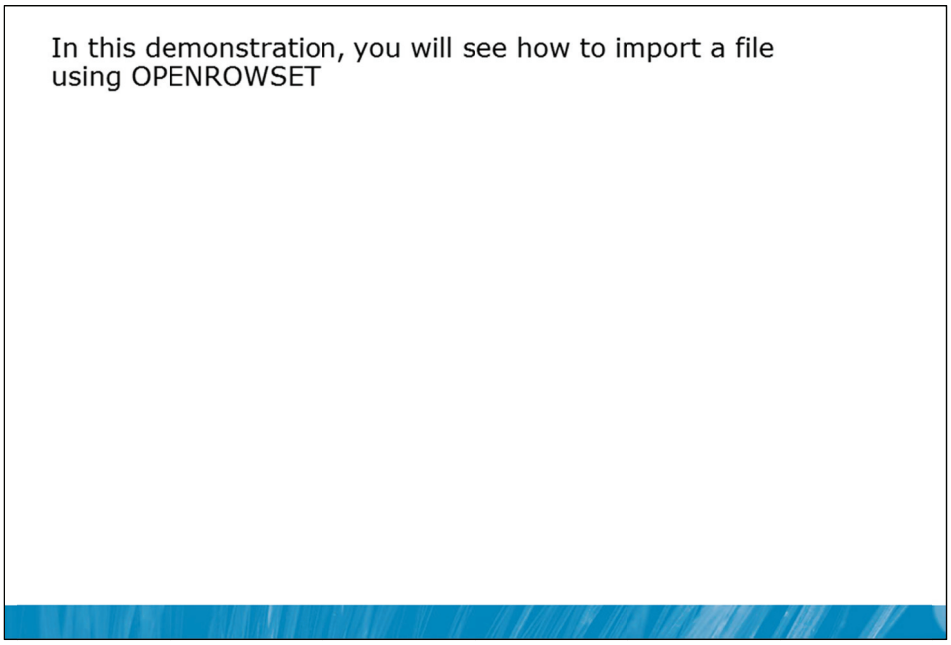
```
INSERT INTO Sales.Documents(FileName, FileType, Document)
SELECT 'JanuarySales.txt' AS FileName,
       '.txt' AS FileType,
       *
FROM OPENROWSET(BULK N'K:\JanuarySales.txt', SINGLE_BLOB) AS Document;
```

In this code sample, the file K:\JanuarySales.txt is being inserted into the Document column of the Sales.Documents table, along with its filename and filetype.

Two key advantages of OPENROWSET compared to bcp are that it can be used in a query with a WHERE clause (to filter the rows that are loaded), and that it can be used in a SELECT statement that is not necessarily associated with an INSERT statement.

Question: When will it make sense to use OPENROWSET instead of bcp or BULK INSERT?

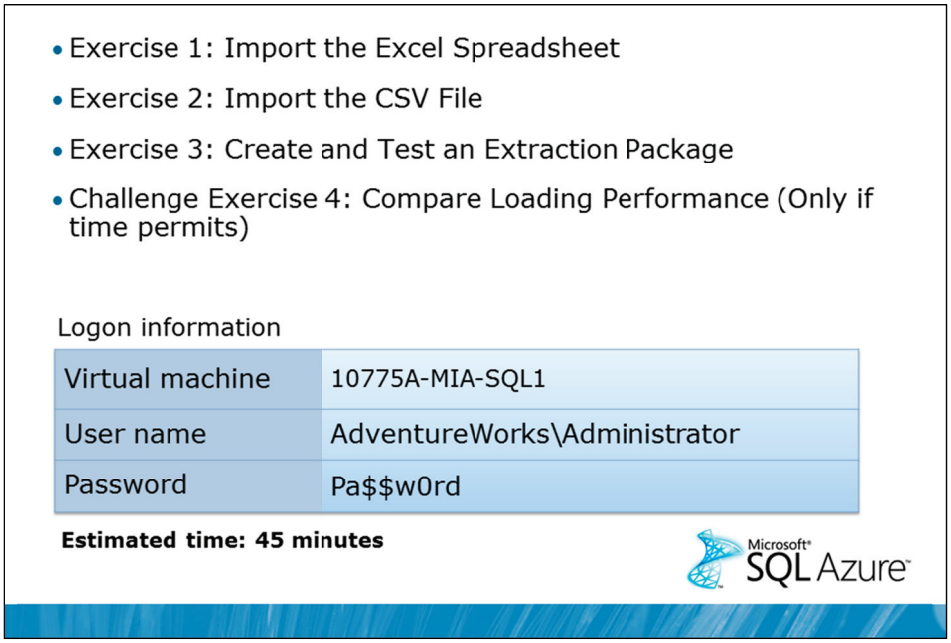
Demonstration 3C: Working with OPENROWSET



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **33 – Demonstration 3C.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lab 8: Importing and Exporting Data



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

Proseware regularly receives updates of currencies and exchange rates from an external provider. One of these files is provided as an Excel spreadsheet, the other file is provided as a comma-delimited text file. You need to import both these files into tables that will be used by the Direct Marketing team within Proseware.

Periodically the Marketing team requires a list of prospects that have not been contacted within the last month. You need to create and test a package that will extract this information to a file for them.

You are concerned about the import performance for the exchange rate file and you are considering disabling constraints and indexes on the exchange rate table during the import process. If you have time, you will test the difference in import performance.

Supporting Documentation

Exercise 1

Item	Description
Output table	DirectMarketing.Currency
Output columns	CurrencyID int not null CurrencyCode nvarchar(3) not null CurrencyName nvarchar(50) not null
Input file	D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ\Currency.xls

Exercise 1: Import the Excel Spreadsheet

Scenario

You need to load a file of currency codes and names from an Excel spreadsheet. In this exercise, you will use the import wizard to perform the data load.

The main task for this exercise is as follows:

1. Import the data using the Import Wizard.

▶ Task 1: Import the data using the Import Wizard

- Import the spreadsheet Currency.xls into a table in the MarketDev database called DirectMarketing.Currency. If the table already exists, delete the table first. Refer to the Supporting Documentation for the file location and output table format.
- Query the DirectMarketing.Currency table to see that the data that was loaded.

Results: After this exercise, you should have imported the DirectMarketing.Currency table.

Exercise 2: Import the CSV File

Scenario

You have also been provided with a comma-delimited file of exchange rates. You need to import these exchange rates into the existing DirectMarketing.ExchangeRate table. The table should be truncated before the data is loaded.

The main task for this exercise is as follows:

1. Import the CSV file.

Note Make sure that you record how long the command takes to execute. Warning: it will take several minutes to complete. Use this time to prepare for the next Exercise.

▶ Task 1: Import the CSV file

- Truncate the DirectMarketing.ExchangeRate table.
- Review the ExchangeRates.xml format file in the Solution Explorer.
- Using BULK INSERT T-SQL command import the ExchangeRates.csv file into the table DirectMarketing.ExchangeRate. The ExchangeRates.csv file can be found in the following location: D:\10775A_Labs\10775A_08_PRJ\10775A_08_PRJ\ExchangeRates.csv.
- Use ExchangeRates.xml as the format file and a batch size of 10,000, and use the option to skip the first row as it contains headings.

Results: After this exercise, you should have imported the ExchangeRate table using T-SQL BULK INSERT statement.

Exercise 3: Create and Test an Extraction Package

Scenario

Periodically the Marketing team requires a list of prospects that have not been contacted within the last month. You need to create and test a package that will extract this information to a file for them.

The main task for this exercise is as follows:

1. Create and test an extraction package.

▶ Task 1: Create and test an extraction package

- Using the Export Wizard, export the Marketing.Prospect table to a text file in the following location: D:\MKTG\ProspectsToContact.csv. Column Names should be included in the first row. The extraction query should be as shown in the snippet below:

```
SELECT ProspectID, FirstName, LastName, CellPhoneNumber,
WorkPhoneNumber,EmailAddress, LatestContact
FROM Marketing.Prospect
WHERE LatestContact < DATEADD(MONTH,-1,SYSDATETIME())
OR LatestContact IS NULL
ORDER BY ProspectID;
```

Note Save the SSIS package that is created by the Export Wizard to SQL Server in the package root location.

Results: After this exercise, you should have created and tested an extraction package.

Challenge Exercise 4: Compare Loading Performance (Only if time permits)

Scenario

You are concerned about the import performance for the exchange rate file and you are considering disabling constraints and indexes on the exchange rate table during the import process. If you have time, you will test the difference in import performance.

The main task for this exercise is as follows:

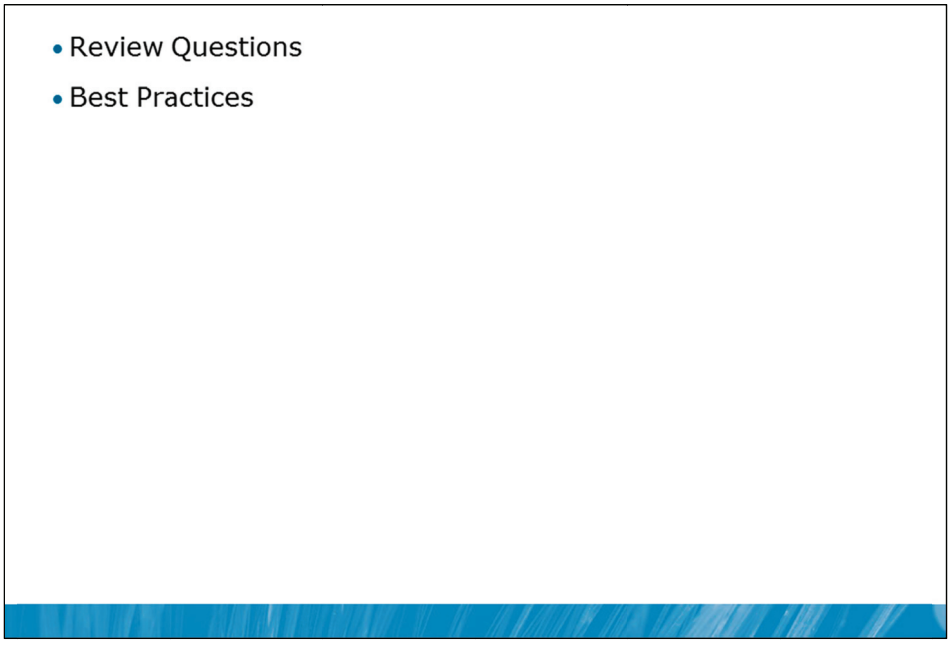
1. Re-execute load with indexes disabled.

▶ Task 1: Re-execute load with indexes disabled

- Alter your script from Exercise 2 to disable any non-clustered indexes on the DirectMarketing.ExchangeRate table before loading the data and to rebuild the indexes after the load completes.
- Execute your modified script and compare the duration to the value recorded in Exercise 2.

Results: After this exercise, you should have compared the load performance with indexes disabled.

Module Review and Takeaways



Review Questions

1. When would you use SSIS instead of other data transfer utilities?
2. Why are minimally logged operations faster than fully logged operations?

Best Practices related to a particular technology area in this module

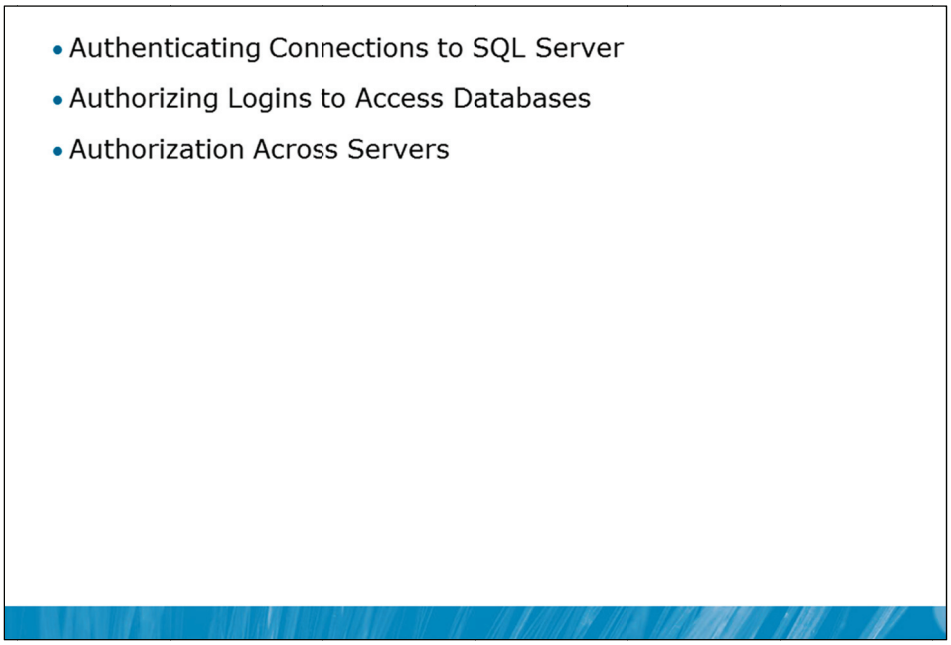
1. Choose the right tool for bulk-imports.
2. Use SSIS for complex transformations.
3. Use bcp or BULK INSERT for fast imports and exports.
4. Use OPENROWSET when data needs to be filtered before it gets inserted.
5. Try to achieve minimal logging to speed up data import.

Module 9

Authenticating and Authorizing Users

Contents:

Module Overview



Securing Microsoft® SQL Server® can be viewed as a series of steps, involving four areas: the platform, authentication, objects (including data), and applications that access the system. Well-planned security is important for protecting your organization's sensitive information.

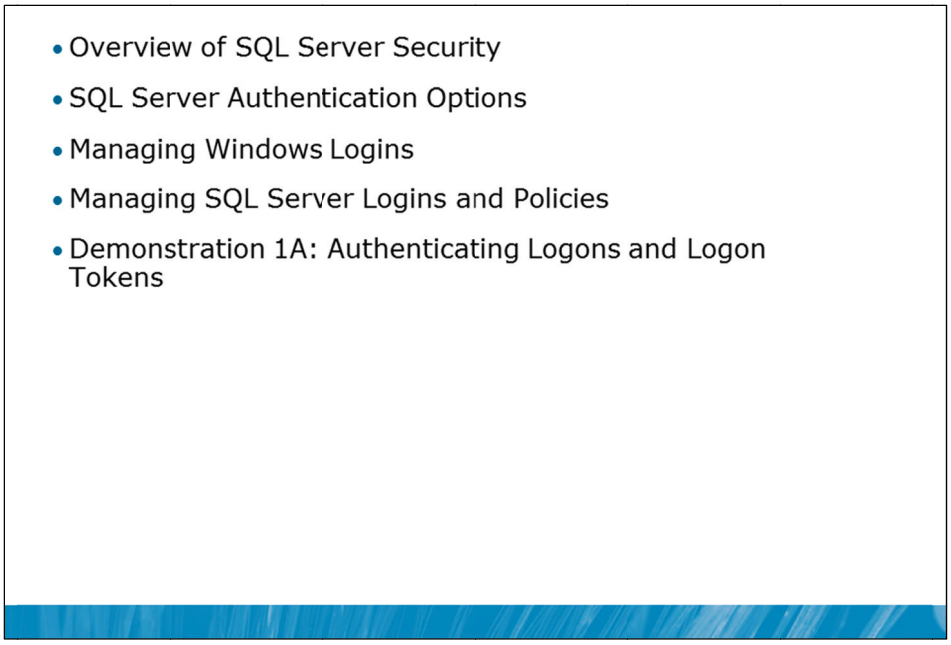
In this module you will be introduced to securing SQL Server. You will learn how SQL Server authenticates users and how it authorizes the users to access databases. Not all resources reside on a single server. In this module, you will also see how distributed authorization is configured; that is, where more than one SQL Server system is involved.

Objectives

After completing this lesson, you will be able to:

- Describe how SQL Server authenticates connections.
- Describe how logins are authorized to access databases.
- Explain the requirements for authorization across servers.

Lesson 1
Authenticating Connections to SQL Server



The first layer of security within SQL Server is authentication of users. Before any other security settings can be examined, SQL Server needs to verify the identity of the user.

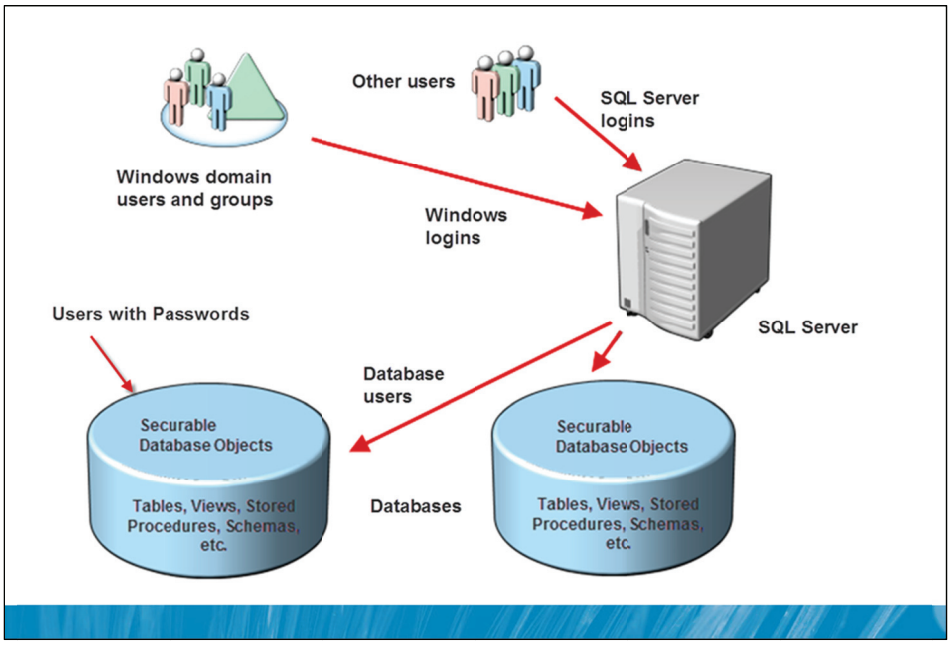
Authentication is the process of identifying users and of verifying their identity. In this lesson, you will learn about the options available for authenticating users, about the available types of logins to the server and about how policies are managed in relation to SQL Server logins.

Objectives

After completing this lesson, you will be able to:

- Describe SQL Server security.
- Implement SQL Server authentication options.
- Provide access to SQL Server for Windows® users and groups.
- Provide access to SQL Server for other users.

Overview of SQL Server Security



Key Points

A user can gain permission to connect to SQL Server in one of four ways:

- They can be a user that SQL Server manages directly. These users become known as SQL Server logins.
- They can be a user that Windows has authenticated and SQL Server has been configured to allow the user to connect. These users become known as Windows logins.
- They can be a Windows user that is a member of a Windows group. SQL Server has been configured to allow members of the group to connect. These users also become known as Windows logins.
- They can be a database user that is assigned a password directly where the user is not associated with a login.

Note It is also possible for logins to be created from certificates and keys but the creation of these is an advanced topic, outside the scope of this course.

Logins vs. Database Users

The distinction between logins and database users is important. The term "Login" is applied to a principal (or process) that has been granted access to SQL Server via one of the three methods mentioned above.

Note Having access to the server does not indicate (in itself) that a login has any access to user databases on the server.

Logins can be granted permission to access one or more databases. A mapping can exist between a login (either Windows Login or a SQL Server Login) and a "Database User" in a particular database. Database users often have the same name as the logins that they are mapped to but the names can be different. A login can even be mapped to different user names in each database.

Note Keeping login and database user names the same is considered a best practice.

Permissions

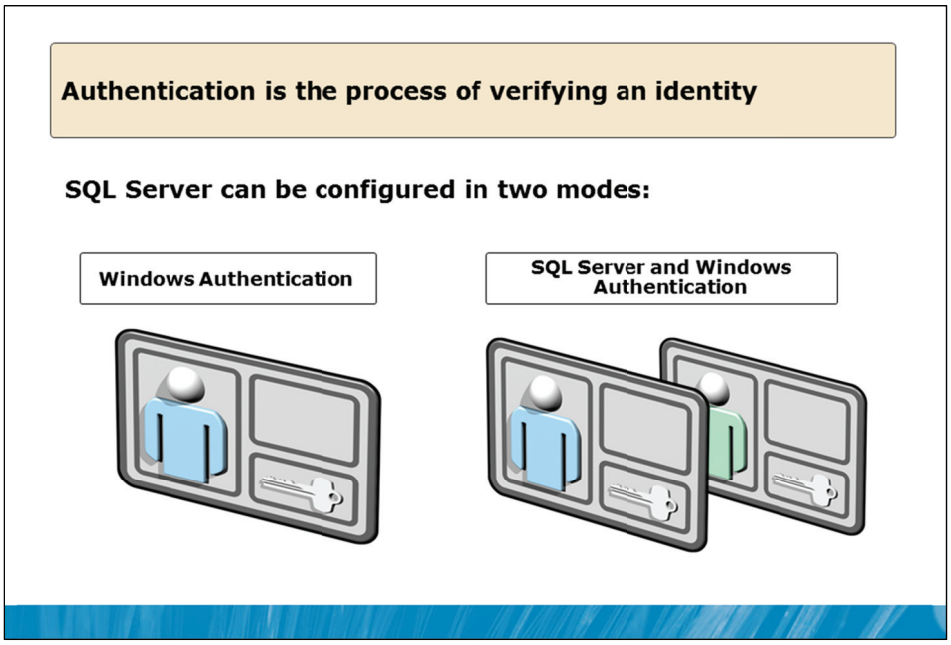
Even once a login is granted access to a database by the creation of a database user, the database user will need to be granted permissions within the database before they can access securable objects such as tables, views, functions, and stored procedures.

Note Some logins have permissions across all databases because they are added to roles at the server level, such as the sysadmin role, but these are exceptions to the general situation.

There are two ways that database users can be granted permission to access securable objects. The database users can be granted permissions on the objects. Alternatively, roles can be created within the database. Roles are granted permissions on the securable objects and the database users are added as members of the roles. Database users inherit all permissions that are associated with any roles that they are members of, along with any permissions that have been directly assigned to them.

Question: Apart from Windows users, what other types of users might want to connect to SQL Server?

SQL Server Authentication Options



Key Points

Authentication is the process of verifying that an identity is valid. If the identity is a user, is the user who they claim to be? There are two basic ways this can occur.

- SQL Server trusts the Windows operating system to verify the identity of a Windows login. Windows might use different methods such as password-checking, biometric checks (for example, fingerprint scanning) or certificates to validate a user's identity. It might even use a combination of such methods. In SQL Server, Windows logins are often referred to as "trusted logins".
- SQL Server can directly verify the identity of a SQL Server login by checking that they know the password associated with that login.

  Question: If you call your bank by phone, how do they verify your identity before speaking to you about your account details?

Server Configuration

SQL Server can be configured in two modes

- Windows Authentication mode.
- SQL Server and Windows Authentication mode.

Windows Authentication mode was formerly known as Integrated Mode. In Windows Authentication mode, only users that the Windows operating system has authenticated are permitted to connect to the server.

In SQL Server and Windows Authentication mode, both users that have been authenticated by the Windows operating system and users that SQL Server has directly authenticated are permitted to connect to the server. This mode is often called Mixed Mode.

The choice between these two modes is made at the SQL Server instance level during the installation of SQL Server and can easily be changed. The instance needs to be restarted after changing the configuration.

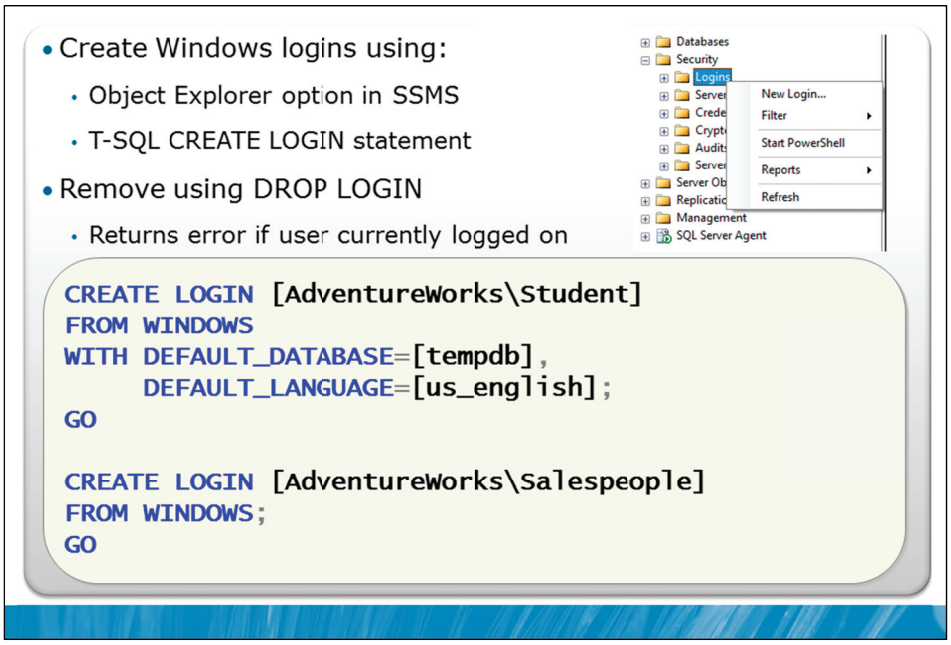
Note The configuration can be made using the GUI tooling in SSMS but it is in fact only a single registry key that is being changed. This registry key could be configured via a group policy within Windows.

If SQL Server and Windows Authentication mode is enabled, a SQL Server login called "sa" is then active. It is important to set an appropriate (and complex) password for the "sa" login and record it somewhere. When SQL Server is installed as Windows Authentication Only, the "sa" account is disabled by default. If SQL Server is installed in Mixed Mode, then "sa" is enabled by default. Changing the server's authentication mode does not change the enabled/disabled state of the "sa" login.

Protocols for Authentication

Windows authentication is typically performed via the Kerberos protocol. The Kerberos protocol is supported with SQL Server over the TCP/IP, Named Pipes, and Shared Memory network protocols. (Support for Kerberos over Named Pipes and Shared Memory was introduced in SQL Server 2008. SQL Server 2005 supported Kerberos on the TCP/IP network protocol).

Managing Windows Logins



Key Points

Logins can be created directly using T-SQL code or via the GUI provided in SSMS. While the GUI option can be used, creating logins is a common operation and if many logins need to be created, you will find it much faster, more repeatable, and accurate to use a script.

CREATE LOGIN

Logins can be created in two ways. To create a login in SSMS, expand the Security node at the server instance level and then right-click Logins to choose New Login.

Alternatively, logins can be created using the CREATE LOGIN statement as shown in the slide example.

Note Windows user and group names must be enclosed within square brackets as shown in the slide example, primarily because they contain a backslash character.

Windows logins can be created for individual users or for Windows groups. The first example in the slide shows the creation of a login for an individual user. Note that a default database and a default language are assigned.

When a default database is not assigned directly, the master database will be assigned by SQL Server. When a default language is not assigned directly, the default language of the server instance will be assigned by SQL Server.

Windows Groups

The second example in the slide is for a Windows group. Members of the group AdventureWorks\Salespeople will be permitted to connect to the server without the need for individual logins.

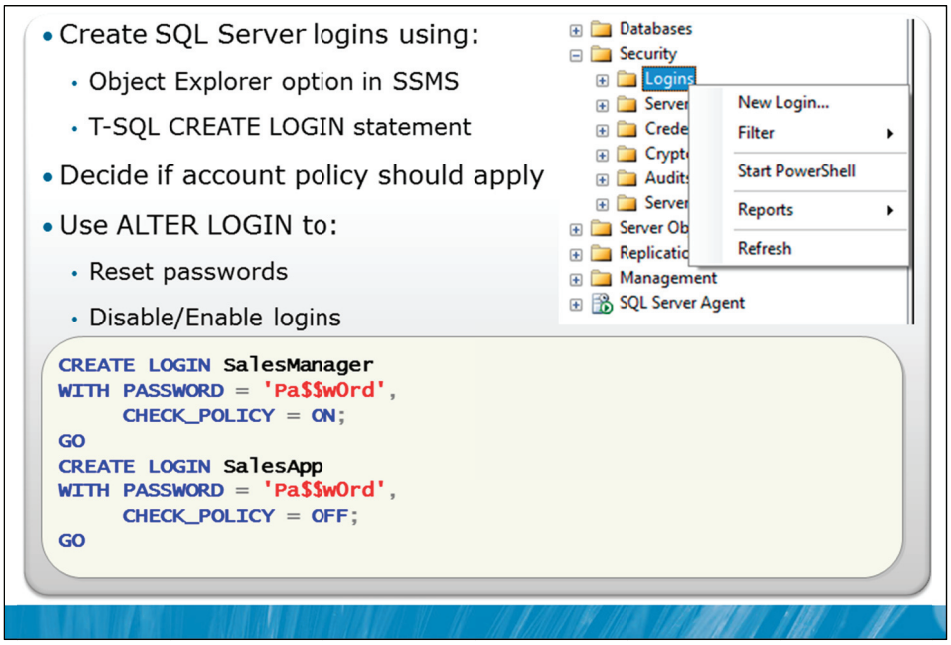
Note Windows users and groups can both refer to local or domain users and groups.

Removing Logins

Logins are removed with the DROP LOGIN statement. Note that you cannot drop a login if the user is currently logged in. Should you need to do this, you will need to locate their session ID in the SSMS Activity Monitor (by viewing the list of processes) and kill the session first. Avoid deleting logins where the database users associated with the logins own objects within databases. (Object ownership is discussed in Module 11).

Question: Why would you create logins based on groups in preference to logins based on users?

Managing SQL Server Logins and Policies



Key Points

SQL Server logins are created for individual identities and are created using the same tools as Windows logins.

Security of SQL Server Logins

For many years, the use of SQL Server logins was considered a poor security practice. There were several reasons for this, including:

- Unencrypted authentication.
- Lack of account policy.

The concern with encryption was that during the authentication phase, the traffic on the network was not encrypted. This means that an attacker that "sniffed" network packets would have been able to detect enough information to access the system via the SQL Server login.

The concern with account policy was that even though the SQL Server system might have been part of a domain that implemented detailed account policy, the policy did not apply to SQL Server logins. Both of these issues were addressed in SQL Server 2005.

Encrypted Authentication

The upgraded SQL Server Native Access Client (SNAC) that was provided with SQL Server 2005 was enhanced to provide encrypted authentication. If SQL Server did not have a Secure Sockets Layer (SSL) certificate installed by an administrator, SQL Server would generate and self-sign a certificate to be used for encrypting authentication traffic.

Note Encrypted authentication only applied to clients running the SQL Server 2005 version of SNAC or later. If an earlier client that did not understand encrypted authentication tried to connect, by default SQL Server would allow this. There is a configurable property for each supported protocol in SQL Server Configuration Manager that can be used to disallow unencrypted authentication from down-level clients if this is a concern.

Account Policy

The implementation of account policy is based on the ability to read policy details from the operating system. Windows Server 2003 and later operating systems introduced an API that allowed applications to read details of account policy.

SQL Server is supported on some operating systems (such as Windows XP) that do not support this API. For those operating systems, account policy is replaced by a basic set of password complexity rules.

The full application of account policy is not always desirable. For example, some applications use fixed logins to connect to the server. Often, these applications do not support regular changing of login passwords. In these cases, it is common to disable policy checking for those logins.

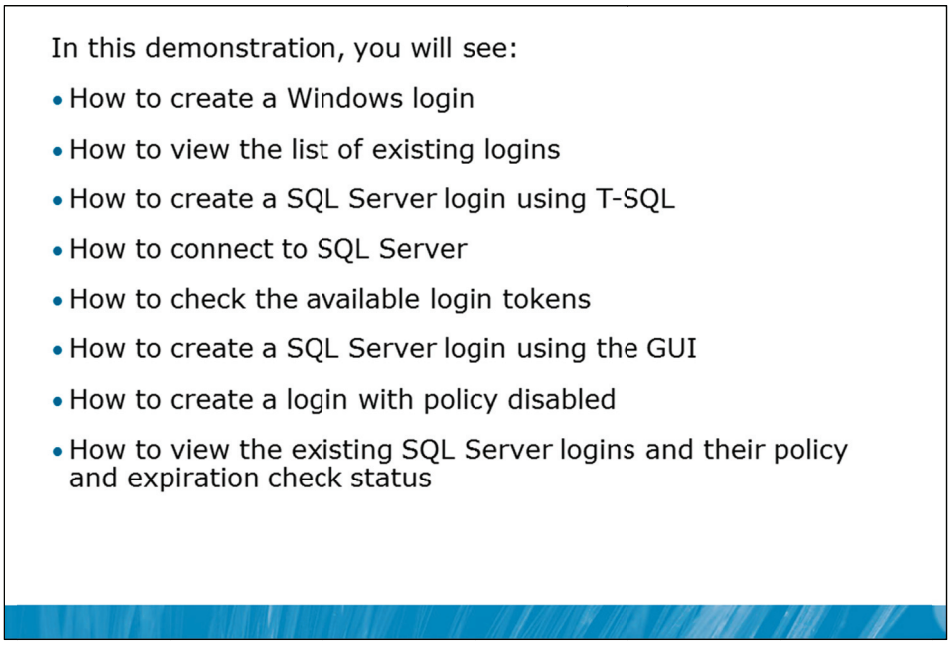
Password Changing and Login Expiry

Passwords can be reset using the GUI in SSMS or via the ALTER LOGIN statement. If logins are not being used for a period of time, they can be disabled and later re-enabled. If there is any chance that a login will be needed again in the future, it would be better to be disabled rather than dropped. Disabling the login is achieved by executing the ALTER LOGIN statement as shown in the code below:

```
ALTER LOGIN James DISABLE;
```

Question: Can you suggest a type of account policy that Windows provides?

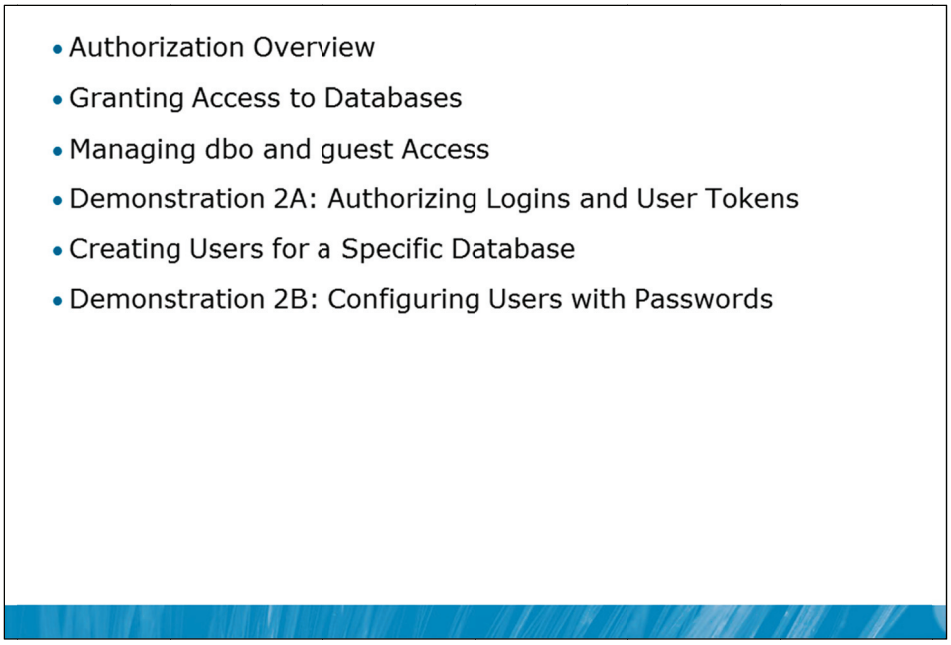
Demonstration 1A: Authenticating Logons and Logon Tokens



Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_09_PRJ\10775A_09_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 2
Authorizing Logins to Access Databases



Once the identity of a login has been verified, it is necessary to provide that login with access to the databases that the login needs to work with. This access is provided by granting the login access to a database and then by providing the login with sufficient permissions within the database to perform the necessary work.

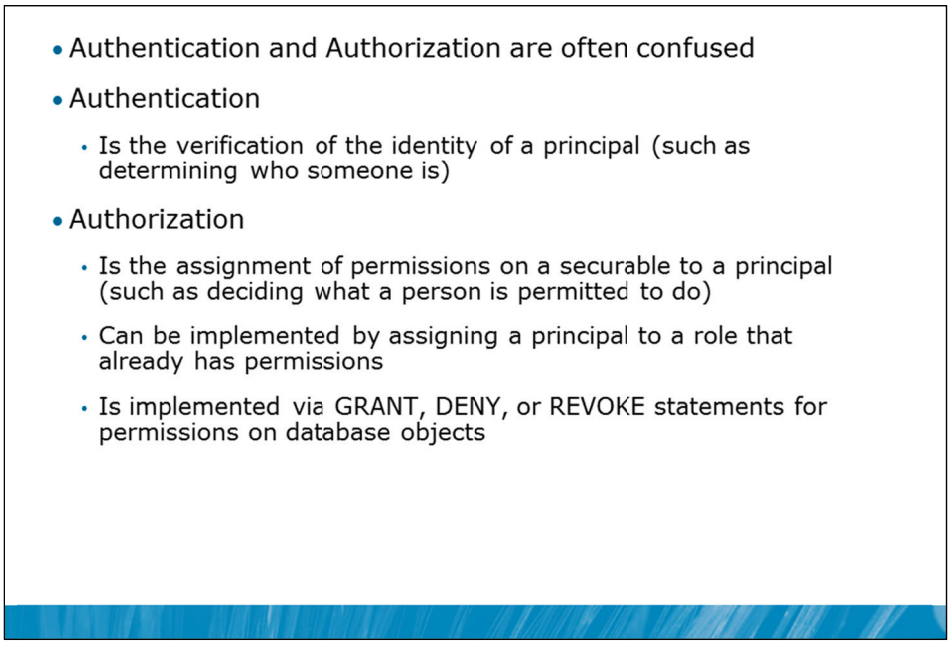
In this lesson, you will see how to grant access to a database to a login and see details of special forms of access to databases.

Objectives

After completing this lesson, you will be able to:

- Describe the SQL Server authorization process.
- Grant access to databases.
- Manage dbo and guest access.
- Configure users with passwords for partially contained databases.

Authorization Overview



Key Points

A very common error when discussing security is to confuse the concepts of authentication and authorization. You can think of authentication as proving who you are and you can think of authorization as determining what you are allowed to do.

Formal Terminology

It is important to be familiar with common security-related formal terminology as shown in the following table:

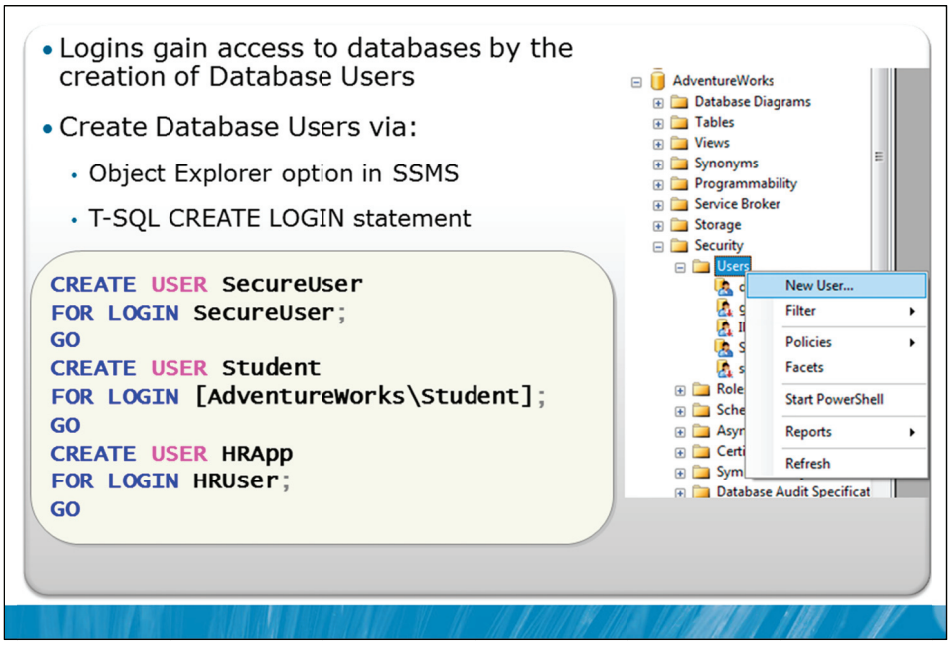
Term	Description
Principal	An entity that can request access to a resource
Securable	A resource that can be secured or an object that you can control access to
Authentication	Ensuring that the principal is who they say they are
Authorization	Providing controlled access to a securable, for a principal. That is, determining the permissions that a principal has on a securable
Role	A container for principals that can be used to assign permissions indirectly

A role is a name that is given to a group of principals. Roles are created to make it easier to grant permissions to many principals that need similar levels of access. Roles in SQL Server are similar to groups within the Windows operating system.

Permissions are controlled by the GRANT, REVOKE, and DENY commands and will be discussed in Module 11.

Question: Would you imagine that a login is a principal or a securable?

Granting Access to Databases



Key Points

Logins are granted access to databases through the creation of database users. A database user is a principal within a database that is mapped to a login at the server.

Database users can be created through the SSMS GUI or via T-SQL commands. To create a new database user via the GUI, expand the relevant database, expand the Security node, right-click the Users node and choose the option for New User.

Note There are special types of database users that are created directly from certificates and are not associated with logins. The creation of these is an advanced topic beyond the scope of this course.

CREATE USER / DROP USER

Once a login exists, it can be linked to a new database user with the CREATE USER statement of T-SQL. The DROP USER statement is used to remove these users.

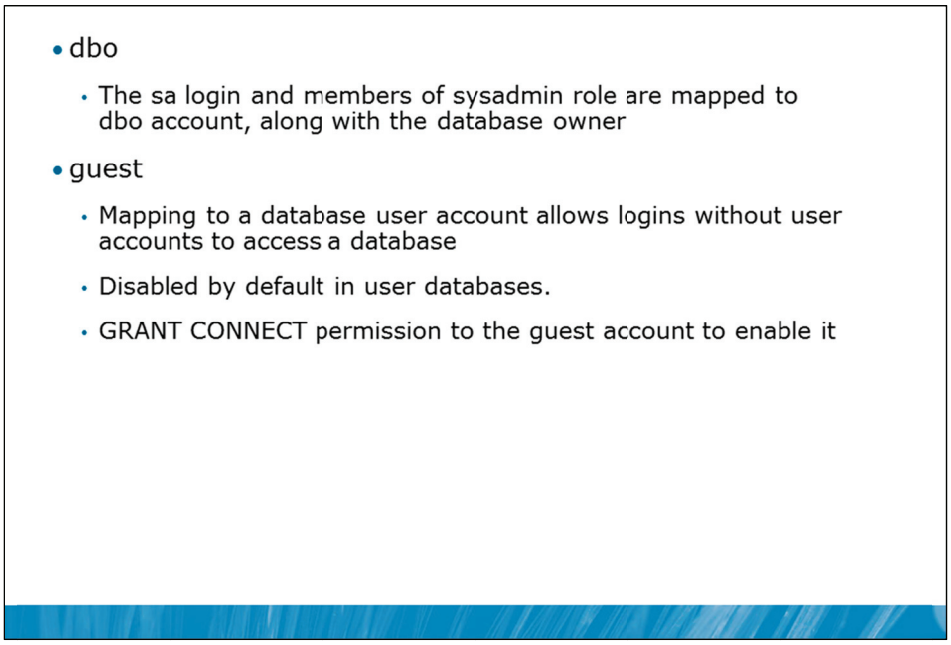
Consider the examples shown on the slide:

- The first example creates a database user named SecureUser for an existing SQL Server login also named SecureUser.
- The second example creates a database user named Student for a Windows login named AdventureWorks\Student.

- The third example creates a database user HRApp for a SQL Server login named HRUser. Note that the database user has been given a different name than the name of the login.

Note The names of Windows logins (either users or groups) must be enclosed in square brackets.

Managing dbo and guest Access



Key Points

Each SQL Server database includes two special database users: dbo and guest.

dbo

dbo is a special user that has implied permissions to perform all activities in the database. Any member of the sysadmin fixed server role (including the "sa" user) who uses a database, is mapped to the special database user called dbo inside each database. The dbo database user cannot be deleted and is always present in every database.

Database Ownership

Like other objects in SQL Server, databases also have owners. The owner of a database is also mapped to the dbo user. The database owner can be modified using the ALTER AUTHORIZATION statement, as shown in the following code:

```
ALTER AUTHORIZATION ON DATABASE::MarketDev
  TO [ADVENTUREWORKS\Administrator];
```

Any object that is created by any member of the sysadmin fixed server role will also automatically have dbo as its owner. Owners of objects have full access to the objects and do not require explicit permissions before they can perform operations on the objects.

guest

The guest user account allows logins that are not mapped to a database user in a particular database to gain access to that database. Login accounts assume the identity of the guest user when the following conditions are met:

- The login has access to SQL Server but does not have access to the database through its own database user mapping.
- The guest account has been enabled.

The guest account can be added to a database to allow anyone with a valid login to access the database. The guest username is automatically a member of the public role. (Roles will be discussed in the next module).

A guest user works as follows:

- SQL Server checks to see whether the login is mapped to with a database user in the database that the login is trying to access. If so, SQL Server grants the login access to the database as the database user.
- SQL Server checks to see whether a guest database user is enabled. If so, the login is granted access to the database as guest. If the guest account does not exist or is not enabled, SQL Server denies access to the database.

The guest user cannot be dropped but you can prevent it from accessing a database by executing the following command:

```
REVOKE CONNECT FROM guest;
```

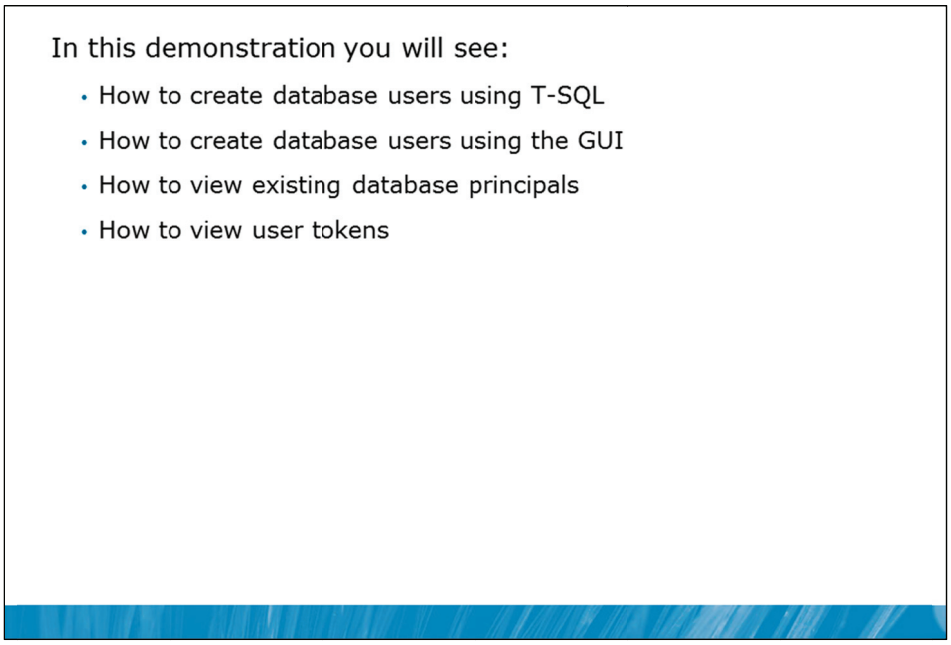
The guest account can be enabled by executing the following command:

```
GRANT CONNECT TO guest;
```

Note The guest user is used to provide access to the master, msdb, and tempdb databases. You should not attempt to revoke guest access to these databases.

Question: What is the guest user useful for?

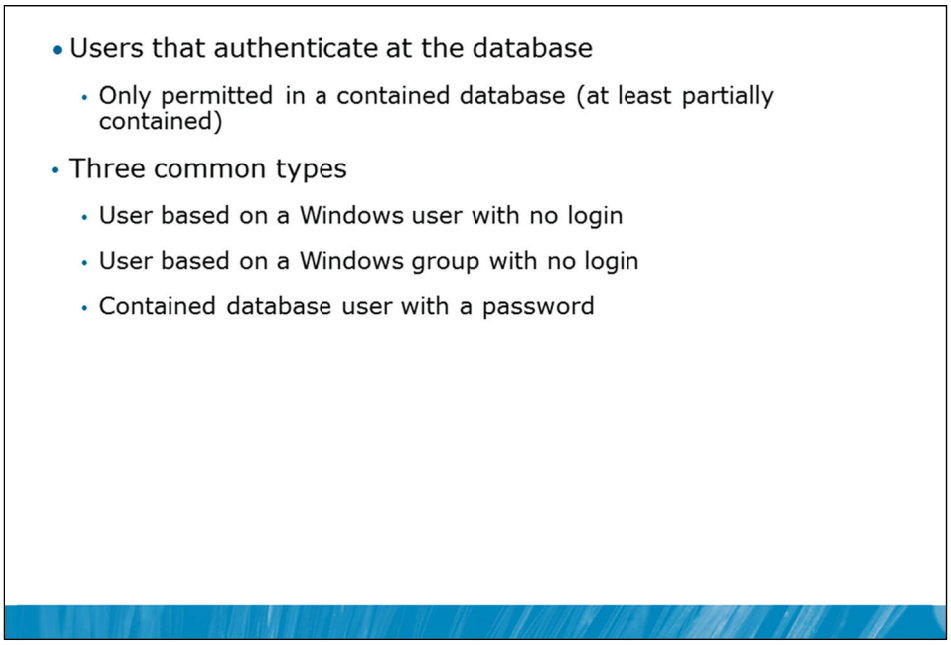
Demonstration 2A: Authorizing Logins and User Tokens



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_09_PRJ\10775A_09_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
   - Open and execute the **11 – Demonstration 1A.sql** script file from within Solution Explorer.
2. Open the **21 – Demonstration 2A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Creating Users for a Specific Database



Key Points

Until SQL Server 2012, for a user to connect to a database, they first needed to have a login created in the master database on the server.

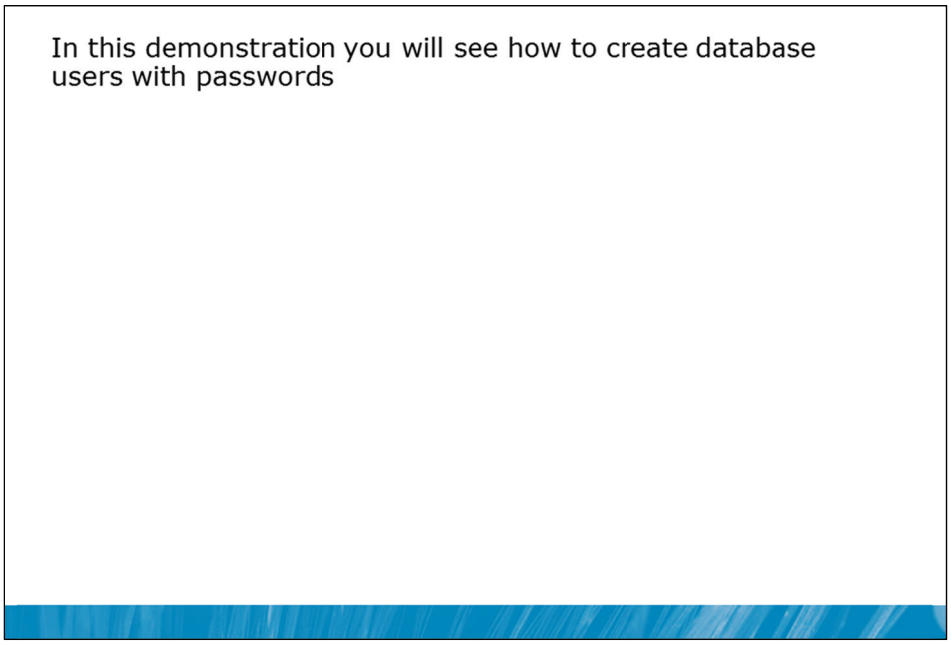
Users can now be created within a database without any entries being made in the master database for the users. A database must be partially contained before it can be used for users without logins. Containment is a property of a database.

There are three types of users that can be authenticated by the database rather than by the server:

- A Windows user can be authenticated at the database level based upon their Windows user credentials.
- Members of a Windows group can be authenticated at the database level.
- SQL Server users can be assigned passwords and authenticated at the database level, without the need for a SQL Server login.

Users that are authenticated by the database cannot access other databases except as guest users.

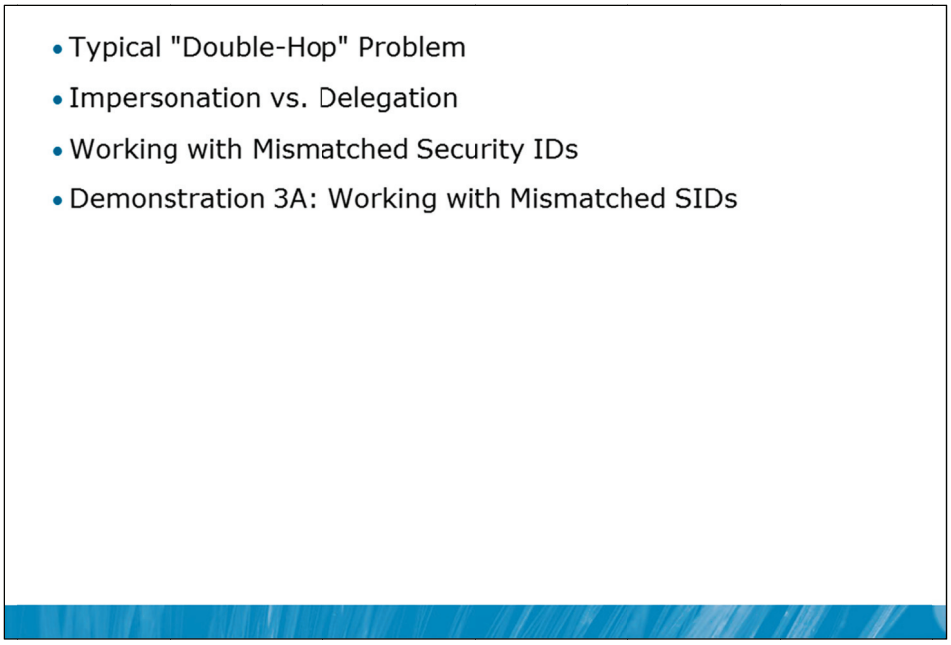
Demonstration 2B: Configuring Users with Passwords



Demonstration Steps

1. If Demonstration 2A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_09_PRJ\10775A_09_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
   - Open and execute the **21 – Demonstration 2A.sql** script file from within Solution Explorer.
2. Open the **22 – Demonstration 2B.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lesson 3
Authorization Across Servers



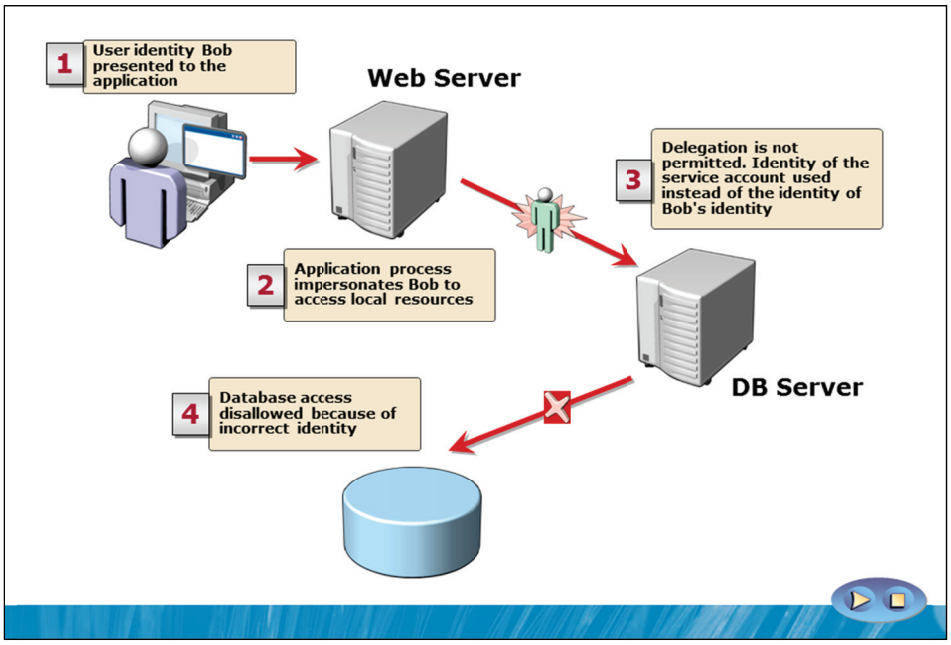
Not all resources reside within a single server. It is common to need to access resources on other servers. There are two common issues that arise in relation to working across servers. One is referred to as the "double-hop" problem and it relates to the difference between impersonation and delegation. In this lesson, these concepts will be explained. The second common problem is referred to as the "mismatched SID" problem that occurs when restoring or attaching a database from another server that was using SQL Server logins. You will also see approaches for dealing with this issue in this lesson.

Objectives

After completing this lesson, you will be able to:

- Describe the typical "Double-Hop" problem.
- Explain the difference between impersonation and delegation.
- Work with security IDs.

Typical "Double-Hop" Problem



Key Points

There is a very common issue that arises when working across multiple servers. The problem occurs as shown on the slide:

1. A user starts an application. This might involve starting a web application from a corporate web server. The application either logs the user onto another Windows identity or, more commonly, impersonates the user's identity.
2. At this point, the application can perform tasks as though it was the user performing the tasks directly. The web server process might be executing as a low-privilege account but it is performing functions as though it was the Windows user. For example, the user would be able to access business functions within the application, based on the user's identity. A bank manager might be able to access the "Delete Bank Account" functionality while a bank teller might be denied access to do this.
3. The application needs to connect to a database to retrieve business data. If the database server is installed on the same server as the web server, access to the database server is made using the same impersonation details. Within the SQL Server, the user would still have the correct Windows identity. The user would have permissions that have been assigned directly to the user and permissions that have been assigned to roles that the user is a member of.
4. If, however, the database server is residing on another server, a problem occurs. When the identity of the user in SQL Server is checked, it is found to be the identity of the web server service, such as the low-privilege account, instead of the identity of the original user.

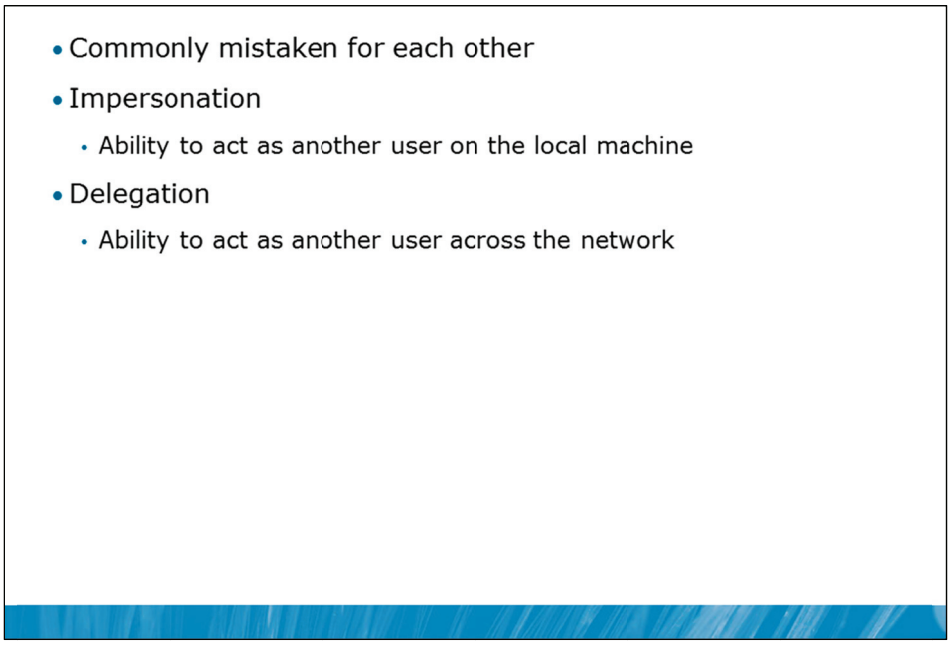
This is a very typical "double-hop" problem caused by the default action of the Windows operating system that permits impersonation but not delegation. Database administrators are often asked why this is occurring. A significant security hole would be opened if this type of access was permitted between servers by default.

What often causes confusion is that the application may have worked as expected in a development environment where the web server and the database server were both on the same server but then fails to work as expected once the application is deployed to a production environment where this is no longer the case.

In the next topic, you will see how impersonation and delegation differ.

Question: If you have seen this problem with a web server, what was the user account that often appears to be connecting to SQL Server instead of the user?

Impersonation vs. Delegation



Key Points

By default, when a user connects to a Windows server, the user is impersonated on that server. That does not, however, give a process on the server the right to impersonate that user across the network. This impersonation across a network is known as delegation. This can apply to any Windows server that is connecting to a separate SQL Server and it also applies to one SQL Server system connecting to another SQL Server system.

Delegation Requirements

To illustrate the requirements for delegation between two SQL Server systems, consider the following scenario:

- A user logs on to a client computer that connects to a server that is running an instance of SQL Server, SQLSERVER1.
- The user wants to run a distributed query against a database on another server, SQLSERVER2.
- This scenario, in which one computer connects to another computer to connect to a third computer, is an example of a "double-hop".

Each server or computer that is involved in delegation needs to be configured appropriately.

Requirements for the Client

- The Windows authenticated login of the user must have access permissions to SQLSERVER1 and SQLSERVER2.
- The user Active Directory property "Account is sensitive and cannot be delegated" must not be selected.
- The client computer must be using TCP/IP or named pipes network connectivity.

Requirements for the First/Middle Server (SQLSERVER1)

- The server must have a Server Principal Name (SPN) registered by the domain administrator.
- The account under which SQL Server is running must be trusted for delegation.
- The server must be using TCP/IP or named pipes network connectivity.
- The second server, SQLSERVER2, must be added as a linked server. This can be done by executing the sp_addlinkedserver stored procedure or by configurations within SSMS. For example:

```
EXEC sp_addlinkedserver 'SQLSERVER2', N'SQL Server‘
```

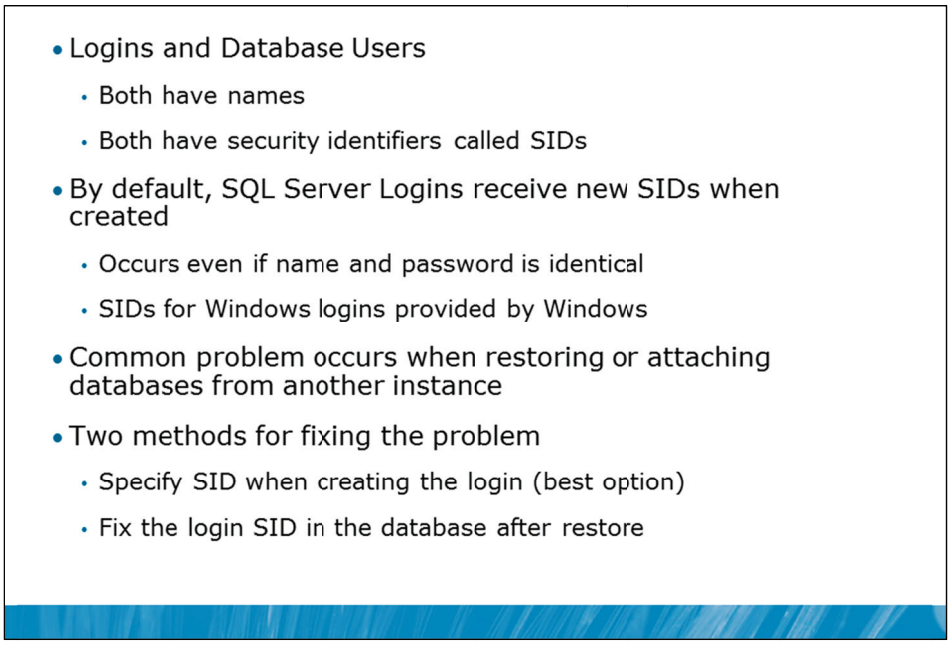
- The linked server logins must be configured for self-mapping. This can be done by executing the sp_addlinkedsrvlogin stored procedure. For example:

```
EXEC sp_addlinkedsrvlogin 'SQLSERVER2', 'true'
```

Requirements for the Second Server (SQLSERVER2)

- If using TCP/IP network connectivity, the server must have an SPN registered by the domain administrator.
- The server must be using TCP/IP or named pipes network connectivity.

Working with Mismatched Security IDs



Key Points

Another very common issue that relates to the use of multiple servers is referred to as the "mismatched SIDs" problem.

Mismatched SIDs

When a SQL Server login is created, the login is allocated both a name and a Security ID (SID). When a database user is created for the login, details of both the name and the SID for the login are entered into the database.

If the database is then backed up and restored onto another server, the database user entry is still present within the database but there is no login on the server that matches it.

Database administrators often then create the new login and map it as a user in the database and they find that this fails. When the login is created, it might have the same name and even the same password as the original login on the other server, but by default SQL Server will allocate it a new SID.

The new login will be unable to be added to the database. This can be a source of frustration because the error that is returned explains that the database user already exists, yet a check on the list of database user mappings for this login in SSMS will not show this entry. This is a good example of a situation where an understanding of T-SQL coding is helpful in SQL Server administration, rather than just an understanding of how to use the GUI in SSMS.

Resolving Mismatched SIDs

In earlier versions of SQL Server, the option provided to deal with this situation was a system stored procedure sp_change_users_login.

Starting with Service Pack 2 of SQL Server 2005, a new alternative was provided:

```
ALTER USER dbuser WITH LOGIN = loginname;
```

The problem with either of these methods is that they "fix" the SID of the database user to match the SID of the login. The next time the database is restored (as often happens), the same problem occurs again.

Avoiding the Issue

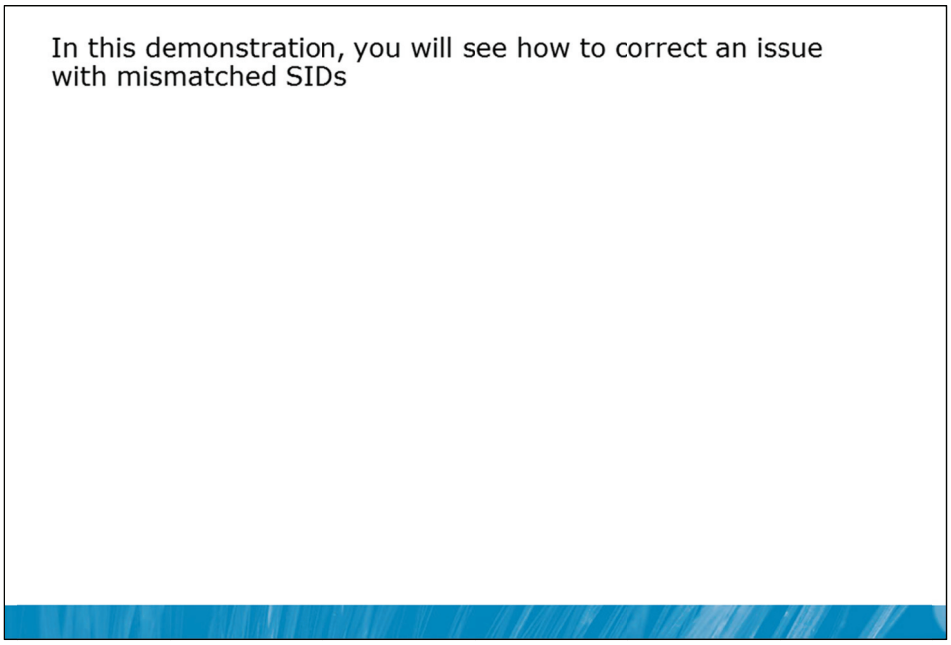
A better way of dealing with mis-matched SIDs is to avoid the problem in the first place.

The CREATE LOGIN statement has a WITH SID option. If you supply the SID from the original server while creating the login on the second server, you will avoid the problem occurring at all. The sp_helprevlogin stored procedure is another option that can be used to help with scripting SQL Server logins and it includes the value of the SID. Logins that are created using the scripts generated by this procedure will also not suffer from the mis-matched SIDs problem. Details of the sp_helprevlogin procedure (along with the source code of the procedure) are provided on the support.microsoft.com web site.

Note While SQL Server Integration Services includes a task for transferring logins, the tool disables the logins and assigns them a random password.

You will see an example of the mismatched SIDs problem in Demonstration 3A.

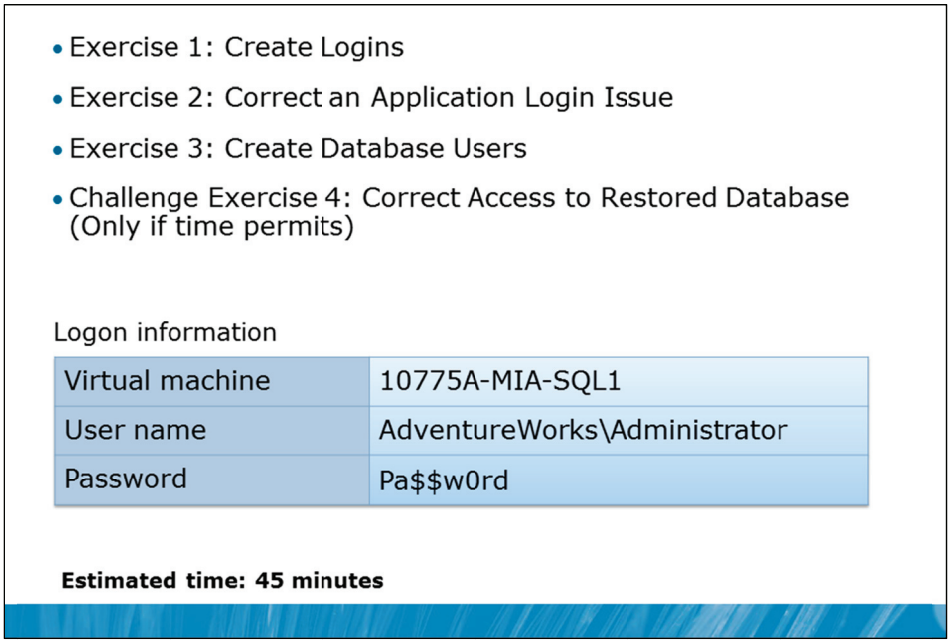
Demonstration 3A: Working with Mismatched SIDs



Demonstration Steps

1. If Demonstration 1A or 2A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server** 2012, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_09_PRJ\10775A_09_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the script files **00 – Setup.sql**, **11 – Demonstration 1A.sql** , and **21 – Demonstration 2A.sql** from within Solution Explorer.
2. Open the **31 – Demonstration 3A.sql** script file and follow the instructions contained within the comments of the script file.

Lab 9: Authenticating and Authorizing Users



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_09_PRJ\10775A_09_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

You need to configure the security for the Marketing database prior to the business accessing the system. You need to configure security so that organizational users are able to connect to SQL Server but are only to access resources that they are permitted to access. Most users will connect using their Windows group credentials. Some users however will need to use individual Windows logins. An application requires the use of a SQL Server login.

If you have time there is a problem with the LanguageDetails database that you should try to solve.

Note The changes you make will later be migrated to the production environment. You should use T-SQL commands to implement the required changes.

Supporting Documentation

Existing Windows User and Group Structure

	ITSupport	SalesPeople	CreditManagement	HumanResources	CorporateManagers
David.Alexander	X				X
Jeff.Hay	X				
Palle.Petersen	X				
Terry.Adams	X				
Darren.Parker		X			X
Mike.Ray		X			
April.Reagan		X			
Jamie.Reding		X			
Darcy.Jayne		X			
Naoki.Sato		X			
Bjorn.Rettig			X		X
Don.Richardson			X		
Wendy.Kahn			X		
Neil.Black				X	X
Madeleine.Kelly				X	

Pre-existing Security Configuration

- The SQL Login PromoteApp has been created.

Security Requirements

Note: this list of security requirements applies to several modules. For this module, you only need to consider those requirements that can be satisfied by topics covered in this module and the assigned tasks in the lab instructions.

1. The senior DBA Jeff Hay should have full access to and control of the entire Proseware server instance.
2. All ITSupport group members should have full access to and control of the MarketDev database.
3. Proseware uses an application called DBMonitor from Trey Research. This application requires a SQL login called DBMonitorApp, which requires the ability to read but not update all objects in the MarketDev database. It does not require access to other databases.
4. All CorporateManagers group members perform periodic Strength, Weakness, Opportunity, and Threat (SWOT) analysis. For this they need to be able to both read and update rows in the DirectMarketing.Competitor table.
5. All SalesPeople group members should be able to read data from all tables in the DirectMarketing schema, except April Reagan who is a junior work experience student.
6. Only ITSupport group members and members of the CreditManagement group should be able to update the Marketing.CampaignBalance table directly.
7. Within the company members of the SalesPeople group, the CreditManagement group, and the CorporateManagers group are referred to as sales team members.
8. All sales team members should be able to read rows in the Marketing.CampaignBalance table.
9. All sales team members should be able to read rows in the DirectMarketing.Competitor table.
10. The Sales Manager should be able to read and update the Marketing.SalesTerritory table.
11. All HumanResources group members should be able to read and update rows in the Marketing.SalesPerson table.
12. The Sales Manager should be able to execute the Marketing.MoveCampaignBalance stored procedure.
13. All sales team members should be able to execute all stored procedures in the DirectMarketing schema.

Exercise 1: Create Logins

Scenario

You have been provided with the security requirements for the MarketDev database. In this exercise you need to create individual Windows logins, Windows group logins, and SQL logins that are required to implement the security requirements.

The main tasks for this exercise are as follows:

1. Review the requirements.
2. Create the required logins.

Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation.

Task 2: Create the required logins

- Create the logins that you have determined are required for the system. This will include Windows logins, Windows group logins, and SQL logins.

Results: After this exercise, you have created the required Windows and SQL logins.

Exercise 2: Correct an Application Login Issue

Scenario

The Adventure Works IT department has implemented a new web application called Promote. The Promote application requires a SQL login called PromoteApp. The SQL login needs to operate with a fixed password. The application has been operating for some time but has now stopped working. It appears the application is unable to log on to the Proseware server. You need to reset the password for the PromoteApp user and disable policy checking for the login.

The main task for this exercise is as follows:

1. Correct an Application Login Issue.

Task 1: Correct an application login issue

- Reset the password for the PromoteApp SQL login to "Pa$$w0rd".
- Disable policy checking for the application login.

Results: After this exercise, you have corrected an application login issue.

Exercise 3: Create Database Users

Scenario

You have created the required logins for the Proseware server as per the security requirements that you have been supplied. You need to create database users for those logins in the MarketDev database.

The main tasks for this exercise are as follows:

1. Review the requirements.
2. Create the required database users.

Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation.

Task 2: Create the required database users

- Create the database users that you have determined are required for the MarketDev database.

Results: After this exercise, you should have created the required database users.

Challenge Exercise 4: Correct Access to Restored Database (Only if time permits)

Scenario

A junior DBA has been trying to restore the LanguageDetails database and grant access to a SQL login named LDUser. He was able to restore the database and to create the login but he has been unable to create the database user. He suspects that something in the existing database is preventing this as he can create and assign other SQL logins without issue. You need to restore the LanguageDetails database, create the LDUser login, create the LDUser database user, and test that the user can access the database.

The main task for this exercise is as follows:

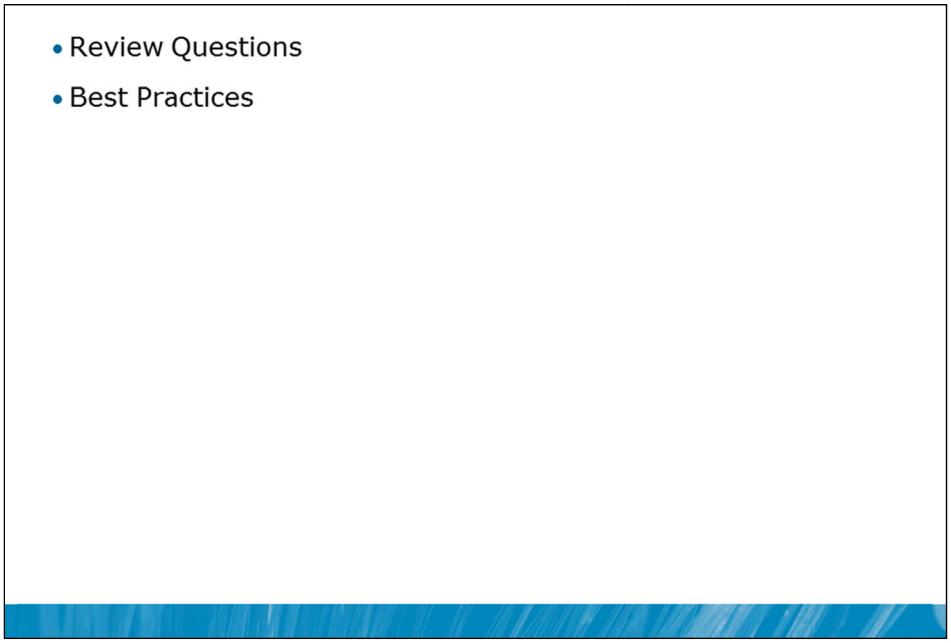
1. Correct Access to a Restored Database.

▶ Task 1: Correct Access to a restored database

- Restore the LanguageDetails database from the file D:\10775A_Labs\10775A_09_PRJ\LanguageDetails.bak to the Proseware server instance.
- Create the login LDUser with policy checking disabled and a password of "Pa$$w0rd".
- Correct access to the LanguageDetails database for the LDUser database user.
- Test that the LDUser login can access the database and can select the rows from the dbo.Language table.

Results: After this exercise, you should have resolved the situation.

Module Review and Takeaways



Review Questions

1. How does SQL Server take advantage of Windows password Policy?
2. What account policy is applied on Windows XP?

Best Practices

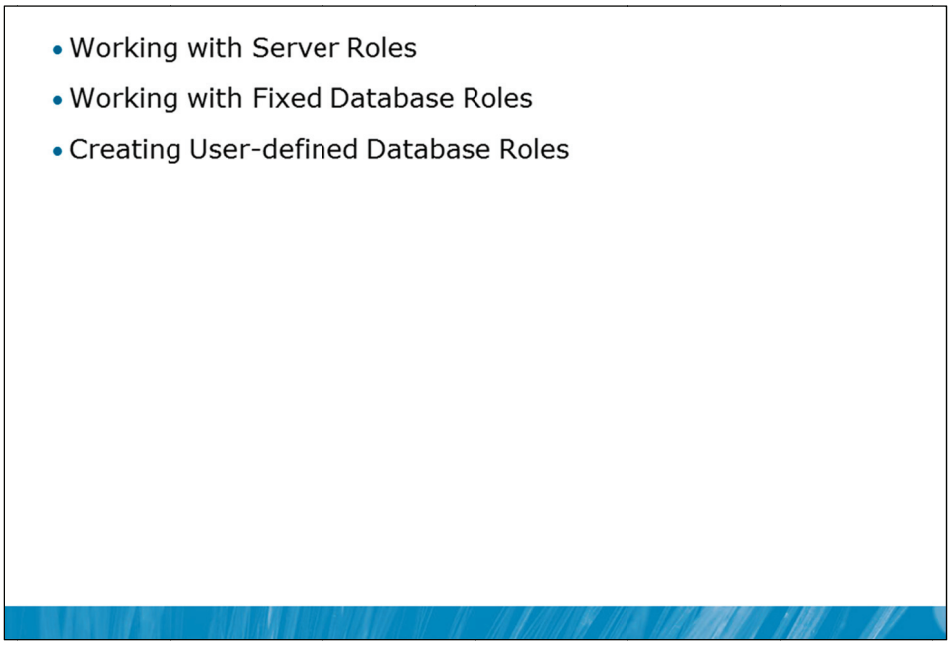
1. Minimize the number of SQL Server logins.
2. Ensure that expiry dates are applied to logins that are created for temporary purposes.
3. Disable logins rather than dropping them if there is any chance that they will be needed again.
4. Configure Kerberos delegation when a Windows user identity needs to be passed between systems.

Module 10

Assigning Server and Database Roles

Contents:

Module Overview



Once a login has been authenticated by the server, and mapped to a database user, you need to assign permissions to the login or database user. Permissions can be assigned directly to the login or database user. In the next module, you will see how permissions are directly assigned but while it is possible to do so, where sets of permissions potentially apply to multiple users, roles can be used to provide the permissions instead.

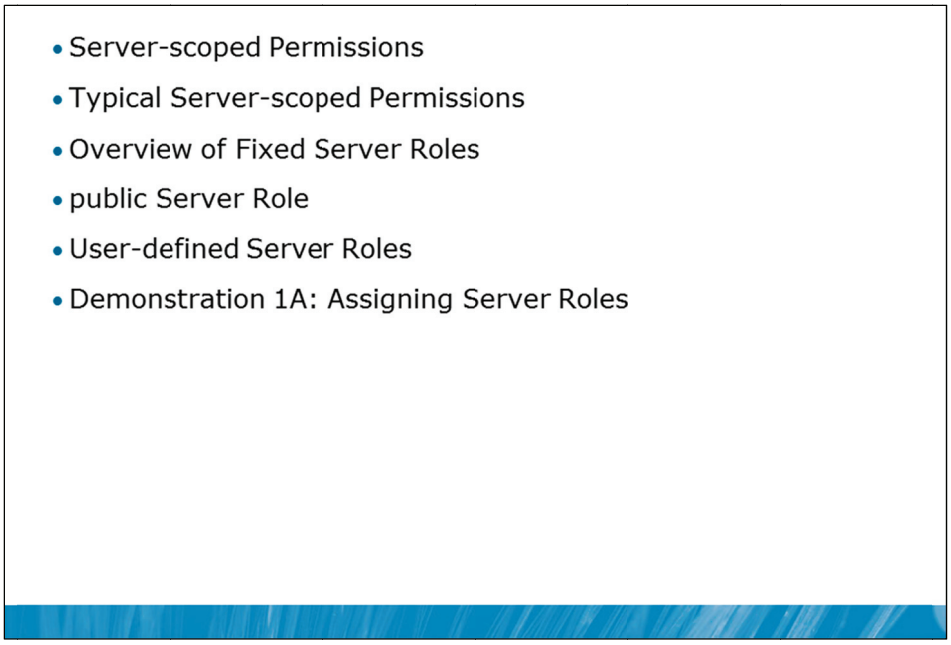
SQL Server® has a set of fixed roles at both the server and database levels and user-defined roles can also be created at both levels. The fixed roles have a specified set of permissions but user-defined roles have a user-defined set of permissions applied to them. A login or database user is assigned the permissions associated with any role that the login or database user is a member of.

Objectives

After completing this module, you will be able to:

- Work with server roles.
- Work with fixed database roles.
- Create user-defined database roles.

Lesson 1
Working with Server Roles



The first type of role that you will investigate is a Server Role. Server Roles have permissions that span the entire server instance. While Server Roles are very powerful, they should be used rarely.

The most powerful Server Role is the sysadmin role. You should be cautious about assigning logins to this role as members of this role have complete access to the entire server.

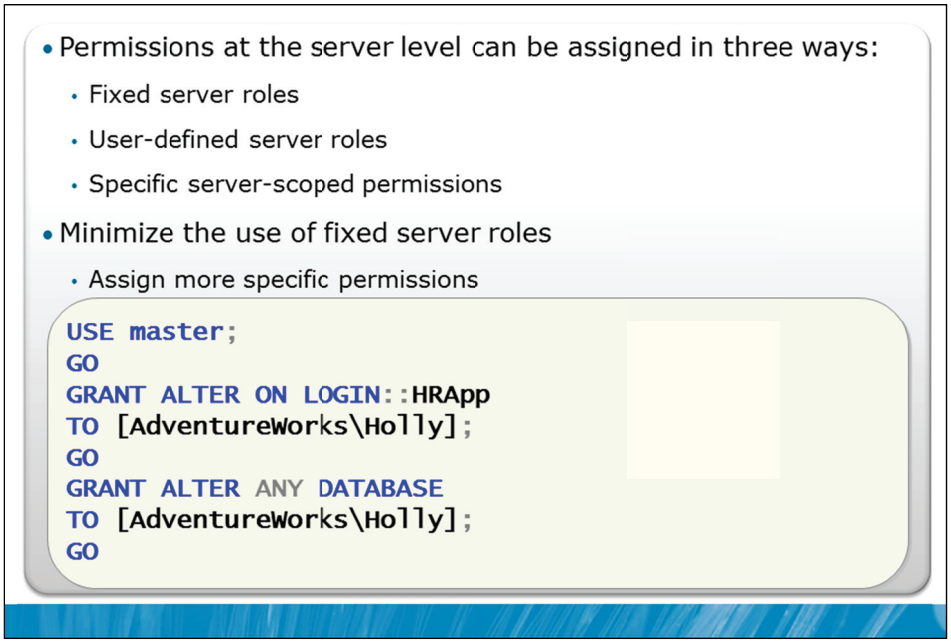
In this lesson, you will also investigate the public role. public can be used to assign instance-level permissions to all logins.

Objectives

After completing this lesson, you will be able to:

- Describe server-scoped permissions.
- Describe permissions that might typically be assigned at the server level.
- Explain the use of fixed server roles.
- Explain the purpose of the public server role.
- Configure user-defined server roles.

Server-scoped Permissions



Key Points

Permissions that are allocated at the server level are very powerful. They apply to the entire server and all databases on the server. For this reason, you should avoid allocating server-scoped permissions and try to assign more specific permissions instead.

Fixed server-scoped roles have been part of SQL Server for many versions and are mostly maintained for backward compatibility.

Slide Examples

On this slide, two examples are provided of granting server-level permissions. The first example shows how to grant a specific permission. The Windows login AdventureWorks\Holly is being granted the permission to alter the login HRApp.

The second example shows how to use the ANY option. ANY can be applied to many object types, including objects that are scoped to the server level. The use of the ANY option allows for permissions to be assigned on specific classes of object without the need to create large numbers of individual permissions. Note that as well as granting AdventureWorks\Holly the ability to alter any existing databases, this permission grant would also apply to any future databases that are created on the server.

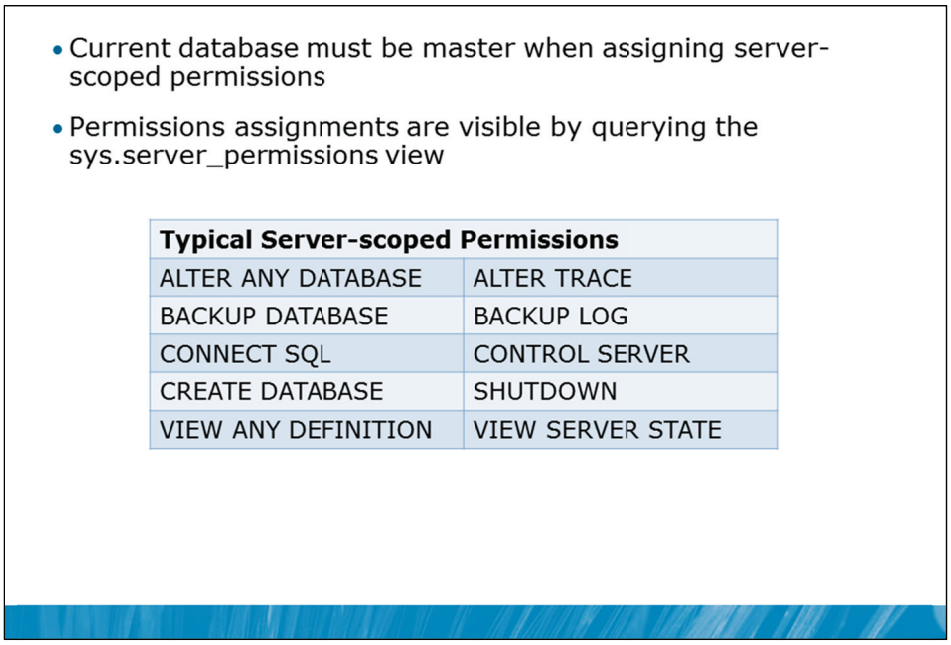
The full syntax of the GRANT statement is complex. More details on the GRANT statement will be provided in Module 11.

Assignment Using the GUI

SQL Server securables are any objects that can be protected via permissions.

Note If no securable is shown on the tab in SSMS, do not assume that no permissions are assigned to any securable. Most times, the securable that needs to be viewed or changed needs to be added manually.

Typical Server-scoped Permissions



Key Points

On this slide, you can see a list of server-scoped permissions that are commonly assigned. As mentioned, very few permissions are typically assigned at the server level.

These permissions cannot be assigned unless the current database is the master database. Otherwise, an error will be returned.

The sys.server_permissions view is used to query the currently assigned server-scoped permissions. You will see an example of it being used in Demonstration 1A.

Overview of Fixed Server Roles

Role	Description	Server-level Permission
sysadmin	Perform any activity	CONTROL SERVER (with GRANT option)
dbcreator	Create and alter databases	ALTER ANY DATABASE
diskadmin	Manage disk files	ALTER RESOURCES
serveradmin	Configure server-wide settings	ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
securityadmin	Manage and audit server logins	ALTER ANY LOGIN
processadmin	Manage SQL Server processes	ALTER ANY CONNECTION ALTER SERVER STATE
bulkadmin	Run the BULK INSERT statement	ADMINISTER BULK OPERATIONS
setupadmin	Configure replication and linked servers	ALTER ANY LINKED SERVER

Key Points

On this slide, you can see the list of fixed server roles, a general description of what they are used for and a list of the major permissions that are associated will each of the roles.

Note The list of permissions shown is not an exhaustive list. For full details, consult Books Online.

These server-level roles are referred to as fixed server roles because you cannot change the permissions associated with these roles.

Populating Fixed Server Roles

You can add logins into server-level roles. One behavior that might not be expected is that each member of a fixed server role can add other logins to that same role.

Roles vs. Permissions

Being assigned the same permissions as those that are assigned to a role is not the same as being a member of the role.

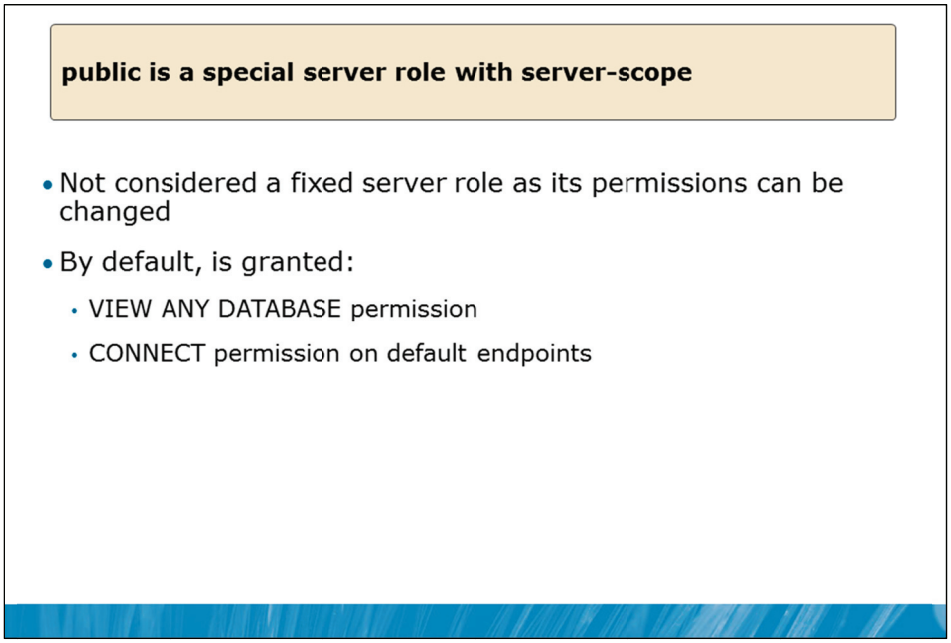
For example, granting CONTROL SERVER permission to a login is not the same as making the login a member of the sysadmin fixed server role. sysadmin members are automatically assigned to the dbo user in a database. This does not happen for logins that have been only been granted CONTROL SERVER permission.

Note Securityadmin is almost functionally equivalent to the sysadmin role and should be granted with caution.

To change the membership of fixed server roles, use the ALTER SERVER ROLE command. You will see an example of this in Demonstration 1A.

Question: Why would the securityadmin role be powerful?

public Server Role



Key Points

public is a special role that is also scoped to the server level. It is not considered a fixed server role as it is possible to alter the permissions that are assigned to the public role. All logins are automatically members of the public role, so this is another role that should be assigned permissions with caution.

VIEW ANY DATABASE

Many SQL Server tools, utilities and applications assume that the list of databases can be viewed, even if the user that is viewing the list does not have permission to perform any actions on or in the database.

This is achieved by the public role having VIEW ANY DATABASE permission. While it is possible to stop the ability to view databases that you do not have access to, be cautious about doing so. SSMS, in particular, becomes very slow to use if you do this, as it then needs to check your permissions within every database.

One possible use case for this would be hosted databases, where the provision of a database for a user should not enable the user to view all the other databases on the server. Test the behavior and performance of your database tooling and applications before deciding to change this.

Connect Permissions

By default, the public role has been granted CONNECT permission on the endpoints for the Shared Memory, TCP, Named Pipes, and VIA protocols. This allows users to connect to the server using any of these protocols.

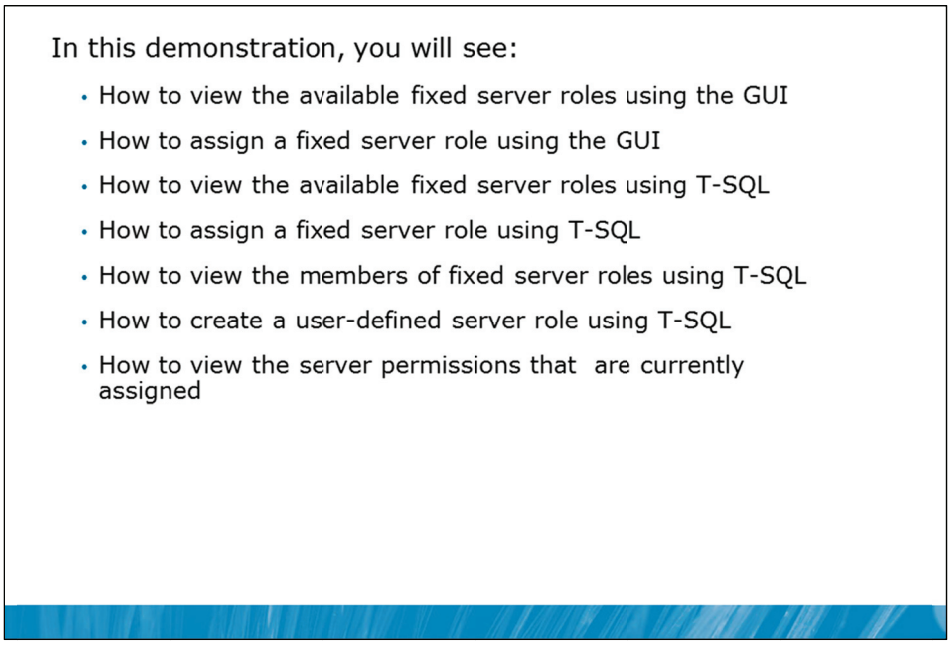
User-defined Server Roles

Key Points

New to SQL Server 2012 is the ability to create user-defined roles at the server level.

In general, you should avoid using fixed server roles as they tend to provide users with more permissions than required to do their assigned tasks. User-defined server roles allow you to configure a specific set of server-level permissions that are required for members of the role.

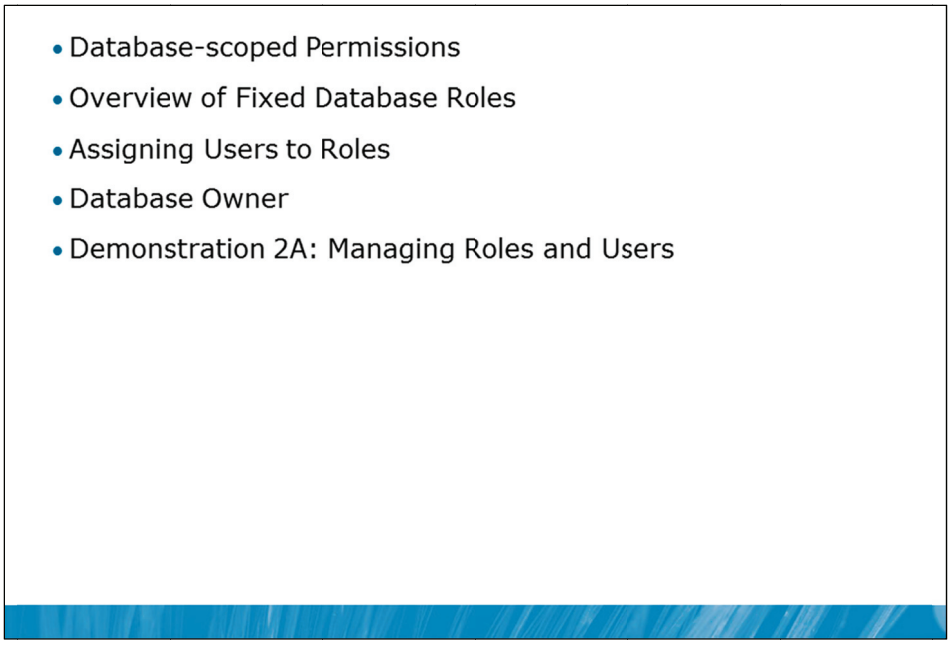
Demonstration 1A: Assigning Server Roles



Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_10_PRJ\10775A_10_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 2
Working with Fixed Database Roles



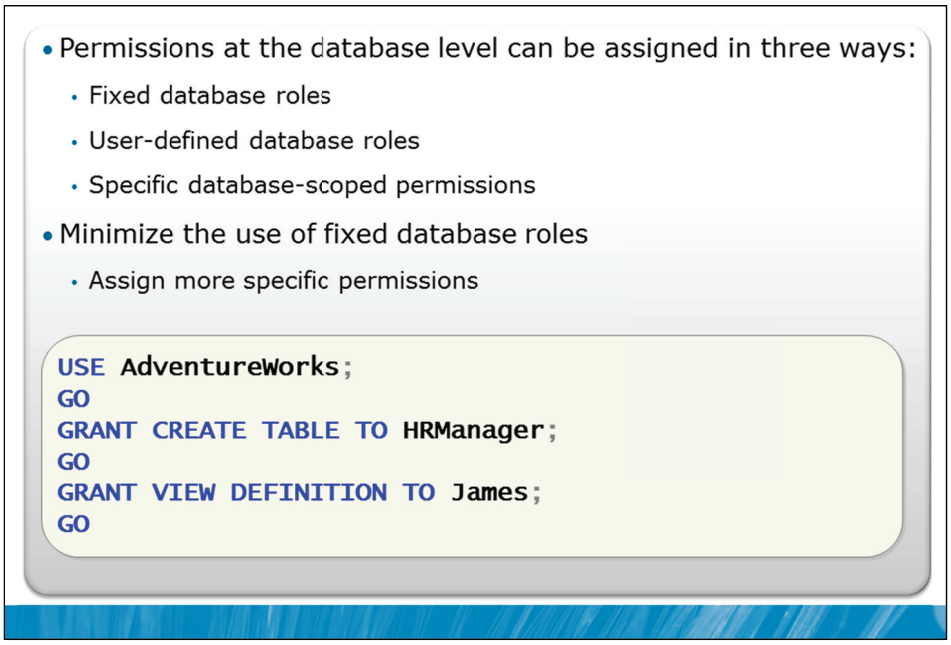
In addition to the roles provided at the server level, SQL Server provides a set of fixed roles at the database level. These fixed database roles are similar in concept to the fixed server roles but they relate to access to database objects or access to the database itself, rather than access to all databases on the server.

Objectives

After completing this lesson, you will be able to:

- Explain the available database-scoped permissions.
- Describe fixed database roles.
- Assign users to roles.
- Describe the concept of the database owner.

Database-scoped Permissions



Key Points

There are three ways that permissions can be assigned at the database level:

- Fixed database roles are very similar to fixed server roles, apart from the objects that they apply to and that the scope is database level rather than server level.
- Similar to server roles, it is possible to create user-defined database roles and to assign a user-defined set of permissions to each role.
- Database-scoped permissions can be individually assigned.

Try to avoid assigning fixed database roles as they usually provide more capabilities than those that are required for most users. Assign much more specific permissions instead of the fixed database role membership.

Slide Example

In the first example on the slide, the database user HRManager is being granted permission to create tables within the database. It is important to understand that several permissions might be required to perform an action. In this example, HRManager would also require ALTER SCHEMA permission to successfully create tables.

In the second example on the slide, the database user James is being assigned permission to view the definitions of objects within the database. This permission grant is common for users that need to perform documentation but who are not permitted to alter the design of the database.

Overview of Fixed Database Roles

Role	Description
db_owner	Perform any configuration and maintenance activities on the DB and can drop it
db_securityadmin	Modify role membership and manage permissions
db_accessadmin	Add or remove access to the DB for logins
db_backupoperator	Back up the DB
db_ddladmin	Run any DDL command in the DB
db_datawriter	Add, delete, or change data in all user tables
db_datareader	Read all data from all user tables
db_denydatawriter	Cannot add, delete, or change data in user tables
db_denydatareader	Cannot read any data in user tables

Key Points

The slide shows the available fixed database roles and a description of the purpose they are intended to fill.

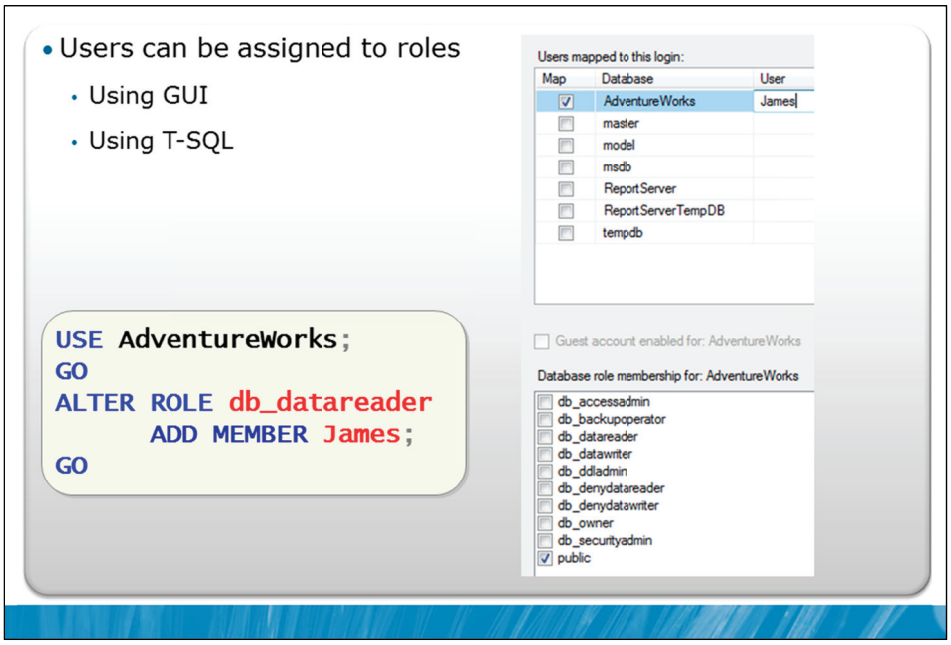
In practice, fixed database roles should be assigned very selectively. Instead of assigning fixed database roles, consider assigning much more specific permissions.

For example, instead of assigning the db_datareader role to a user, consider assigning SELECT permission on the objects that the user needs to be able to SELECT. This avoids situations where additional tables are added to a database and the user already has permissions on those tables. Even if the user needs permission to SELECT all objects in the database, it would be preferable to assign SELECT permissions at the database level than to assign the user membership of the db_datareader role. One reason for this is that it is easier to review the permission assignments using views such as sys.database_permissions.

Similar to fixed server roles, fixed database roles exist largely for backward compatibility.

Note Similar to the securityadmin fixed server role, adding users to the db_securityadmin role should be performed with caution as the permission is almost functionally equivalent to the dbo user.

Assigning Users to Roles



Key Points

Users can be assigned to roles either using the GUI in SSMS or by using T-SQL commands.

GUI

To assign a role to a user using SSMS, expand the server, expand the Security node, expand the Logins node, then right-click the login and click Properties. In the Properties window for the login, click on the User Mapping tab to see the list of databases that the login has been mapped to. As you select each database in the upper pane, a list of database roles appears in the lower pane. You can then select the roles that the login should be assigned to.

You can also assign role membership from the perspective of the role. To assign or remove users from a role using SSMS, expand the relevant database, expand the Security node, expand the Roles node, expand the Database Roles node, then right-click the relevant role and click Properties. From the Properties screen for the role, you can add and delete users from that database.

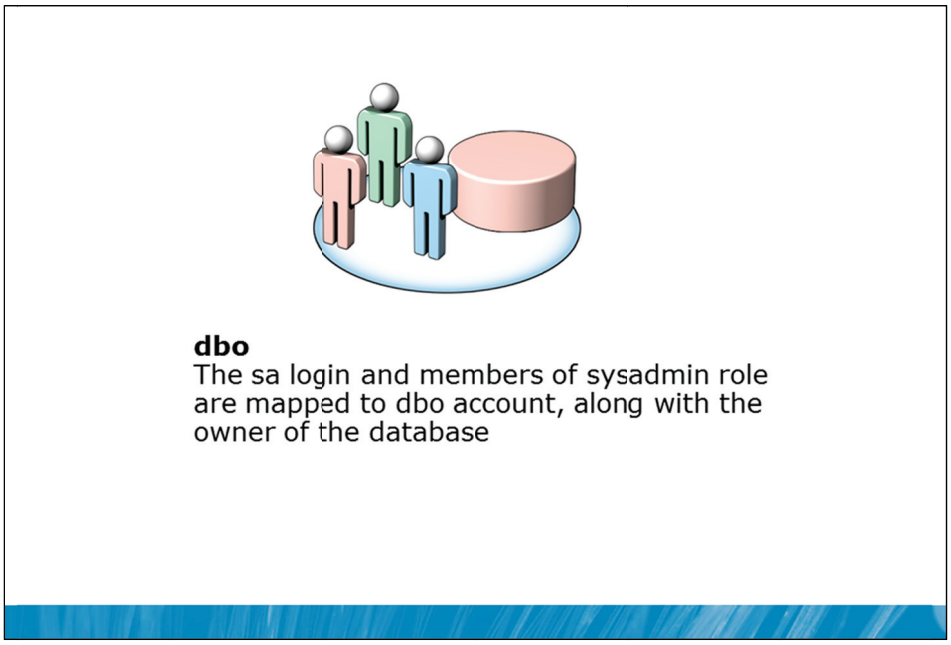
Managing Role Membership via T-SQL

In the example on the slide, you can see the use of the ALTER ROLE statement. The user James is being added to the db_datareader fixed database role.

The DROP MEMBER clause of the ALTER ROLE statement removes members from roles.

> **Question:** Can you think of an example of a command that could be executed by a member of the db_ddladmin role?

Database Owner



Key Points

It was mentioned in the last module that there are two special users: dbo and guest. Guest was described in the last module, along with an introduction to dbo.

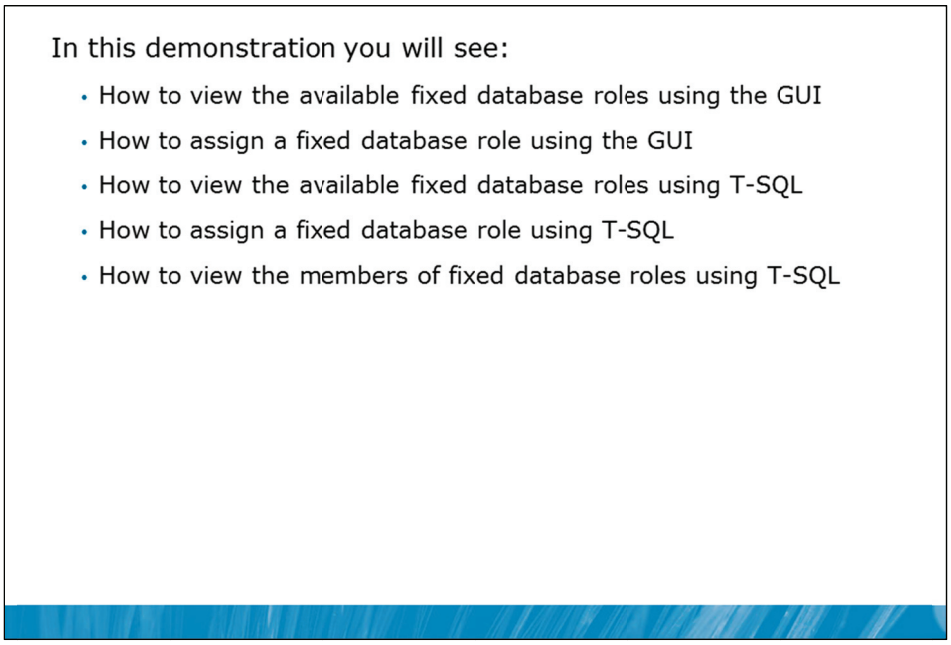
dbo

A dbo user has implied permissions to perform all activities in the database. The "sa" login and any member of the sysadmin fixed server role that uses a database are automatically mapped to the dbo user. Like other objects in SQL Server, databases have owners. The owner of the database is also mapped to the dbo user.

Note Schema-scoped objects are automatically owned by the owner of the schema, regardless of who creates them. Non-schema-scoped objects are automatically owned by the database principal that created them. For any principal with db_owner rights, the owner would be dbo.

dbo cannot be deleted and is present in every database. Equivalence to dbo is special in that once SQL Server finds that a user is mapped to dbo, no other permission checks are made within the database. This means that you cannot later DENY a permission within the database to a user with dbo equivalence.

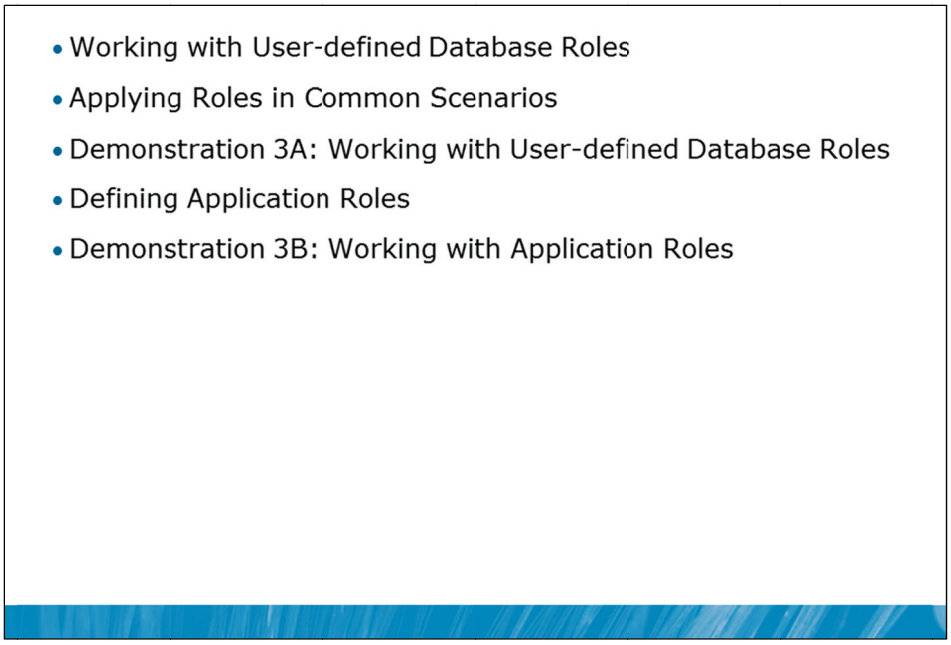
Demonstration 2A: Managing Roles and Users



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_10_PRJ\10775A_10_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
   - Open and execute the **11 – Demonstration 1A.sql** script file from within Solution Explorer.
2. Open the **21 – Demonstration 2A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lesson 3
Creating User-defined Database Roles



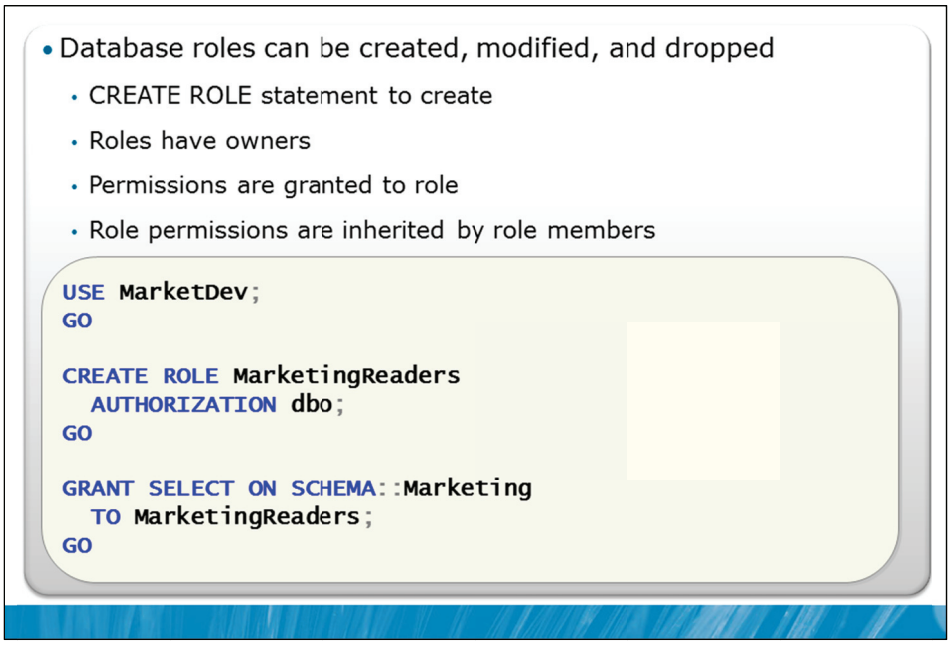
In the first two lessons in this module, you have seen how to work with to both fixed and user-defined server roles and with fixed database roles. Similar to user-defined server roles, SQL Server provides you with an option to create user-defined database roles. You should consider user-defined database roles rather than fixed database roles in most situations. The appropriate design of user-defined database roles is critical when designing a security architecture for your database.

Objectives

After completing this lesson, you will be able to:

- Work with user-defined database roles.
- Apply roles in common scenarios.
- Define application roles.

Working with User-defined Database Roles



Key Points

User-defined database roles are created using the CREATE ROLE statement.

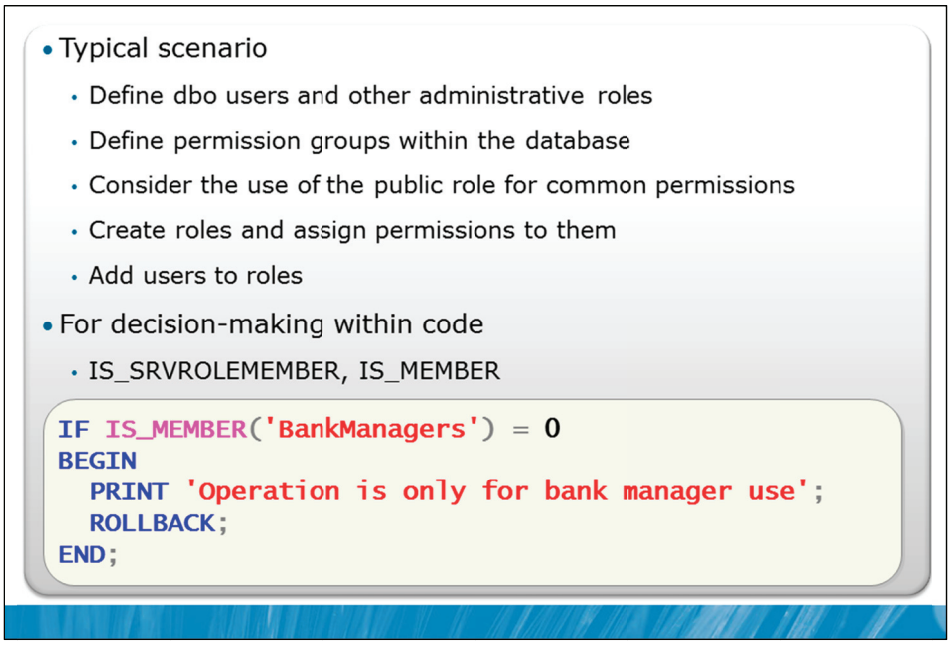
In the example shown on the slide, a new role called MarketingReaders is being created. Note that like other objects in SQL Server, roles have owners. Owners of roles have full control of the role. In the example on the slide, dbo is being assigned as the owner of the MarketingReaders role.

Roles and Permissions

Once user-defined database roles are created, they are assigned permissions in the same way that permissions are assigned to database users.

In the example shown on the slide, SELECT permission is being granted on the Marketing schema to members of the MarketingReaders role. From that point on, any users that are added to the MarketingReaders role will have permission to SELECT any object that is contained within the Marketing schema. (The assignment of permissions on objects including schemas is covered in the next module).

Applying Roles in Common Scenarios



Key Points

It is important to apply a prescriptive process when applying roles in typical business applications.

1. You should start by defining any administrative roles at the server or database level and determining an appropriate dbo user. There should be few users who are equivalent to dbo.
2. You should consider the types of access that each user needs. This may involve considering how the requirements overlap with their Windows group membership. Where a number of users (including group users) need common permissions, define the permission groups as roles.
3. If all users need a set of permissions, consider the use of the public role within the database. For example, formatting functions would be good examples of code that any user could potentially be permitted to execute.
4. For remaining permission groups, create appropriate roles and assign the permissions to those roles.
5. Add the users that need the groups of permissions to the roles that provide those permissions.

Note that unlike Windows domain groups that are typically named after the users that are members of them (such as Salespeople), roles should be named based on the permissions that they provide. They are more like Windows local groups in this regard.

Testing for Role Membership in T-SQL Code

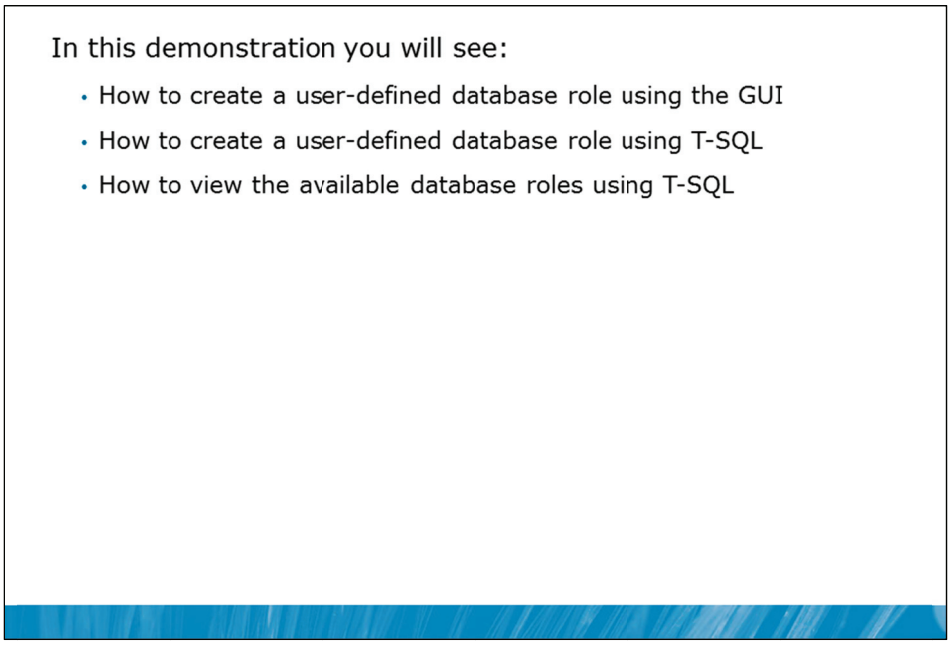
While you may decide to allow members of a role to execute T-SQL code (such as a stored procedure) or not, it is also possible to place limits within the code module.

For example, you may decide that BankManagers and BankTellers can both transfer funds but BankTellers might have a lower limit on the transfer amounts.

The IS_SRVROLEMEMBER function tests for server role membership, and the IS_MEMBER function tests for database role membership. IS_MEMBER can also test for Windows group membership.

In the code fragment shown in the slide, an operation is being rolled back if the user is not a member of the BankManagers group.

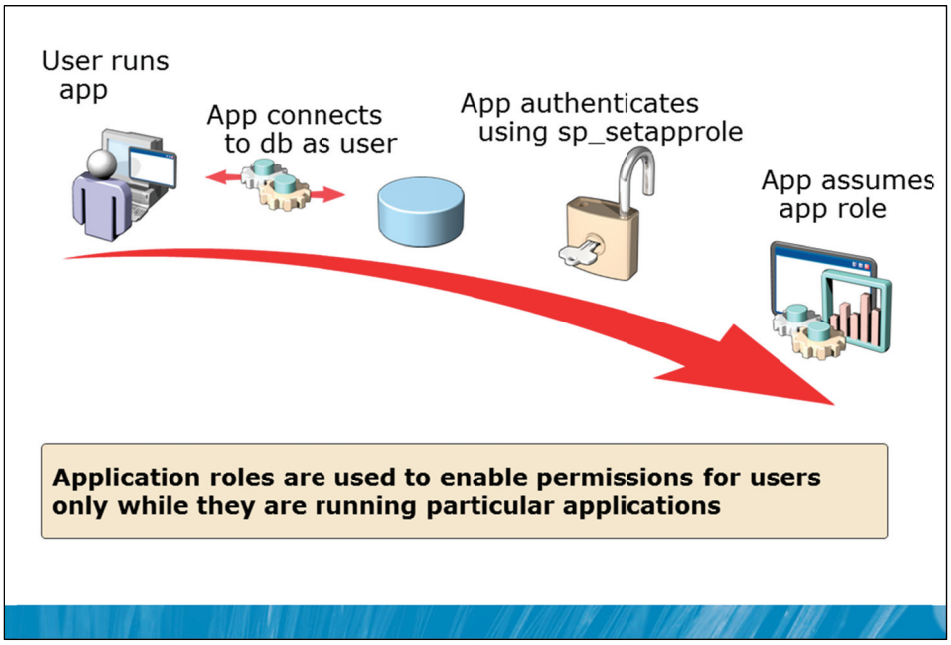
Demonstration 3A: Working with User-defined Database Roles



Demonstration Steps

1. If Demonstration 1A or 2A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_10_PRJ\10775A_10_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the script files **00 – Setup.sql**, **11 – Demonstration 1A.sql** , and **21 – Demonstration 2A.sql** from within Solution Explor**er**.
2. Open the **31 – Demonstration 3A.sql** script file.
3. Follow the instructions contained within the comments of the script file to execute each T-SQL batch contained in the file.

Defining Application Roles



Key Points

Application roles are used to enable permissions for users only when they are running particular applications.

For example, imagine that you might want a user to be able to update rows in a table while using an application. You might not, however, want the same user to be able to open the table in SSMS and edit the rows, or to connect to the table from Microsoft Excel.

Application Roles

Application roles are one solution to this scenario. Application roles contain no members. They are a special type of role that is assigned permissions.

A user will typically connect to an application using a low-privilege account. The application then calls the sp_setapprole system stored procedure to "enter" the application role. At that point, the user permissions for the existing connection are replaced by those from the application role.

Note The permissions of the application role are not added to the permissions of the user. The permissions of the application role replace the permissions of the user.

Reverting from an Application Role

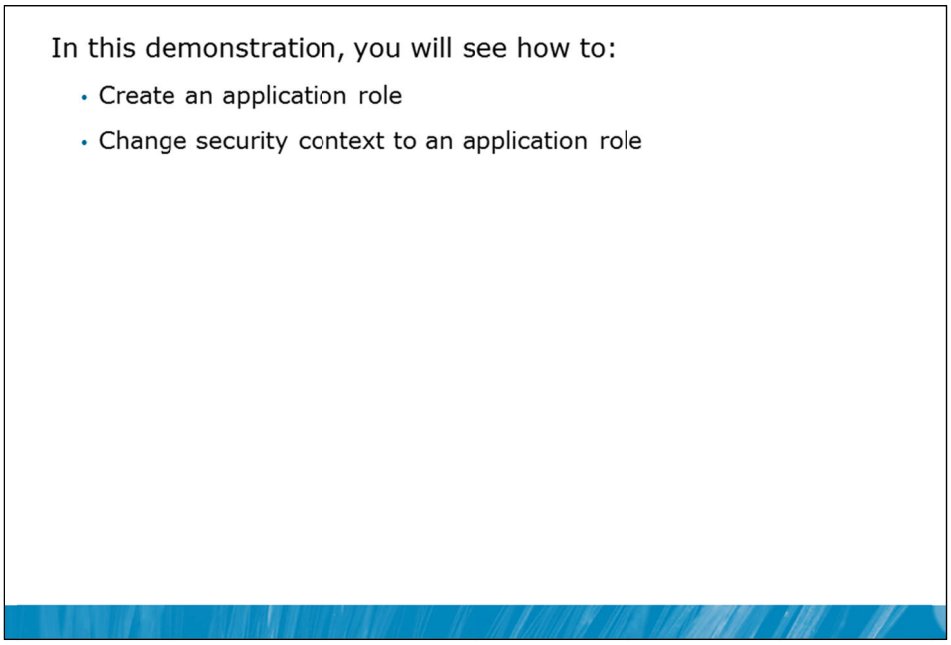
In versions of SQL Server prior to SQL Server 2005, there was no way to revert to the original security context. From SQL Server 2005 onwards, the sp_setapprole system stored procedure creates a cookie. The application can store the cookie and later pass the cookie back to the sp_unsetapprole system stored procedure, to revert to the original security context.

Password and Encryption

To stop other applications and users from entering the application role, each application role is assigned a password. The application must provide the password when calling sp_setapprole.

To avoid exposure to network packet sniffing an option is provided to send the password in an encrypted form, to avoid it being visible in network traces. Be cautious about depending upon this behavior entirely as the encryption method used is not particularly strong.

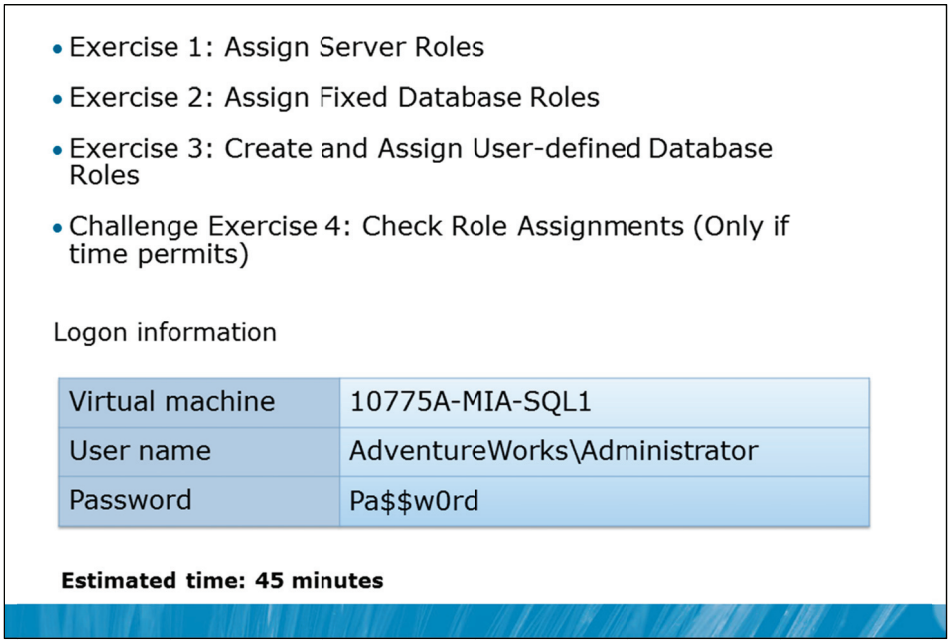
Demonstration 3B: Working with Application Roles



Demonstration Steps

1. If Demonstration 1A, 2A or 3A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_10_PRJ\10775A_10_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the script files **00 – Setup.sql**, **11 – Demonstration 1A.sql** , **21 – Demonstration 2A.sql**, and **31 – Demonstration 3A.sql** from within Solution Explorer.
2. Open the **32 – Demonstration 3B.sql** script file.
3. Follow the instructions contained within the comments of the script file to execute each T-SQL batch contained in the file.

Lab 10: Assigning Server and Database Roles



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_10_PRJ\10775A_10_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

You have created the SQL Server logins and Database users. You now need to assign the logins and users to the required roles based upon the security requirements for the MarketDev database. You should assign the minimum level of access that will allow each user to perform their job. This will require a combination of server, fixed database, and user defined database roles.

Do not be concerned with object and schema permissions as these will be assigned in Module 11 but you do need to consider the role requirements that will be required at that time.

Note The changes you make will later be migrated to the production environment. You should use T-SQL commands to implement the required changes.

Supporting Documentation

Existing Windows User and Group Structure

	ITSupport	SalesPeople	CreditManagement	HumanResources	CorporateManagers
David.Alexander	X				X
Jeff.Hay	X				
Palle.Petersen	X				
Terry.Adams	X				
Darren.Parker		X			X
Mike.Ray		X			
April.Reagan		X			
Jamie.Reding		X			
Darcy.Jayne		X			
Naoki.Sato		X			
Bjorn.Rettig			X		X
Don.Richardson			X		
Wendy.Kahn			X		
Neil.Black				X	X
Madeleine.Kelly				X	

Pre-existing Security Configuration

- The following Windows group logins and database users have been created:
  - AdventureWorks\ITSupport
  - AdventureWorks\SalesPeople
  - AdventureWorks\CreditManagement
  - AdventureWorks\HumanResources
  - AdventureWorks\CorporateManagers
- The following Windows logins and database users have been created:
  - AdventureWorks\Jeff.Hay
  - AdventureWorks\April.Reagan
  - AdventureWorks\Darren.Parker
- The following SQL logins have been created:
  - PromoteApp
  - DBMonitorApp

Security Requirements

1. The senior DBA Jeff Hay should have full access to and control of the entire Proseware server instance.
2. All ITSupport group members should have full access to and control of the MarketDev database.
3. Proseware uses an application called DBMonitor from Trey Research. This application requires a SQL login called DBMonitorApp, which requires the ability to read but not update all objects in the MarketDev database.
4. All CorporateManagers group members perform periodic Strength, Weakness, Opportunity, and Threat (SWOT) analysis. For this they need to be able to both read and update rows in the DirectMarketing.Competitor table.
5. All SalesPeople group members should be able to read data from all tables in the DirectMarketing schema, except April Reagan who is a junior work experience student.
6. Only ITSupport group members and members of the CreditManagement group should be able to update the Marketing.CampaignBalance table directly.
7. Within the company members of the SalesPeople group, the CreditManagement group, and the CorporateManagers group are referred to as sales team members.
8. All sales team members should be able to read rows in the Marketing.CampaignBalance table.
9. All sales team members should be able to read rows in the DirectMarketing.Competitor table.
10. The Sales Manager should be able to read and update the Marketing.SalesTerritory table.

11. All HumanResources group members should be able to read and update rows in the Marketing.SalesPerson table.
12. The Sales Manager should be able to execute the Marketing.MoveCampaignBalance stored procedure.
13. All sales team members should be able to execute all stored procedures in the DirectMarketing schema.

Exercise 1: Assign Server Roles

Scenario

You need to implement any required server roles that are needed to support the supplied security requirements.

The main tasks for this exercise are as follows:

1. Review the requirements.
2. Assign any required server roles.

▶ Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation.

▶ Task 2: Assign any required server roles

- Assign any server roles that are required to support the security requirements for the MarketDev database.

Results: After this exercise, you should have assigned any required server roles.

Exercise 2: Assign Fixed Database Roles

Scenario

You have been provided with a set of requirements detailing the access that each login needs to the MarketDev database. Some of these requirements might be met by fixed database roles but it is important to not provide permissions that are not specifically required. If you consider there is a need for user-defined database roles these will be created in the next exercise.

The main tasks for this exercise are as follows:

1. Review the requirements.
2. Assign any required fixed database roles.

▶ Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation.

▶ Task 2: Assign any required fixed database roles

- Assign any fixed database roles that are required to support the security requirements for the MarketDev database.

Results: After this exercise, you have assigned fixed database roles as required.

Exercise 3: Create and Assign User-defined Database Roles

Scenario

You have been provided with a set of requirements detailing the access that each login needs to the MarketDev database. In Exercise 2, you assigned fixed database role membership. Other requirements might be best supported by user-defined database roles. In this exercise you will create and assign required user-defined database roles.

The main tasks for this exercise are as follows:

1. Review the requirements.
2. Create and assign any required user-defined database roles.

▶ Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation.

▶ Task 2: Create and assign any required user-defined database roles

- Create and assign any user-defined database roles that are required to support the security requirements for the MarketDev database.

Results: After this exercise, you have created and assigned user-defined database roles as required.

Challenge Exercise 4: Check Role Assignments (Only if time permits)

Scenario

You have created logins and database users, assigned server and database roles, and created and assigned user-defined database roles. It is important to check that the role assignments are operating as expected. In this exercise you will use the sys.login_token and sys.user_token system views to check the available tokens for Darren Parker.

The main task for this exercise is as follows:

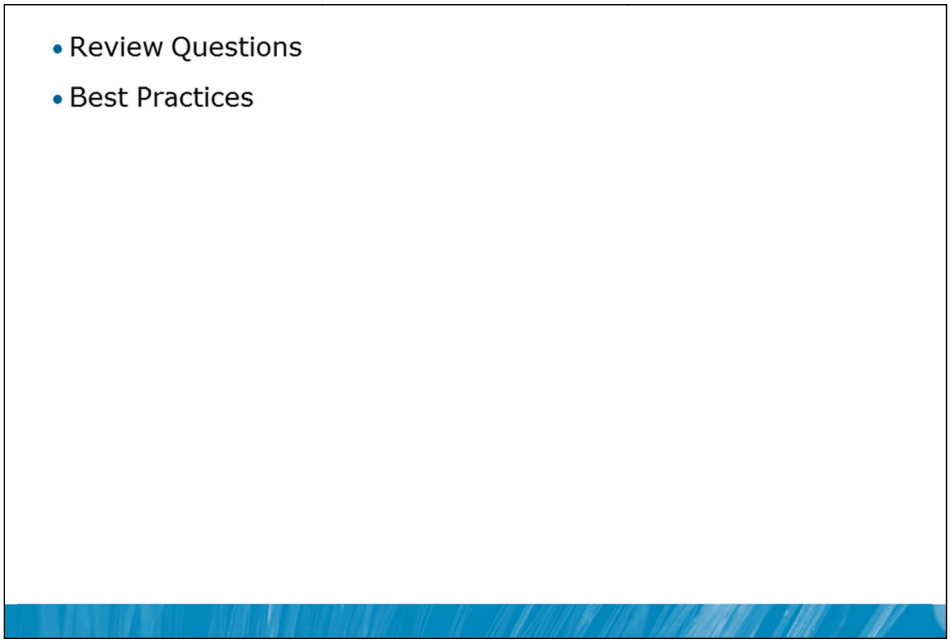
1. Check the role assignments for Darren Parker.

▶ Task 1: Check the role assignments for Darren Parker

- Using the EXECUTE AS statement, change your security context to the login AdventureWorks\Darren.Parker.
- Query the sys.login_token and sys.user_token system functions to check the available tokens for Darren Parker.
- Change your security context back using the REVERT command.

Results: After this exercise, you should have tested the role assignments for Darren Parker.

Module Review and Takeaways



Review Questions

1. Is it possible to create new database roles in SQL Server 2012?
2. Which function allows you to determine in T-SQL code whether or not a user is a member of a Windows group?

Best Practices

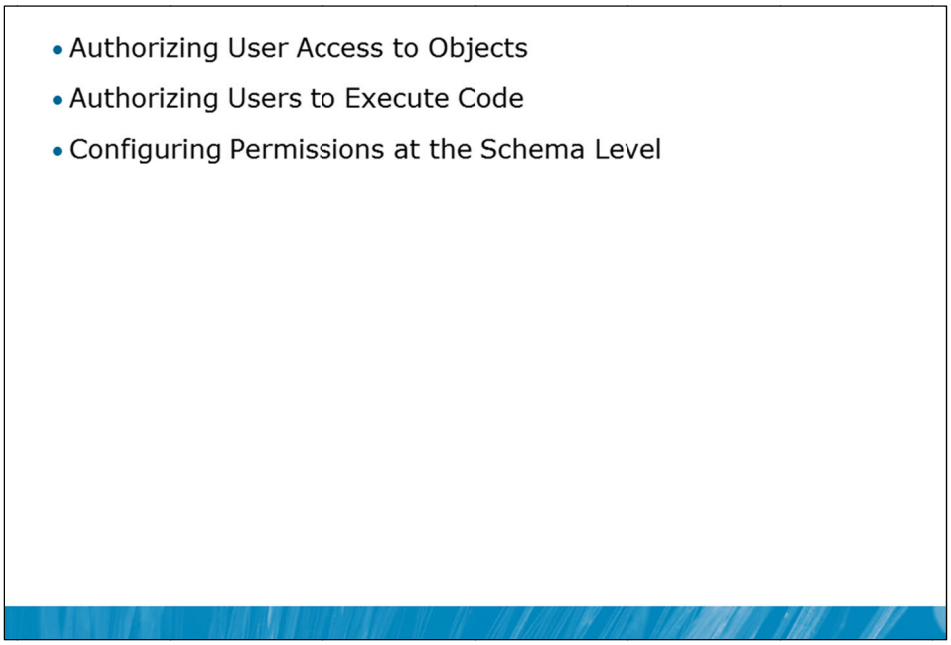
Avoid granting more permissions than are necessary. It is very common to see SQL Server systems where excessive permissions have been granted. Often the installers for applications will assume the need for much higher level permissions than should be necessary. Users should push back on vendors who do this. Even better, make appropriate security and permissions configuration a criterion for vendors to meet.

Module 11

Authorizing Users to Access Resources

Contents:

Module Overview



In the previous two modules, you have seen how Microsoft® SQL Server® security is organized and how sets of permissions can be assigned at the server and database level via fixed server roles, user-defined server roles, fixed database roles, user-defined database roles, and application roles.

The final step in authorizing users to access SQL Server resources is the authorization of users and roles to access server and database objects. In this module, you will see how these object permissions are managed. As well as access permissions on database objects, SQL Server provides the ability to determine which users are allowed to execute code, such as stored procedures and functions.

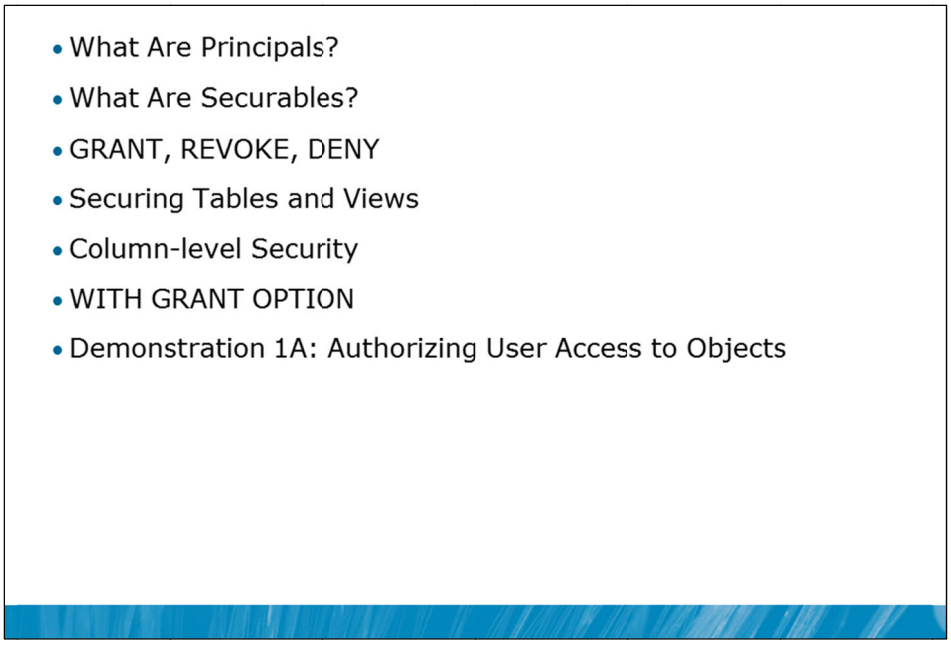
In many cases, these permissions and the permissions on the database objects are best configured at the schema level rather than at the level of the individual object. Schema-based permission grants can simplify your security architecture. You will explore the granting of permissions at the schema level in the final lesson in this module.

Objectives

After completing this lesson, you will be able to:

- Authorize user access to objects.
- Authorize users to execute code.
- Configure permissions at the schema level.

Lesson 1
Authorizing User Access to Objects



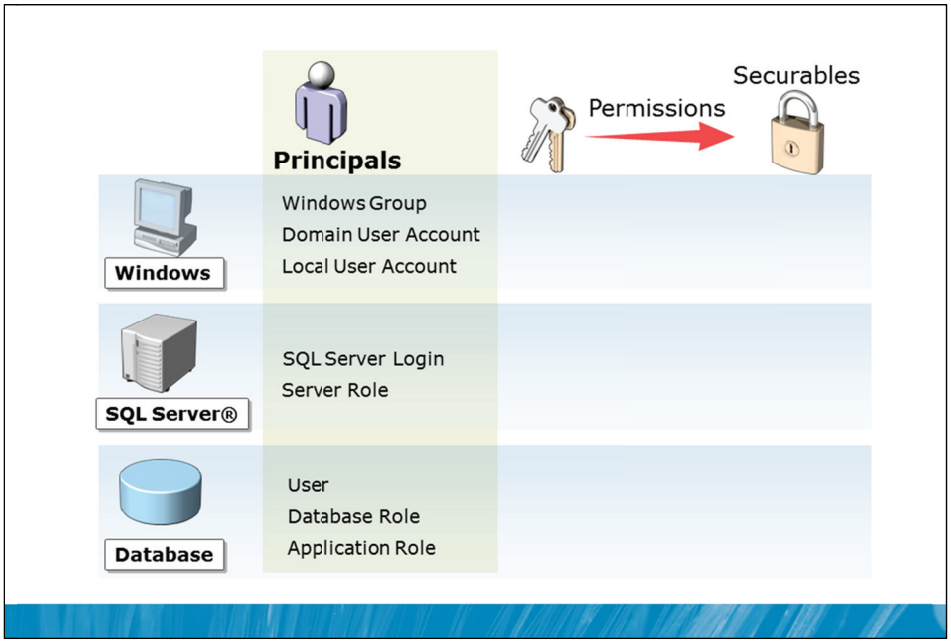
Before moving on to managing permissions on code, you need to consider how permissions are managed on database objects. SQL Server has a fine-grained security model that allows you to grant the minimum permissions to users that will allow them to do their work. In particular, permissions can be granted at the column-level, not just at the table and view level. You will also see how you can delegate the work of granting permissions to other users.

Objectives

After completing this lesson, you will be able to:

- Explain the role of principals.
- Explain the role of securables.
- Use the GRANT, REVOKE, and DENY commands.
- Secure tables and views.
- Implement column-level security.
- Delegate the ability to assign permissions by using WITH GRANT OPTION.

What Are Principals?



Key Points

In the previous two modules, you have seen a number of security principals.

Principals are entities that can request and can be granted access to SQL Server resources. Like other components of the SQL Server authorization model, principals can be arranged in a hierarchy. This slide summarizes the principals that have been discussed and places them into their appropriate locations within the hierarchy.

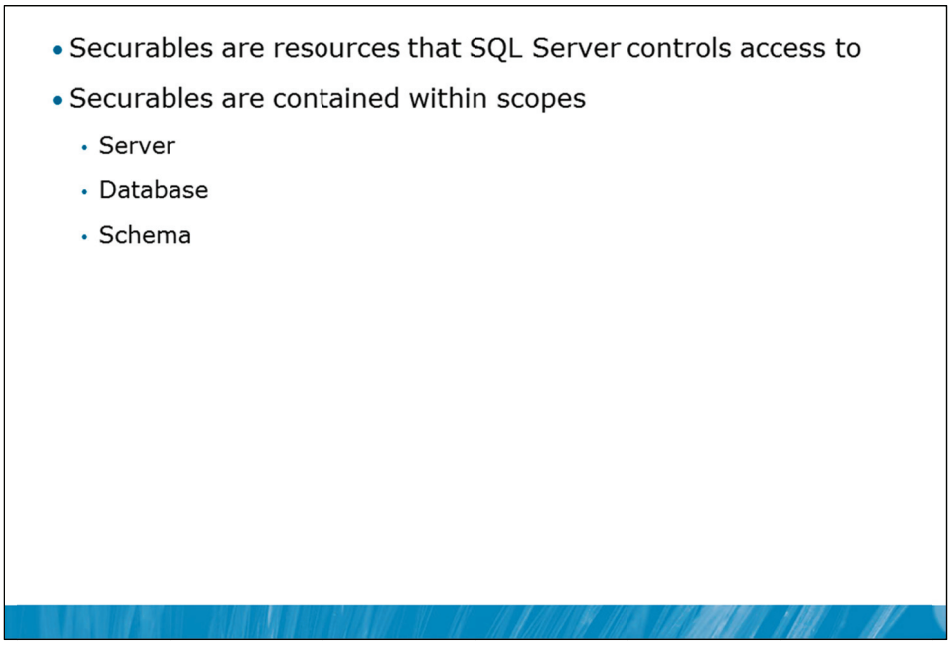
At the Windows® level, principals include users and groups. These users and groups can be domain-based if the server is part of a Windows domain. Local accounts can be used from servers, whether the server is a member of a domain or not.

At the SQL Server level, logins can be created for users that are either not Windows users or for users that are part of non-trusted Windows environments, such as users in other Windows domains where no trust relationship is in place with the domain containing the SQL Server system. Also at the SQL Server level, fixed and user-defined server roles are a form of principal that contains other principals.

At the database level, you have seen database users, fixed and user-defined database roles and application roles.

Every principal has two numeric IDs associated with it: a principal ID and a security identifier (SID).

What Are Securables?



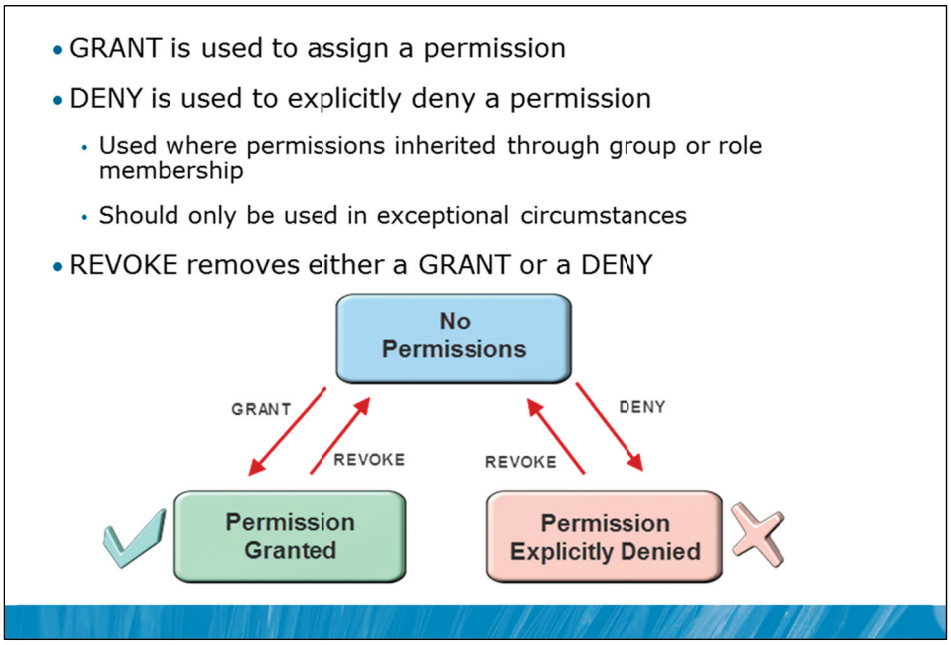
Key Points

Securables are the resources to which the SQL Server Database Engine authorization system regulates access. Some securables can be contained within others, creating nested hierarchies called scopes that can themselves be secured. The securable scopes are server, database, and schema.

It is important to understand the different securable scopes in SQL Server to plan your security model.

Question: Can you suggest a reason why a Login is a securable? What types of permissions would be needed on a Login?

GRANT, REVOKE, DENY



Key Points

Permissions are managed via the GRANT, DENY, and REVOKE T-SQL commands. Most permissions (but not all permissions) can also be managed via the GUI in SSMS.

GRANT and REVOKE

A user that has not been granted a permission is unable to perform the action related to the permission. For example, users have no permission to SELECT data from tables if they have not been granted permission at some level. Some other database engines consider this an implicit form of denial.

The GRANT command is used to assign permissions to database users. The REVOKE command is used to remove those same permissions.

DENY

ANSI SQL does not provide a DENY command. If a user does not have permission to perform an action, they cannot perform the action.

What is different about Windows-based systems is group membership. ANSI SQL has no concept of groups. In SQL Server, a Windows user can receive permissions directly but can also receive permissions through membership in Windows groups or receive permissions through membership in roles.

The DENY command allows you to deny a permission to a user that has been granted permission by membership in a group or role that has permission. This is very similar to how deny permissions work in Windows. For a Windows example, consider that you could decide that all members of the Salespeople group can access a Color printer, except Holly (who is a member of Salespeople) because she causes problems with it. You grant access to the Salespeople group then deny access to Holly.

SQL Server and DENY

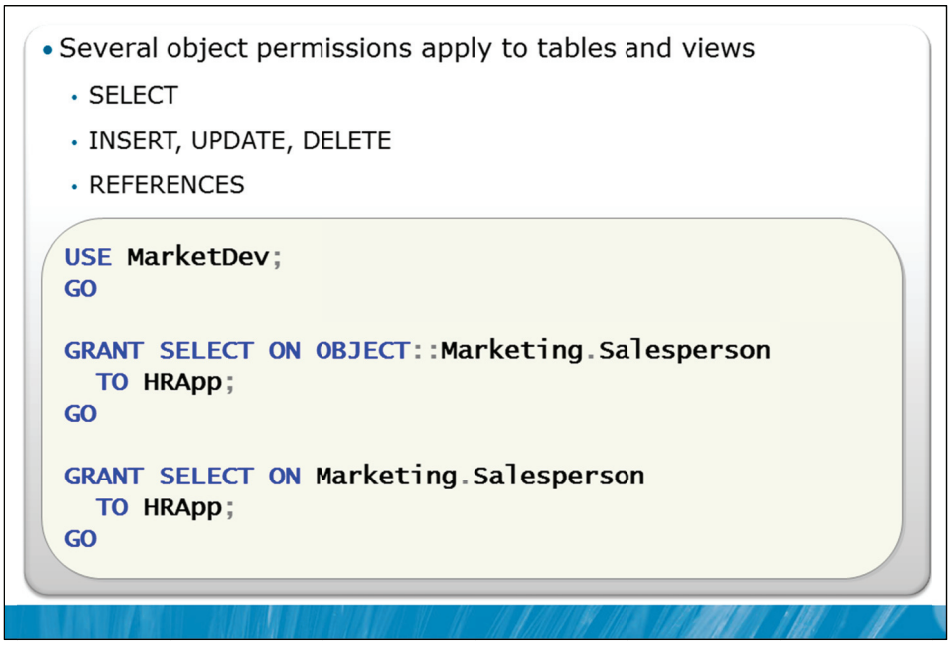
SQL Server works this same way. You could grant SELECT permission on a table to every Salesperson but deny Holly access to that table.

As with Windows, you should use DENY sparingly. A need to DENY many permissions tends to be considered a "code-smell" that indicates a potential problem with your security design.

What does tend to confuse new users is that the REVOKE command is also used to remove a DENY, not just to remove a GRANT. This means that it could cause a user to have access that they did not have before the REVOKE command was issued. For example, if you revoke the DENY permission from Holly, she would then be able to access the table.

Question: If a user cannot perform an action without permission, why is there any need to DENY a permission?

Securing Tables and Views



Key Points

The permissions to access data that apply to tables and views are SELECT, INSERT, UPDATE, DELETE, and REFERENCES.

In the example shown on the slide, SELECT permission on the Marketing.Salesperson object (which is likely to be a table or view) is being granted to the HRApp user within the MarketDev database.

Optional Components of GRANT

Note that two forms of the command are shown. While the full terminology involves OBJECT:: as a prefix, this prefix is optional. In the second example, the same GRANT is shown without the OBJECT:: prefix.

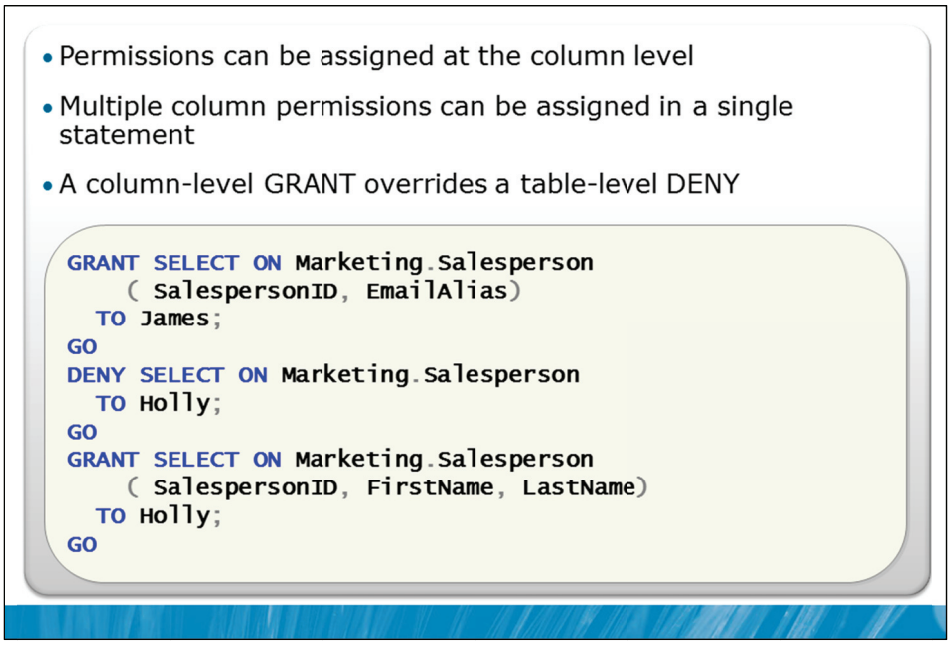
It is not necessary to specify the schema for the table or view but doing so is highly recommended. If the schema name is not specified, the default schema for the user that is granting the permission is used. If the object is not found in the user's default schema, the dbo schema is used instead.

REFERENCES

While the meaning of the SELECT, INSERT, UPDATE, and DELETE permissions will likely be obvious to you, the meaning and purpose of the REFERENCES permission might not be. The REFERENCES permission is necessary before a foreign key relationship can specify the object as a target, and is only required if no other permissions exist on the object.

> **Question:** Why would there be a need for a permission to refer a table in a foreign key reference?

Column-level Security



Key Points

While they are not implemented as often as table or view level permissions, column-level permissions can also be assigned. This provides an even finer grain of security control than is provided by controlling access to tables and views.

You do not need to execute separate GRANT statements for every column that you wish to assign permissions on. Where a set of columns needs to be controlled in the same way, a list of columns can be provided in a single GRANT statement. In the first example shown on the slide, SELECT permission on the Marketing.Salesperson table is being granted to James but access to the entire table is not being granted. Only permission to the SalespersonID and EmailAlias columns is permitted.

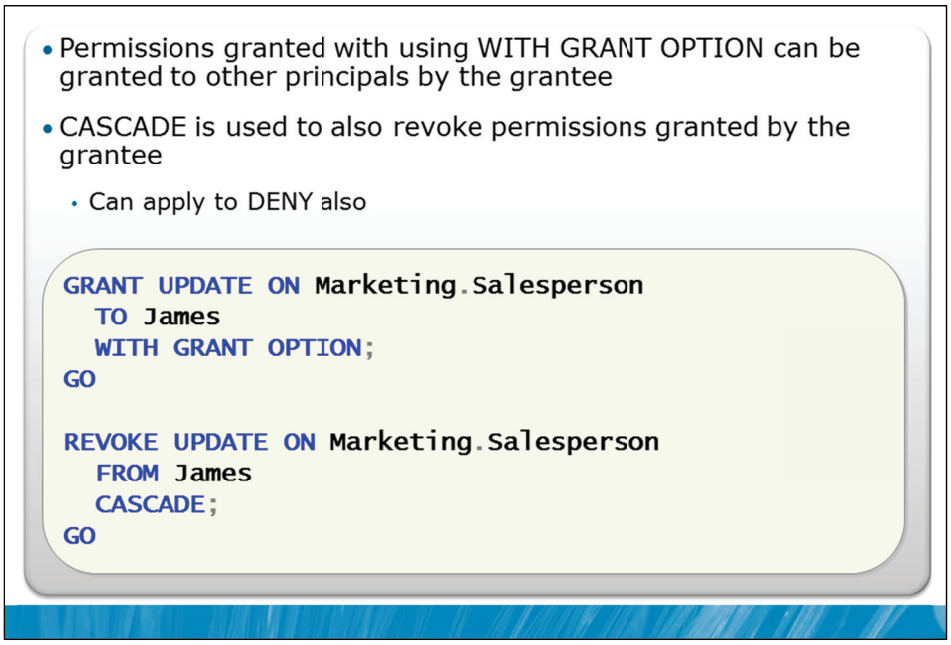
Table-level DENY and Column-level GRANT

There is an anomaly in the SQL Server security model that you need to be aware of.

A table-level DENY does not take precedence over a column-level GRANT. This is acknowledged as an inconsistency but needed to be preserved for backward compatibility. There is a plan to remove this inconsistency in the future. Do not depend upon it in new development work.

This anomaly is demonstrated in the second example on the slide. Holly is denied permission to SELECT from the Salesperson table but is then granted permission to SELECT specific columns in the table. The result (probably an unexpected result) is that Holly would still be permitted to SELECT from those columns in the table.

WITH GRANT OPTION



Key Points

When a principal is granted a permission, it is also possible to grant the principal the right to re-grant the permission to other principals. This further right is assigned by use of the WITH GRANT OPTION clause. This mechanism allows you to delegate responsibility for managing permissions but needs to be used with caution. In general, WITH GRANT OPTION should be avoided.

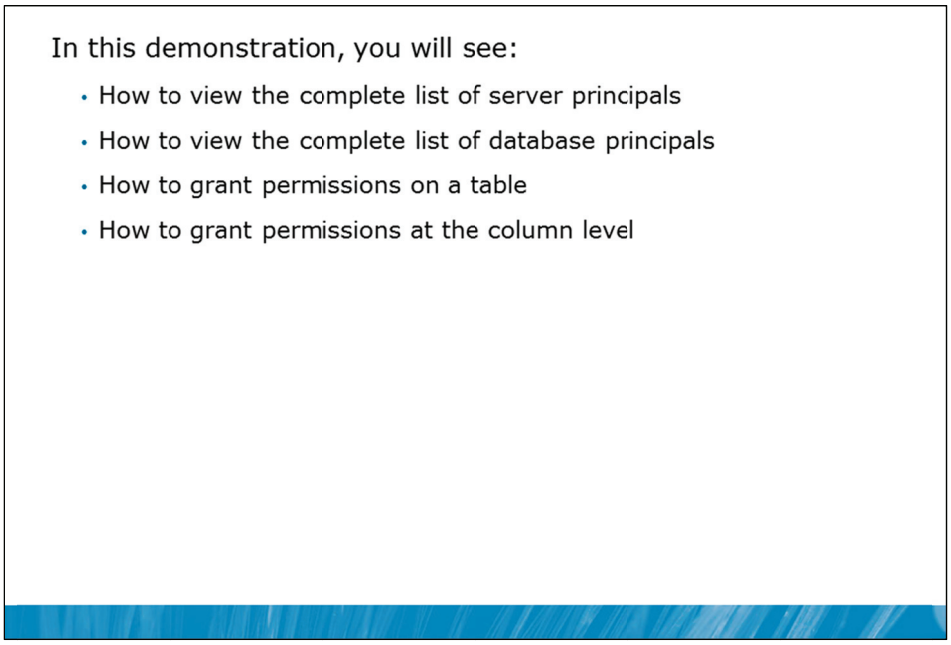
In the first example on the slide, James is granted permission to update the Marketing.Salesperson table. In addition, James is granted the right to grant this same permission to other database users.

CASCADE

The challenge of the WITH GRANT OPTION clause comes when you need to REVOKE or DENY the permission that was granted to James using WITH GRANT OPTION. You do not know which other users James has already granted the permission to.

When revoking or denying a permission that has been granted, the CASCADE option revokes or denies the permissions that James had granted as well.

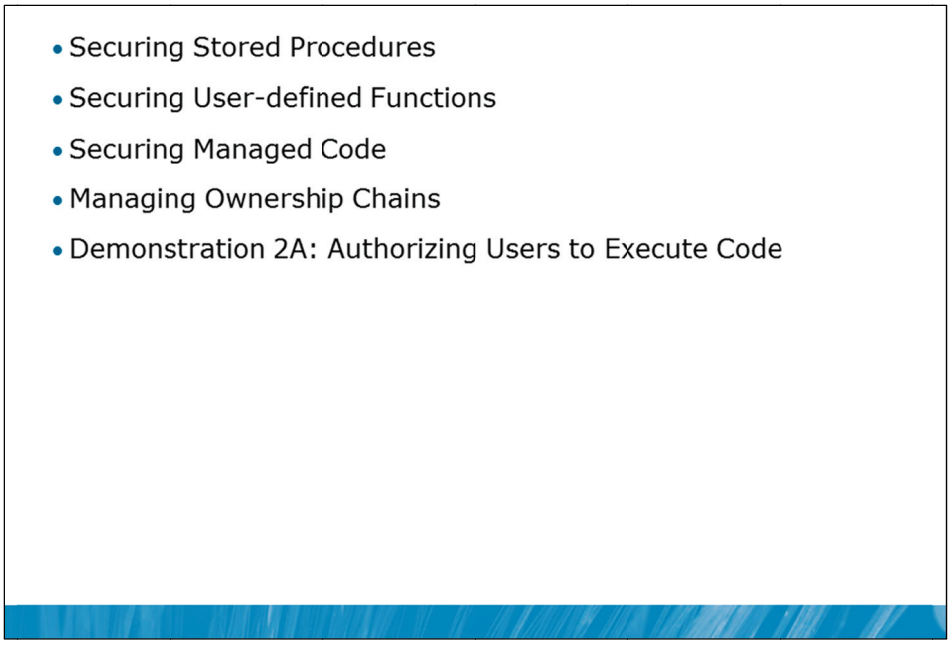
Demonstration 1A: Authorizing User Access to Objects



Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_11_PRJ\10775A_11_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 2
Authorizing Users to Execute Code



In addition to providing you with control over who accesses data in your database or the objects in your server, SQL Server allows you to control which users can execute your code. Appropriate security control of code execution is an important aspect of your security architecture.

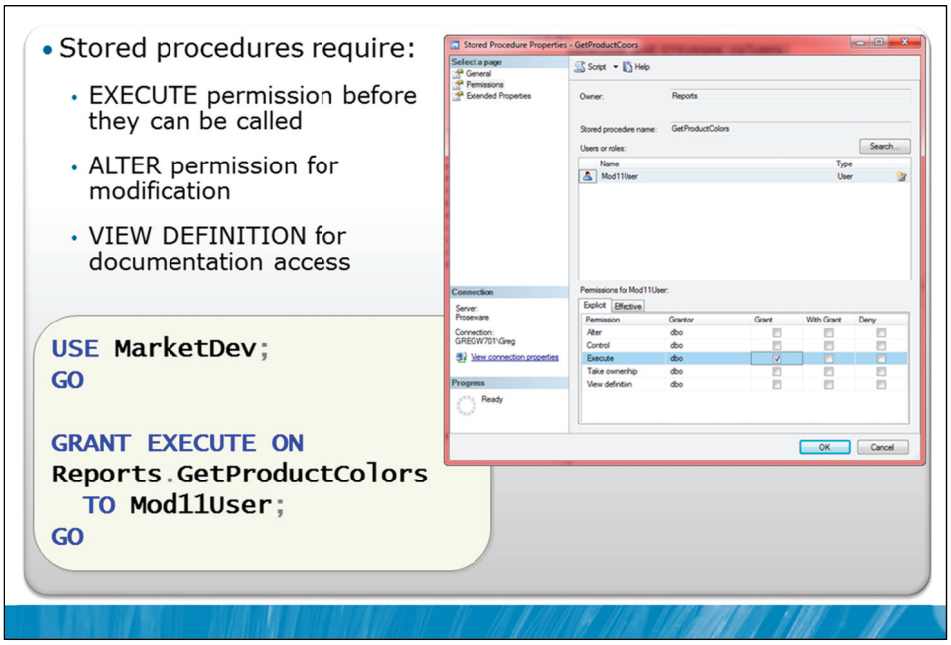
In this lesson, you will see how to manage the security of stored procedures and functions. You will also learn how to manage security for code that lives in .NET managed code assemblies that are used with SQL CLR integration. Finally, you will see how ownership chains affect the security relationship between code and database objects.

Objectives

After completing this lesson, you will be able to:

- Secure stored procedures.
- Secure user-defined functions.
- Secure managed code.
- Manage ownership chains.

Securing Stored Procedures



Key Points

By default, users cannot execute stored procedures that you (or any user) create. Users need EXECUTE permission before they can execute a stored procedure. They may also need permissions to access the objects that the stored procedure uses. You will see more about this issue later in the lesson.

In the example shown on the slide, the database user Mod11User is being granted the permission to execute the stored procedure Reports.GetProductColors.

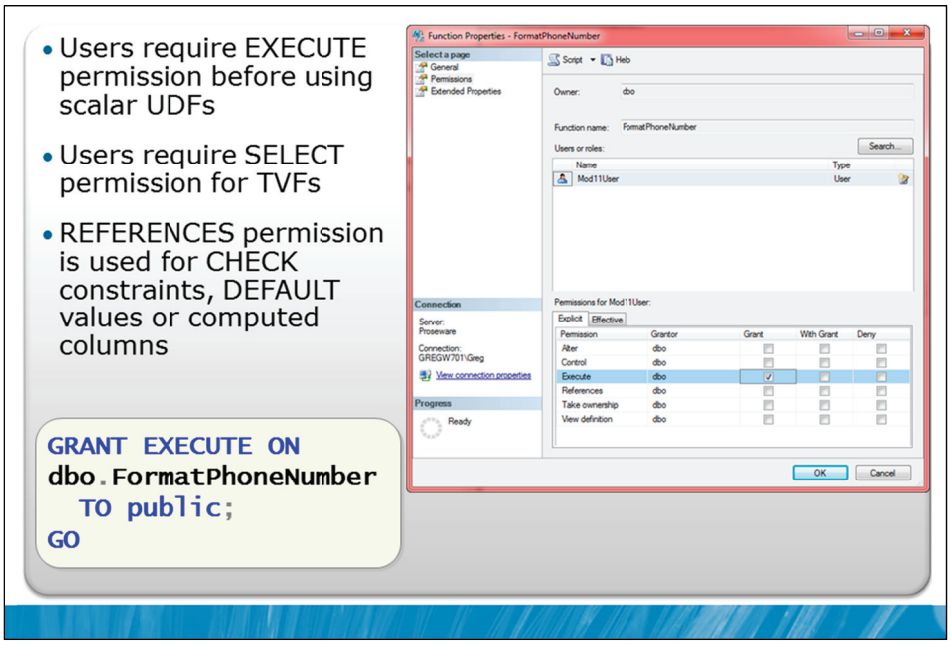
Managing Stored Procedures

Two other permissions are related to the management of stored procedures:

- The ALTER permission allows a user to change the definition of a stored procedure.
- The VIEW DEFINITION permission was added in SQL Server 2005. In earlier versions, a user needed ALTER permission on a stored procedure before they could view its definition. This represented an unnecessary permission grant for users involved in documenting systems. The VIEW DEFINITION permission was introduced to remove the need for such a high level permission when only documentation access was needed.

Note You cannot use SSMS to grant permissions on system stored procedures. SSMS can be used to grant permissions on other stored procedures.

Securing User-defined Functions



Key Points

User-defined functions (UDFs) also require permissions before they can be used.

- Scalar UDFs return a single value. Users accessing these functions require EXECUTE permission on the UDF.
- Table-valued UDFs (which are TVFs) return a table of results rather than a single value. Accessing a TVF requires SELECT permission rather than EXECUTE permission, similar to the permissions on a table.

While it is uncommon to directly update a table-valued function, it is possible to assign INSERT, UPDATE, and DELETE permissions on one form of TVF known as an inline TVF, as this particular form of TVF can be updated in some cases.

REFERENCES

REFERENCES permission is required for:

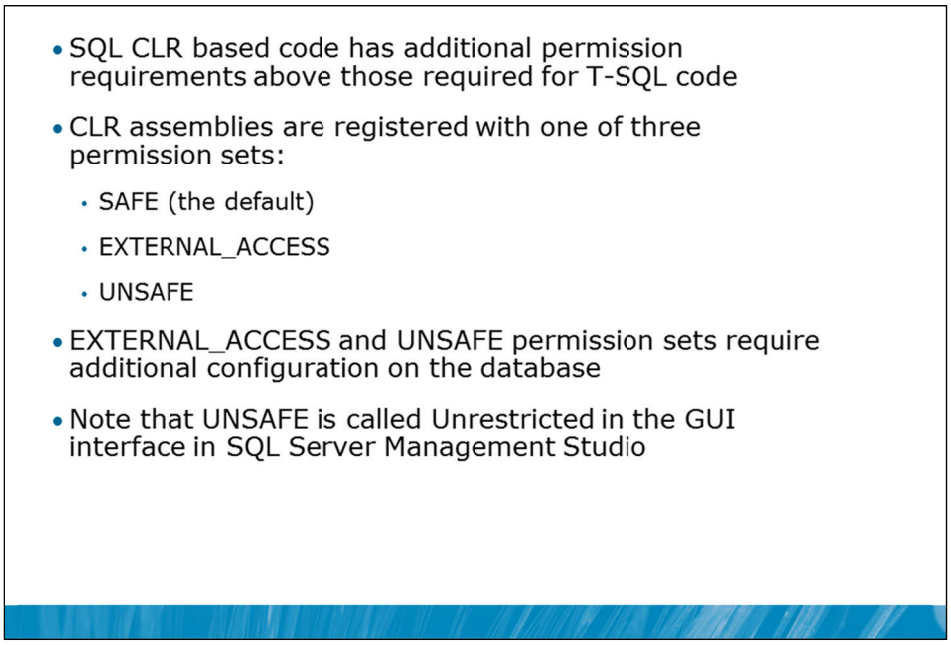
- Functions that are used in CHECK constraints.
- To calculate values for DEFAULT constraints.
- To calculate values for computed columns.

public Role and Functions

Functions often provide very basic capabilities within systems and with low risk. Because of this, it is fairly common practice to grant permissions on basic functions that are contained in a database to the public role of the database. This allows any user within the database to use those functions without the need for further permission grants.

Note While this is common for basic functions, this is rarely done (or even appropriate) for stored procedures.

Securing Managed Code



Key Points

Managed code is .NET code that is provided in assemblies. While assemblies are contained within DLL or EXE files, only assemblies contained within DLL files can be loaded into SQL Server via SQL Server CLR integration.

After an assembly is catalogued, procedures, functions, and other managed code objects that are contained within the assembly are also catalogued. These objects then appear as standard objects within SQL Server and the standard T-SQL object permissions also apply. For example, a user requires EXECUTE permission on a stored procedure, whether it is written in managed code or in T-SQL.

Permission Sets

No matter what .NET code is included in an assembly, the actions it is allowed to take are determined by the permission set it is catalogued under.

The SAFE permission set strictly limits the actions that the assembly can perform. It is the default permission set.

The EXTERNAL_ACCESS permission set is needed to access local and network resources, environment variables, and the registry. EXTERNAL_ACCESS is even necessary for accessing the same SQL Server instance if a connection is made through a network interface. This permission set is not necessary for direct internal connections from the managed code to SQL Server, as a separate direct access path (called a context connection) is provided for access to the local instance, without using a network interface.

The UNSAFE permission set relaxes many standard controls over code and should be avoided.

Recommendations

In general, SQL Server DBAs should find the use of SAFE assemblies to be acceptable; they should require some discussion before EXTERNAL_ACCESS assemblies are used; and they should need particularly solid justifications (which should be rare) for any UNSAFE assemblies. UNSAFE is the permission set with the most capabilities.

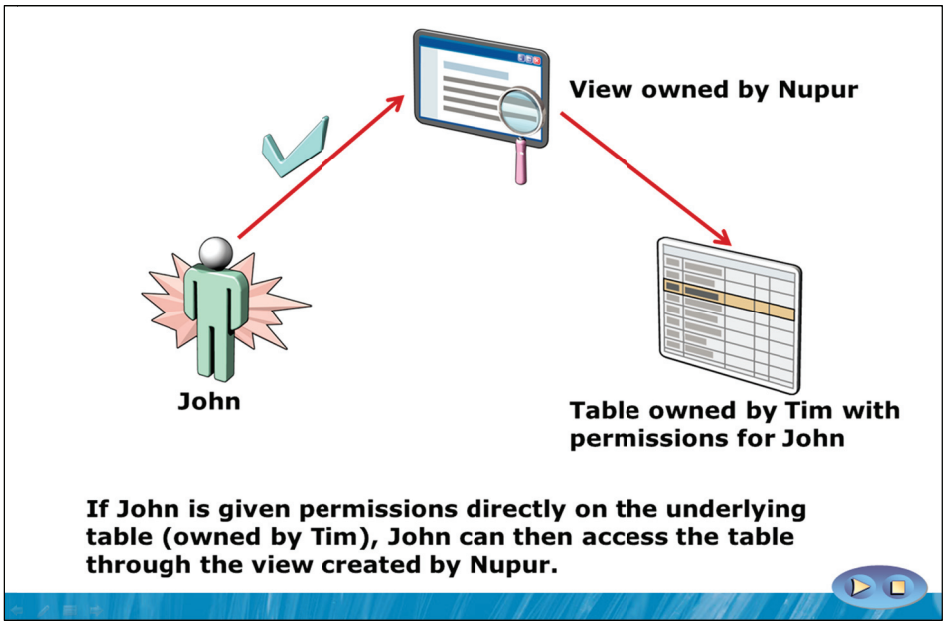
Configuration

The EXTERNAL_ACCESS and UNSAFE permission sets also require additional setup. You cannot specify the need for an EXTERNAL_ACCESS permission set when executing the CREATE ASSEMBLY statement. Either the database needs to be flagged as TRUSTWORTHY (which is easy but not recommended) or an asymmetric key needs to be created from the assembly file in the master database, a login created that maps to the key and the login granted EXTERNAL ACCESS ASSEMBLY permission on the assembly.

Note This last option is clearly an advanced topic that is beyond the scope of the course and is only mentioned for completeness.

Question: Which permission set should be rarely allowed?

Managing Ownership Chains



Key Points

All database objects have owners. Schema-scoped objects are owned by the schema owner and the principal_id (owner) property for new objects is NULL by default. An object with a NULL principal_id inherits its ownership from the schema it is contained in. The best practice is to have all objects owned by the schema object owner.

When an object such as a stored procedure references another object, an ownership chain is established. An unbroken ownership chain exists when each object in the chain has the same owner. When an unbroken ownership chain exists, access is permitted to the underlying objects when access is permitted to the top level objects.

Ever since SQL Server 2005 introduced the concept of schemas, it has been widely misconstrued that SQL Server objects no longer have owners. This is not true. Objects still have owners.

Having the same owner for all objects in a schema (which itself also has an owner) of a database keeps permission management easier but it is important to understand that ownership chain problems can still occur even though they are much less common now.

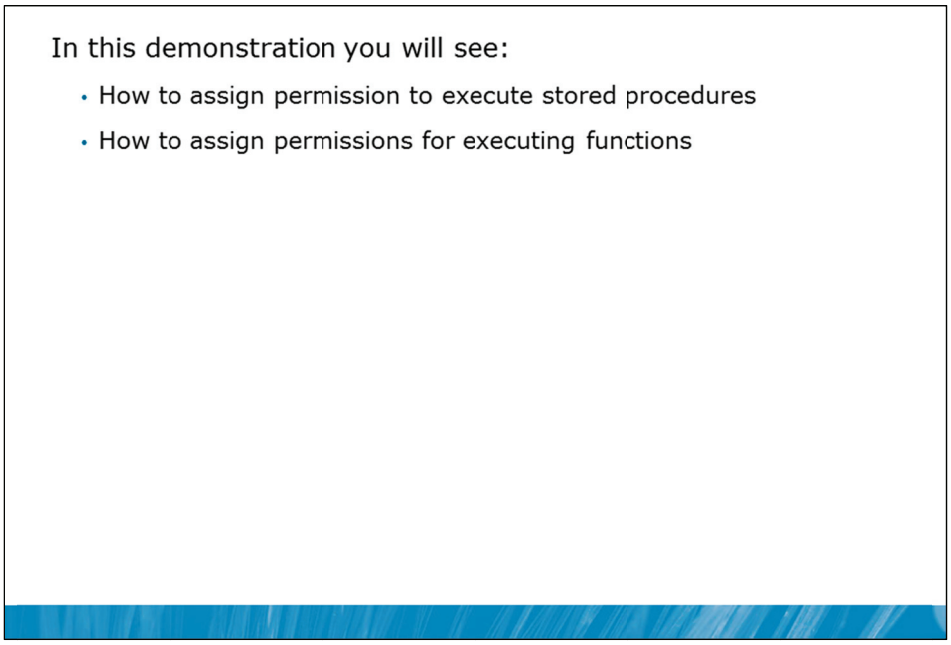
Slide Example

Ownership chaining applies to stored procedures, views, and functions. The slide shows an example of how ownership chaining applies to views or stored procedures.

1. John has no permissions on the table owned by Nupur.
2. Nupur creates a view that accesses the table and grants John permission to access the view. Access is granted as Nupur is the owner of both the top level object and of the underlying object (that is her table).
3. Nupur than creates a view that accesses a table that is owned by Tim. Even if Nupur has permission to access the table, and grants John permission to use the view, John will be denied access. This is because of the broken chain of ownership from the top level object to the underlying object.
4. However, if John is given permissions directly on the underlying table owned by Tim, he can then access the view that Nupur created to access that table.

The problem with step #4 is that one of the main reasons for creating views or stored procedures is to prevent the need for users to have permissions on the underlying objects.

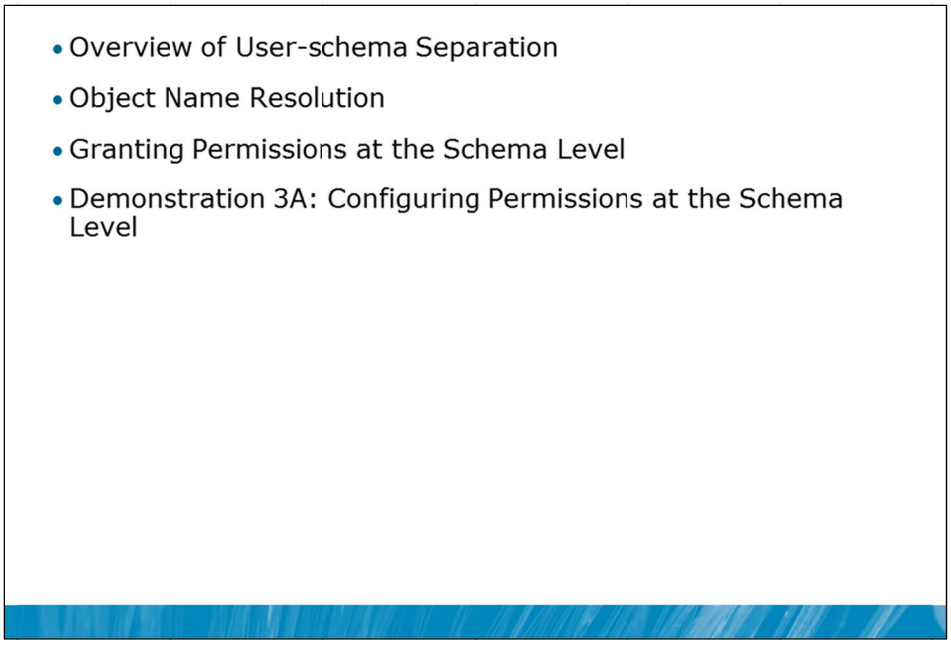
Demonstration 2A: Authorizing Users to Execute Code



Demonstration Steps

1. If Demonstration 1A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_11_PRJ\10775A_11_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **21 – Demonstration 2A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lesson 3
Configuring Permissions at the Schema Level



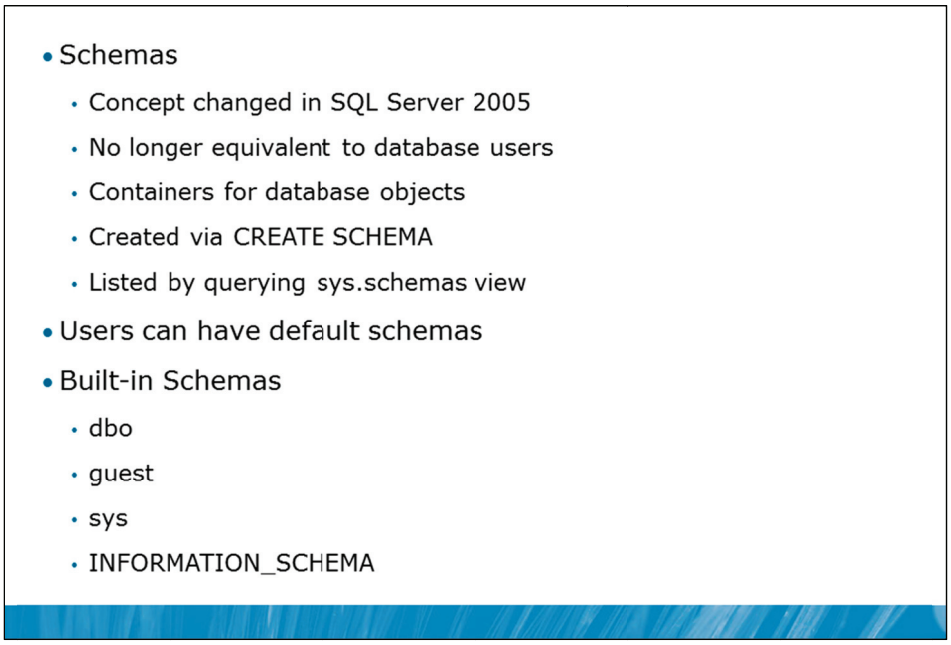
SQL Server 2005 introduced a change to how schemas are used. Since that version schemas are used as containers for objects such as tables, views, and stored procedures. Schemas can be particularly helpful in providing a level of organization and structure when large numbers of objects are present in a database. Security permissions can also be assigned at the schema level rather than individually on the objects contained within the schemas. Doing this can greatly simplify the design of system security requirements.

Objectives

After completing this lesson, you will be able to:

- Describe user-schema separation.
- Describe the role of object name resolution.
- Grant permissions at the schema level.

Overview of User-schema Separation



Key Points

Schemas are used to contain objects and to provide a security boundary for the assignment of permissions.

Schemas

In SQL Server, schemas are essentially used as containers for objects, somewhat like a folder is used to hold files at the operating system level. Since their change of behavior in SQL Server 2005, schemas can be used to contain objects such as tables, stored procedures, functions, types, views, etc. Schemas are created with the CREATE SCHEMA statement and schemas form a part of the multi-part naming convention for objects. In SQL Server, an object is formally referred to by:

```
Server.Database.Schema.Object
```

Security Boundary

Schemas can be used to simplify the assignment of permissions. An example of applying permissions at the schema level would be to assign the EXECUTE permission on a schema to a user. The user could then execute all stored procedures within the schema. This simplifies the granting of permissions as there is no need to set up individual permissions on each stored procedure.

Upgrading Older Applications

If you are upgrading applications from SQL Server 2000 and earlier versions, it is important to understand that the naming convention changed when schemas were introduced. Previously, names were of the form:

```
Server.Database.Owner.Object
```

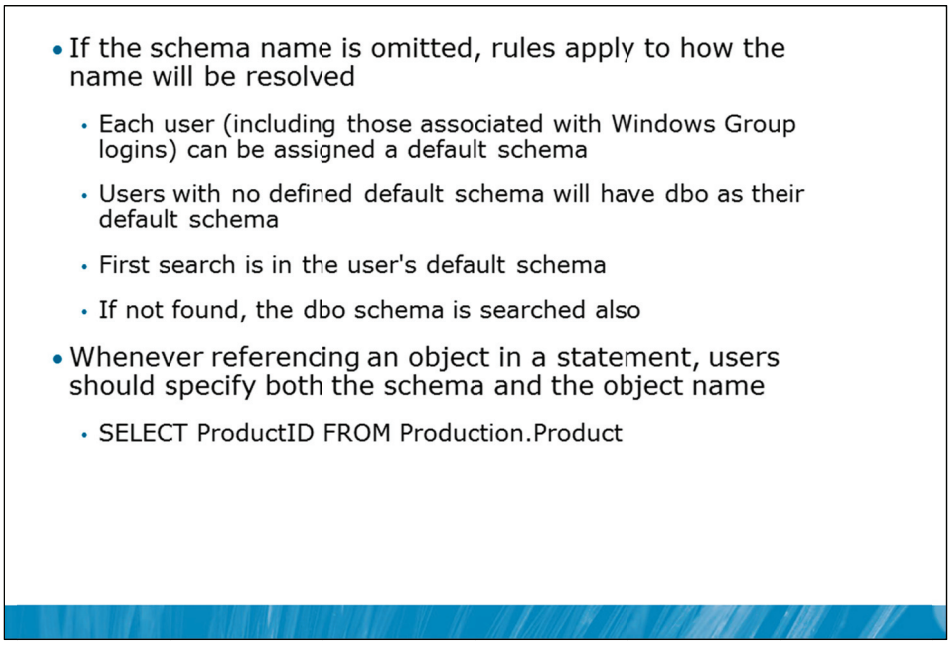
Objects still have owners but the owner's name does not form a part of the multi-part naming convention from SQL Server 2005 onwards. When upgrading databases from earlier versions, SQL Server will automatically create a schema with the same name as existing object owners, so that applications that use multi-part names will continue to work.

Each user can be assigned a default schema that is used when a user refers to an object without specifying a schema name.

Built-in Schemas

dbo and guest have been discussed in the last module. dbo has an associated schema. The sys and INFORMATION_SCHEMA schemas are reserved for system objects. You cannot create objects in the sys and INFORMATION_SCHEMA schemas and you cannot drop those schemas.

Object Name Resolution



Key Points

It is important to use at least two-part names when referring to objects in SQL Server code such as stored procedures, functions, and views.

Object Name Resolution

When object names are referred to in the code, SQL Server must determine which underlying objects are being referred to. For example, consider the following statement:

```
SELECT ProductID, Name, Size FROM Product;
```

More than one Product table could exist in separate schemas with the database. When single part names are used, SQL Server must then determine which Product table is being referred to.

Most users have default schemas assigned but not all users are assigned a default schema. Default schemas are not assigned to users based on certificates but they do apply to users created from standard Windows and SQL Server logins. SQL Server 2012 introduced the ability to assign a default schema to a Windows Group. Users without default schemas assigned to them have the dbo schema as their default schema.

Note Users created from certificates is an advanced topic that is out of scope for this course but mentioned for completeness.

Locating Objects

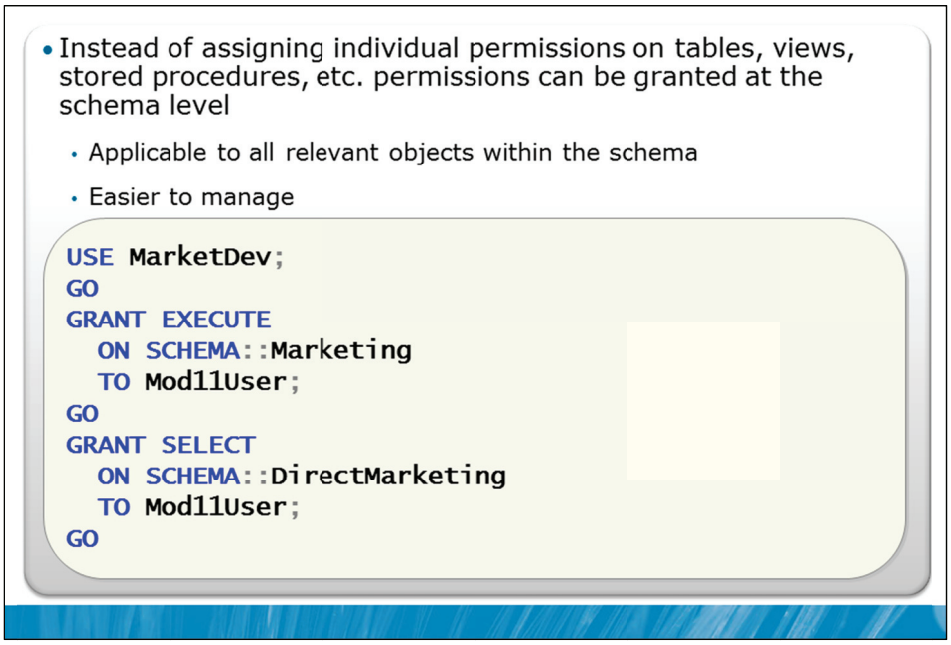
When locating an object, SQL Server will first check the user's default schema. If the object is not found, SQL Server will then check the dbo schema to try to locate the object.

It is important to include schema names when referring to objects instead of depending upon schema name resolution, such as in this modified version of the previous statement:

```
SELECT ProductID, Name, Size FROM Production.Product;
```

Apart from rare situations, using multi-part names leads to more reliable code that does not depend upon default schema settings.

Granting Permissions at the Schema Level



Key Points

Instead of assigning individual permissions on tables, views, and stored procedures, permissions can be granted at the schema level.

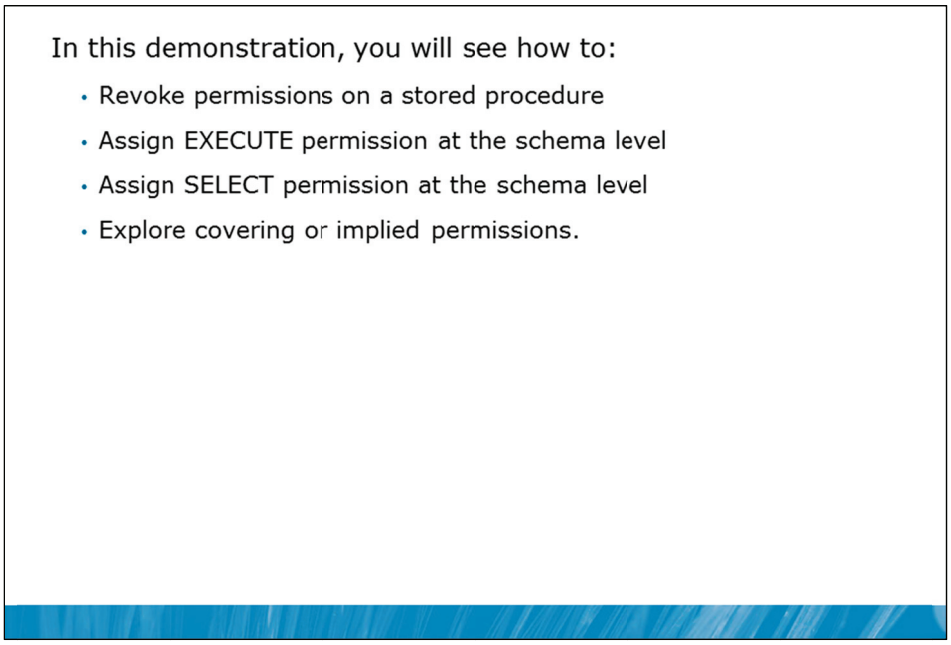
Slide Example

In the first example on the slide, EXECUTE permission on the Marketing schema is granted to Mod11User. This means that Mod11User could then execute all stored procedures and scalar functions within the schema.

In the second example on the slide, SELECT permission on the DirectMarketing schema is granted to Mod11User. This means that Mod11User could then select from all tables, views, and table-valued functions in the schema.

Question: Why would granting permissions at the schema level be easier to manage?

Demonstration 3A: Configuring Permissions at the Schema Level

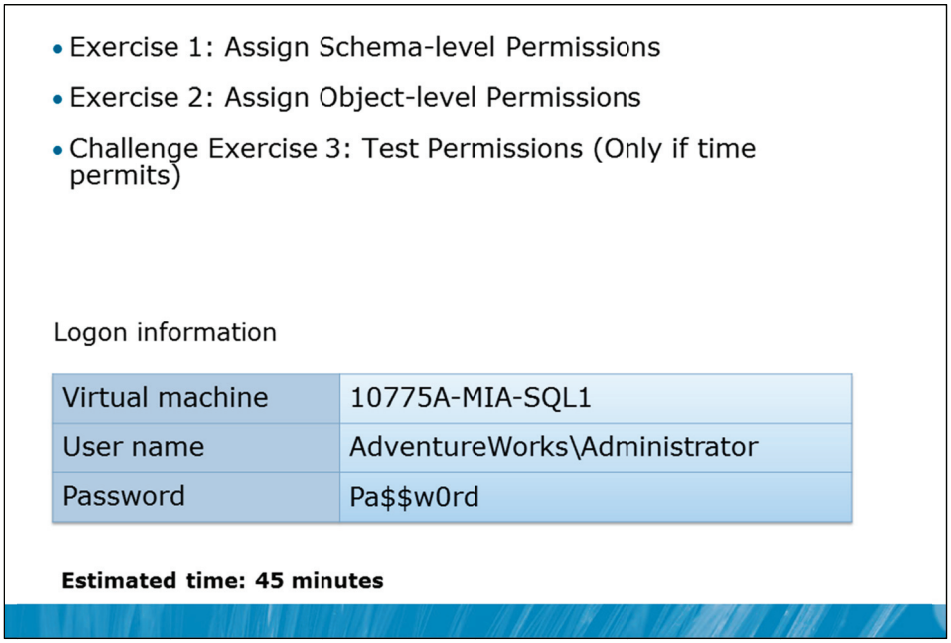


Demonstration Steps

1. If Demonstration 1A or 2A was not performed:
   - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
   - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_11_PRJ\10775A_11_PRJ.ssmssln** and click **Open**.
   - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** and **21 – Demonstration 2A.sql** script files from within Solution Explorer.
2. Open the **31 – Demonstration 3A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

   Question: The user has EXECUTE at the schema level and DENY at the procedure level. Should execution be permitted?

Lab 11: Authorizing Users to Access Resources



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_11_PRJ\10775A_11_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

You have created the SQL Server logins and Database users and assigned them to appropriate roles. You now need to grant permissions to the database users and roles so that users can access the resources they need within the MarketDev database, based on the supplied security requirements.

Supporting Documentation

Existing Windows User and Group Structure

	ITSupport	SalesPeople	CreditManagement	HumanResources	CorporateManagers
David.Alexander	X				X
Jeff.Hay	X				
Palle.Petersen	X				
Terry.Adams	X				
Darren.Parker		X			X
Mike.Ray		X			
April.Reagan		X			
Jamie.Reding		X			
Darcy.Jayne		X			
Naoki.Sato		X			
Bjorn.Rettig			X		X
Don.Richardson			X		
Wendy.Kahn			X		
Neil.Black				X	X
Madeleine.Kelly				X	

Pre-existing Security Configuration

- The following Windows group logins and database users have been created:
  - AdventureWorks\ITSupport
  - AdventureWorks\SalesPeople
  - AdventureWorks\CreditManagement
  - AdventureWorks\HumanResources
  - AdventureWorks\CorporateManagers
- The following Windows logins and database users have been created:
  - AdventureWorks\Jeff.Hay
  - AdventureWorks\April.Reagan
  - AdventureWorks\Darren.Parker
- The following SQL logins have been created:
  - PromoteApp
  - DBMonitorApp
- The following server role assignment has been made:
  - Jeff Hay → sysadmin
- The following fixed database roles member assignments have been made:
  - AdventureWorks\ITSupport → db_owner
  - DBMonitorApp → db_datareader
- The following user-defined database roles member assignments have been made:
  - AdventureWorks\SalesPeople → SalesTeam
  - AdventureWorks\CreditManagement → SalesTeam
  - AdventureWorks\CorporateManagers → SalesTeam
  - AdventureWorks\Darren.Parker → SalesManagers

Security Requirements

1. The senior DBA Jeff Hay should have full access to and control of the entire Proseware server instance.
2. All ITSupport group members should have full access to and control of the MarketDev database.
3. Proseware uses an application called DBMonitor from Trey Research. This application requires a SQL login called DBMonitorApp, which requires the ability to read but not update all objects in the MarketDev database.
4. All CorporateManagers group members perform periodic Strength, Weakness, Opportunity, and Threat (SWOT) analysis. For this they need to be able to both read and update rows in the DirectMarketing.Competitor table.

5. All SalesPeople group members should be able to read data from all tables in the DirectMarketing schema, except April Reagan who is a junior work experience student.
6. Only ITSupport group members and members of the CreditManagement group should be able to update the Marketing.CampaignBalance table directly.
7. Within the company members of the SalesPeople group, the CreditManagement group, and the CorporateManagers group are referred to as sales team members.
8. All sales team members should be able to read rows in the Marketing.CampaignBalance table.
9. All sales team members should be able to read rows in the DirectMarketing.Competitor table.
10. The Sales Manager should be able to read and update the Marketing.SalesTerritory table.
11. All HumanResources group members should be able to read and update rows in the Marketing.SalesPerson table.
12. The Sales Manager should be able to execute the Marketing.MoveCampaignBalance stored procedure.
13. All sales team members should be able to execute all stored procedures in the DirectMarketing schema.

Exercise 1: Assign Schema-level Permissions

Scenario

You have been supplied with a list of security requirements. Some of these requirements can be met using permissions assigned at the schema level. Even though it is easy to grant substantial permissions at the schema level, you should be careful to only grant permissions that are required.

The main tasks for this exercise are as follows:

1. Review the security requirements that have been updated from the previous module.
2. Assign the required permissions.

▶ Task 1: Review the security requirements that have been updated from the previous module

- Review the supplied requirements in the supporting documentation for the exercise.
- Determine the permissions that should be assigned at the schema level.

▶ Task 2: Assign the required permissions

- Assign the required permissions at the schema level.

Results: After this exercise, you should have assigned the required schema-level permissions.

Exercise 2: Assign Object-level Permissions

Scenario

You have been supplied with a list of security requirements. Some of these requirements need to be met using permissions assigned at the object level. In this exercise you will assign the required level object permissions.

The main tasks for this exercise are as follows:

1. Review the security requirements.
2. Assign the required permissions.

▶ Task 1: Review the security requirements

- Review the supplied requirements in the supporting documentation for the exercise.
- Determine the permissions that should be assigned at the object level. This would include permissions on tables, views, stored procedures, and functions where required.

▶ Task : Assign the required permissions

- Assign the required permissions at the object level.

Results: After this exercise, you should have assigned the required object-level permissions.

Challenge Exercise 3: Test Permissions (Only if time permits)

Scenario

You need to test some of your permission assignments. In particular you need to test that salespeople can select rows from the Marketing.CampaignBalance table. However you will also need to test that the work experience student April Reagan cannot select rows from that table even though she is a member of the SalesPeople group.

The main task for this exercise is as follows:

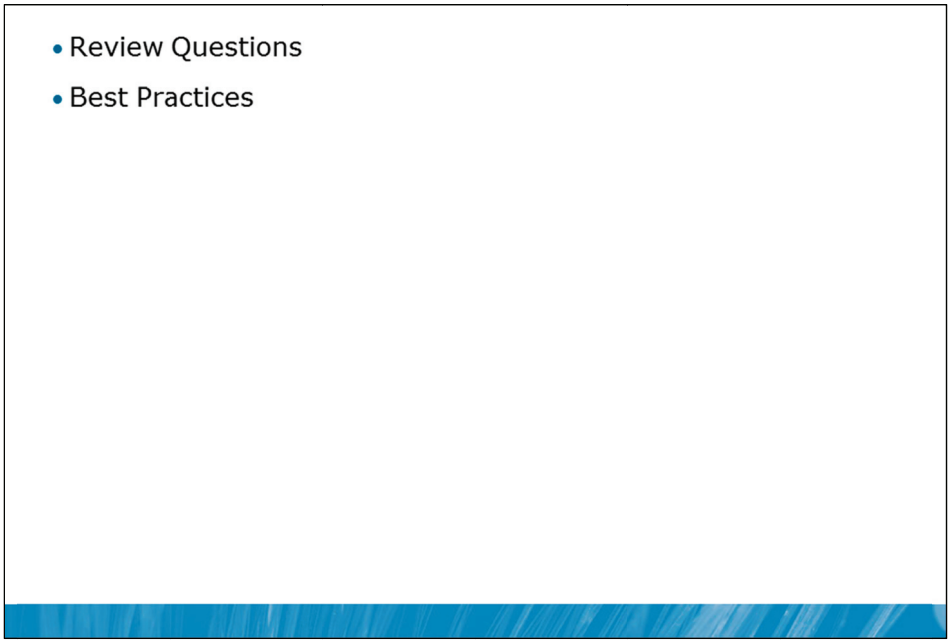
1. Design and execute a test.

▶ Task 1: Design and execute a test

- Design and execute a test to show that Darcy Jayne can select rows from the Marketing.CampaignBalance table.
- Design and execute a test to show that April Reagan cannot select rows from the Marketing.CampaignBalance table.

Results: After this exercise, you should have tested the required permissions.

Module Review and Takeaways



Review Questions

1. What permission needs to be assigned to a function before it can be used in a CHECK constraint?
2. What permission should be assigned to a schema to allow a user to read the data in all the tables, views and table-valued functions?

Best Practices

1. Always assign the least possible privileges that users need.
2. Test code as a standard user instead of testing as an administrator.
3. Use EXECUTE AS and REVERT for quick testing of user permissions.