

LogicalCHOICE

Facilitator's Guide

LogicalCHOICE

Facilitator's Guide

Welcome and Introduction.....	3
You Have Choices.....	4
Case Study: The Connected Classroom.....	5
Making the Transition to the Connected Classroom.....	9
Configuring the Learning Environment.....	10
The LogicalCHOICE Experience.....	18
Delivering a LogicalCHOICE Course.....	21
Presentation Best Practices.....	24
Conclusion: Making the Transition to Virtual	28

Welcome and Introduction

Time and distance are no longer the barriers to delivering training that they once were. Social networks, collaboration tools, and videoconferencing technologies enable you to overcome such barriers, so you can deliver training that meets the needs of learners, facilitators, and your organization. However, teaching in an environment that uses these technologies provides new challenges and requires that facilitators and learners get accustomed to new ways of interacting.

The Logical Operations LogicalCHOICE product line recognizes these new challenges and provides support for teaching and learning in such an environment as well as in the traditional classroom. This guide will take a look at special considerations of teaching in the twenty-first century classroom. It will introduce to the LogicalCHOICE product and provide guidelines to help you successfully deliver courses in the new paradigm.

You Have Choices

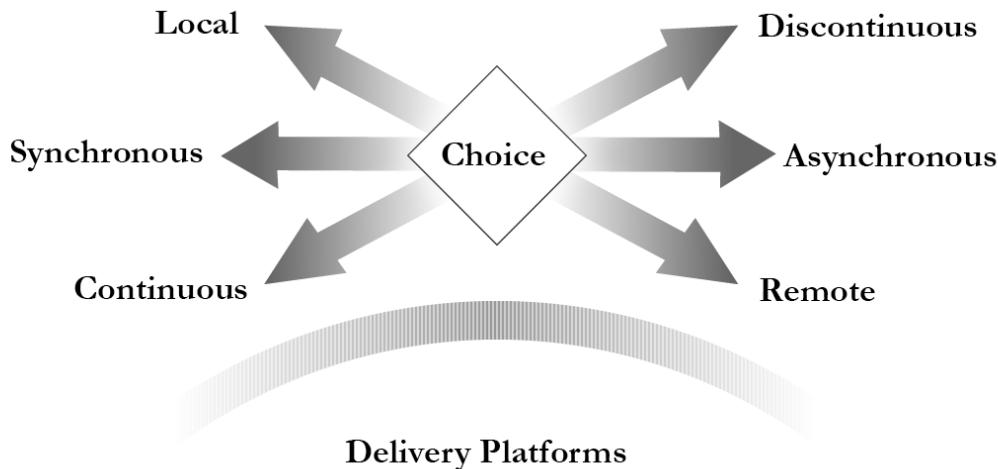


Figure 1: Choice across multiple dimensions of delivery

When you plan a collaborative learning experience, you make decisions along various dimensions that define the collaborative learning experience. With today's conferencing and collaboration technologies, you have more options than ever before in regard to the logistics of delivery, including:

- **Presence** (*local versus remote*): Learners may be present within the same classroom, or they may be virtual—connected through videoconferencing technologies such as teleconferencing, messaging, and computer desktop sharing.
- **Synchronicity** (*synchronous versus asynchronous*): Learners may work through the course at the same time, or they may work through parts of the course on their own, at different times.
- **Continuity** (*continuous versus discontinuous*): Learning may occur over a single span, such as an eight-hour training day. Or it may occur in installments over multiple periods of time.
- **Delivery Platforms:** An array of technologies is available to support the delivery of training across the other dimensions of learning.

You may need to cover a combination of these dimensions. For example, while you may have some learners present in the classroom, you may have others simultaneously connect to the classroom through videoconferencing technologies. They interact with you and other learners through audio and video connections provided through videoconferencing tools.

Case Study: The Connected Classroom

Twelve students have just settled in for a hands-on introductory course on Microsoft® Excel®. At first glance, the classroom looks like any other. Students are seated at computer workstations with Excel displayed on their screens. Each student has a training manual. The instructor writes on a whiteboard and discusses points highlighted on a Microsoft® PowerPoint® slide. But something is different about this classroom. It's a LogicalCHOICE classroom.



Figure 2: LogicalCHOICE facilitator workstation and classroom layout

Students Can Choose a Location

Seven of the 12 students are present and seated in the classroom, but five students aren't physically present in the classroom at all. They're in remote locations, accessing the classroom through video-conferencing tools. In the classroom, a video feed of each student is displayed on one of the large monitors mounted on the classroom wall, with each student's name displayed above his or her video feed. Some of the remote students, it seems, are seated at kitchen tables or in home offices. Another student is in an office cubicle setting and is wearing a headset so his audio feed doesn't disturb his cubicle neighbors. Some students have print manuals, and others are viewing their manuals on ebook readers and tablets.

The Instructor Interacts with Local and Remote Learners

The classroom is extended through the connections to the outside world. The technology, course materials, and interactions are designed so the instructor can interact with local and remote learners on an equal basis.

The classroom is arranged so the facilitator can see students' computer screens and move freely about the room. At times the instructor teaches from the whiteboard. Other times, the instructor teaches from her workstation near the back of the room.

The instructor—let's call her the *course facilitator*—actually has several monitors attached to her presentation computer. One is the monitor that displays the student video feeds in the classroom.

Another monitor is displayed through an interactive display in front of the room. The facilitator

touches the screen to interact with it as though using a mouse on a giant computer monitor. The facilitator highlights and marks up portions of the screen so that all students, local and remote, can see the markup. This is what she uses for most of her presentations.

The facilitator can also use the interactive display through a special wireless tablet and pen, which she carries with her as she moves throughout the room. Sometimes she works directly at the interactive whiteboard. Other times, she presents from her workstation. Because she has complete control from any location, she isn't bound to the front of the room or to a facilitator station. She can freely move about the room and interact with both her local and remote learners. She wears a wireless microphone so her voice is always audible.

One monitor at the facilitator's workstation is set up for her private viewing. She uses this monitor for private side chats with learners to monitor their progress and to answer questions. From this monitor, she also views and retrieves her teaching notes, and opens files and launches presentations and external resources that she will incorporate into the course.

Remote and Local Students Interact With Each Other

LogicalCHOICE courses are designed to promote activity and collaboration among learners, regardless of learner location. In our example class, remote students have web cams mounted on their monitors, which give them a visual presence in the class. And learners in the local classroom are logged in to the same web conferencing tools that are being used by the remote learners, even though they are on-site. This levels the playing field, enabling on-site learners to interact with the remote learners through collaboration tools such as online messaging and polling tools. There are two additional cameras in the room, which the facilitator can use as needed to provide remote learners with a wide view of activities in the classroom.

When the class began, the facilitator zoomed up each student's video feed in turn, as students introduced themselves to the class and shared their expectations for the course. The facilitator's role in this environment is much expanded over that of *instructor*. The facilitator is constantly aware of *delivery* – attending to logistical details that influence the learning experience for both local and remote students. She adjusts the online presentation as needed to create the best possible experience. That includes making sure all students have an opportunity to see what they need to see, hear what they need to hear, say what they need to say, and do what they need to do.

The Facilitator Fosters a Productive Learning Environment

As the facilitator leads students through course activities, the local and remote students follow along in their LogicalCHOICE course manuals. Some students have chosen to use the tablet or e-book version of the manual. Others have chosen to use a print version.

As in the traditional local classroom, the facilitator promotes *interaction*, helps students to stay *engaged*, and *monitors* learners' progress, making course adjustments as needed. However, some of this must now be done in a mixed environment or from a distance. As in the traditional classroom, some training organizations may choose to split this responsibility among co-instructors. Or depending on class size and available resources, they may provide technical resources and support to enable one facilitator to handle it all.

When you're teaching across a distance, monitoring learner progress can be a challenge. For exam-

ple, when the facilitator presents a new topic, she may begin with slides that introduce underlying concepts and principles. She then demonstrates the concept and associated procedures within the software application. Up to this point, there may be questions and dialogue, which are conducted through audio/video feeds supported by a web conferencing application. She might even use integrated polling and questioning tools to deliver a “pop quiz” to see if students are grasping the new material.

To attend to learner progress, when the facilitator releases learners to perform the activity on their own, she asks learners to click a button in the web conferencing tool to indicate when they have completed the activity, and she asks them to send her a private chat message if they are stuck or have any questions.

Learners Need Support in Hands-on Activities

One student has a question about something odd happening on his computer as he performs the activity. He shares his screen with her privately. The facilitator figures out the problem, but thinks it might actually provide an interesting learning opportunity for the entire class. She asks the student if he minds if she shares his screen with the whole class. He approves, so he shares his screen, and the facilitator guides the class through a troubleshooting exercise. Together they come up with the solution, and everyone learns from the experience.

Challenges and Opportunities of the Connected Classroom

In summary, training delivery is complicated enough. Add to that the complexity of teaching from a distance, and it's clear that the facilitator has a lot to tend to, including challenges such as:

- Communicating effectively.
- Promoting collaboration and interaction among participants.
- Fostering engagement.
- Monitoring learner performance; adjusting pacing and delivery as needed.
- Juggling among many concerns: content delivery, technology, learners with different needs, training logistics, and so forth.

However, with these challenges come new opportunities to extend the reach of your training, engage users in new ways, and incorporate new technology and new learning components to re-invent the traditional classroom learning experience. Let's see how you can make the transition to LogicalCHOICE.

Making the Transition to the Connected Classroom

As you start implementing training across the dimensions of delivery, you will find that your old definitions and ways of doing things start to change. The words *instructor*, *teacher*, and *trainer* may no longer adequately describe the role of the course leader. You may also find that you no longer need an instructor station at the front of the room; in fact, with such a connected classroom, the person who leads the course could be located *anywhere*—even offsite.

The role of instructor in the new paradigm tends to be far more than one who instructs. In the connected classroom, the course leader not only *delivers instruction* but plays the more comprehensive role of *facilitating learning*. An expert facilitator is adept at not only delivering instruction, but in facilitating the types of activities in which students learn together.

Conversely, students play a more active role than simply receiving the lecture. The design of the connected classroom helps to create an effective environment for collaboration and learning.

LogicalCHOICE training materials have been designed to support you in making the transition. They are designed with features to help you be more effective in a connected classroom as well as in the traditional classroom.

The following sections provide you with guidelines to successfully make the transition to a LogicalCHOICE classroom, including setting up the facilitator and student environments, using training materials, and facilitation best practices.

Configuring the Learning Environment

There are many ways to configure your connected classroom. The precise configuration you use will be determined by various factors, including budget, space, availability of hardware and resources, and the needs of your learners and other training stakeholders. As a model, here is a list of the major components you'll typically want to incorporate into your LogicalCHOICE virtual classroom.

Facilitator Environment Design

A good design for the facilitator's working environment is the starting point for a successful Logical CHOICE course experience.



Figure 3: LogicalCHOICE facilitator workstation and classroom layout

Facilitator's workstation: This will serve several functions. It is the command, communication, and control center for the training event. This is where you will coordinate and guide the learning event. It is also the primary broadcast and communication hub for the classroom, hosts the web conference software, and is also configured with the demo applications, slides, and other media that you'll broadcast to the participants during the session. Look for a robust, reliable, versatile PC with plenty of memory, storage capacity, and flexible and powerful graphics support.

Internet: Well-connected, reliable, fast Internet connectivity is a must.

Web sharing software: Selecting and mastering your websharing tool is second in importance only to your mastery of the course content itself. See the “Enabling Technologies” section for some of the features and capabilities you may want to consider when you standardize on this key tool.

Video: You'll need appropriately-placed web cams so virtual students can see the facilitator and other aspects of the classroom, and so the facilitator is able to move away from the workstation as needed.

Displays: Plan for multiple displays; one for the facilitator's own desktop (which will also be projected in the classroom and shared in the web meeting); a large monitor to display the video feeds of all participants, giving them a visual presence in the classroom, and potentially a private monitor for housekeeping tasks and for private communications. For example, there may be times when learners do not want to have their questions to the instructor made public, so they can post private chat messages here. You can also use the private monitor to review teaching notes, launch presentation files, and so forth.

Interactive whiteboard: In the classroom, an interactive whiteboard can be a great tool for the instructor to focus learner attention by performing markup or highlighting important screen features. Remote students will need a video feed on the board; or you may choose to rely entirely on the whiteboarding features of your web sharing application.

Audio: You'll need an audio solution. You can use microphones and speakers, or opt for phone communications. If you use a mic and speakers, a wireless microphone is a great tool for maintaining the audio portion of the web broadcast from anywhere in the room. You may need additional microphones for the student areas, as well as speakers placed so everyone can hear virtual students when they contribute.

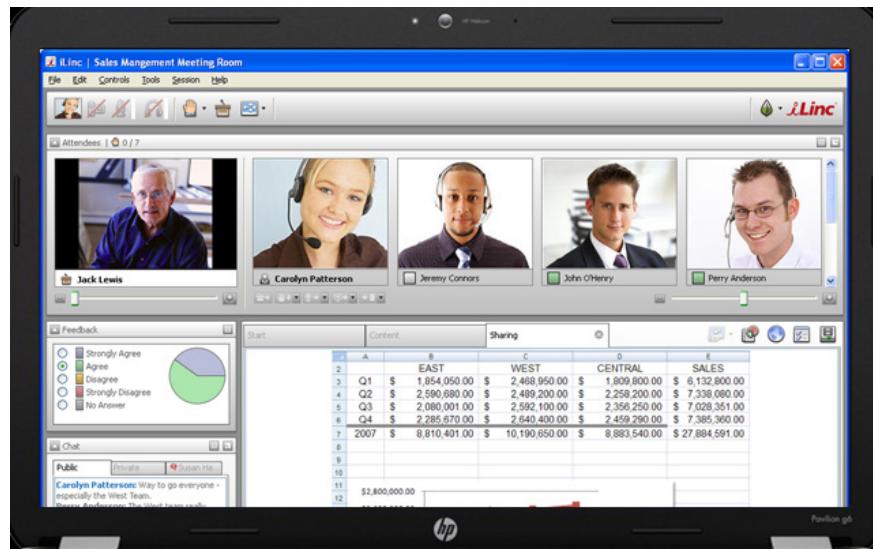


Figure 4: Facilitator screen layout

Student Environment Design



Figure 5: LogicalCHOICE student work areas—local and remote

The LogicalCHOICE student environment includes:

- Access to the software environment used to key the course, when appropriate. This may be a classroom computer, a locally-configured computer for remote students, or access to a virtual lab environment.
- Client access to the web-sharing software. This may involve installing a small client component on the student's local computer, or it may be purely a browser-based solution.
- A web-connected computer with a web cam. Remote students should have personal webcams if possible; on-site students may have individual cameras as well, or there may be general classroom cameras.

- Microphone or telephone headset/speakerphone, depending on whether audio will be provided through the web conferencing software or through the telephone. Remote students working from an office cubicle environment may find a headset necessary to avoid disturbing cubicle neighbors. On site, the audio can be configured through classroom microphones and speakers or the telephone.

In general, the audio/video link should be designed so that remote and on-site learners can all see and interact with each other as smoothly as possible.



Figure 6: Example of a student screen layout

The student screen layout should be arranged to minimize the need for switching between windows. In the best possible configuration, students have a large monitor (or multiple monitors in an extended desktop) and can arrange everything (web conferencing windows, local instance of applications they are working with in practice activities, and student manual ebook) with minimal overlap of windows, so they can work without having to switch between windows.

Enabling Technologies

A wide variety of technologies are available to support the classroom of choice.

Web Conferencing Software

Web conferencing software provides a delivery platform for distance learning and collaboration, including many of the tools you need to provide instructor-led training from a distance. Example web conferencing platforms include iLinc, Cisco WebEx, Citrix GoToMeeting, Microsoft Lync, Adobe Connect, and numerous others.

Features that are commonly found in web conferencing technologies include the following.

Computer desktop sharing	The ability to broadcast the display of a computer desktop is obviously useful for delivering hands-on computer training demos. But other subject matter may be delivered successfully this way as well. Screen sharing enables you to broadcast to a learner audience many of those things that you can display on a computer, including slides, graphics, and documents. But some materials, such as videos and animations, may not transmit particularly well in this manner. You may find you need to use other means to broadcast such media. If your system supports it, learners can share their screens with you, enabling you to provide remote support and help as learners perform activities on their own computers.
Remote control	If desktop sharing includes the ability for a remote user to use their own mouse and keyboard to control the shared computer, then you have <i>remote control</i> capability. This opens up numerous possibilities, such as giving learners practice time on remote systems that they might not normally be able to access on their own.
Slide display	PowerPoint and other presentation programs can be used in conjunction with desktop sharing to present slides. Some presentation tools provide a special environment for PowerPoint slide display, including options to highlight and mark up slides, as well as share slide sets with learners after the presentation.
Shared whiteboard	By typing or drawing on a whiteboard application, presentation window, or shared application window, meeting participants can add drawings or annotations that can be viewed by all.
File sharing	File-sharing features provided by many web conferencing applications enable participants to exchange files during a session.
Integrated text-based messaging	Integrated text-based messaging enables participants to communicate in real time through private and public text-based chat.
Integrated video	Integrated video enables participants to share a video feed from a web cam.
Integrated Voice over Internet Protocol (VoIP) audio	Integrated VoIP audio enables two-way voice conversation among participants directly through the connected computer rather than a separate telephone connection. For some remote users, using VoIP audio directly through their computer may be an attractive alternative to using telephone for the audio connection, especially if the only telephone they have access to in their learning environment is a cell phone.

Session archiving	Session archiving provides an audio/video recording of various media shared through web conferencing. This can be useful for recording training sessions for review and to share with those who were unable to attend.
Polling and questioning	Polling and questioning features enable participants to post responses to impromptu questions.
Breakout rooms	Breakout rooms provide a session within the session that enables participants to temporarily split off into a separate sub-session where they can collaborate separately from other participants and return to the main session when they are done.

Other Conferencing and Collaboration Tools

In addition to web conferencing tools, you may choose to use additional tools to support conferencing and collaboration, especially in regard to activities before and after the training session itself.

Web-conferencing tools are typically session-based. That is, to be available for participants, a facilitator must start a session and students must log in to that session. When the session ends, access to the tools ends. This approach works acceptably in support of learning sessions that occur in an agreed-upon time, but may not work well for ad-hoc or asynchronous learning before or after a session. You can initiate an ad hoc web conferencing session or you can use other tools to support learning and collaboration outside of a web conferencing session.

Furthermore, you may find that the tools integrated within a web conferencing system may not completely meet your needs. Or you may already have other tools in place, such as a standalone text messaging system that your organization has standardized on. You may choose to supplement web conferencing tools with other standalone tools to assemble a system that will work best for you and your learners.

Social network sites	Social networking web sites provide features that supplement (and to some extent overlap) those found in web conferencing tools. Example sites include LinkedIn, Facebook, and Google+. Whereas web conferencing tools tend to be session-based (lending themselves to synchronous learning), features of most social networking sites tend to be asynchronous. Social networking sites may provide a good tool for asynchronous communication among class participants before and after a course.
Email	Email provides an always-on way to communicate outside of a class session. It can provide a useful way to provide support to learners beyond a class session. If you use email this way you may find it beneficial to set up a separate account exclusively for this purpose, to keep it separate from your normal email.

File sharing	Numerous cloud-based file-sharing services are available. These may be easier for many users to use than FTP or other traditional file-sharing tools.
Standalone communication and collaboration tools	Some of the features in web conferencing tools are available in stand-alone products, such as remote desktop sharing tools, text messaging, VoIP audio and video conferencing, discussion boards, and so forth. In many cases, you do not have to set up a special session to use them, so such programs may be suitable for one on one support after class—especially if all of the course participants already have such tools in place.

Other Classroom Technology Needs

Other technologies that you may need to configure your connected classroom include the following.

Individual web cams	Each participant is represented in the web conferencing software through a web cam directed at their face so other participants can see and interact with them. If possible, even students in the classroom should have a face cam so they can interact with remote students on an equal basis.
Classroom video	The classroom may also have one or more classroom cameras to provide different views of the classroom environment.
Alternate audio approaches	Audio conferencing by telephone is a tried-and-true technology, and many participants have extensive experience using such systems. This can provide an effective alternative to using special microphones or web cam audio in the classroom. Furthermore, some learners may not have microphones in the environment where they will be taking the course, but may have telephone access.
In-class projection systems	An in-class projection system can provide you with a way to share your presentation screen with local learners. The monitor can be shared with remote learners through web conferencing tools.
Interactive whiteboard	This is a projection monitor or large display monitor that provides a touch or pen interface that enables you to interact with the large display as though you were using a mouse. Typically, such devices include other capabilities, such as markup, the ability to draw on top of the currently displayed screen.

Remote labs

Virtual labs can support hands-on learning, even in situations where providing direct access to software and hardware may be a challenge.

The LogicalCHOICE Experience

Your training materials should recognize and support the choices you have, and should accommodate and support your needs. In recognition of this, Logical Operations has developed the LogicalCHOICE product line. For each course in the series, LogicalCHOICE includes print and e-book editions of the facilitator's manual, printed and e-book editions of the student manuals, and online resources presented through the LogicalCHOICE Platform.

The LogicalCHOICE Platform

LogicalCHOICE includes a collection of online resources that provide a key component of the LogicalCHOICE learning experience and serve as the launch pad for various resources used in the course. Students and facilitators log in to access course resources on the LogicalCHOICE Course screen, the home page for a particular course. The LogicalCHOICE Course Screen is configurable. The person responsible for configuring courses in your organization can use the administrator account to configure the platform according to your organization's needs.

The LogicalCHOICE Course screen provides access to the following class resources:

- *Ebook*, an interactive electronic version of the printed book for your course, which can be loaded and viewed from a variety of platforms including iPad®/iOS, Android™, Windows® tablets, and other devices.
- *LearnTOs*, animated components that enhance and extend the classroom learning experience.
- A link to the web conferencing session for the course.
- Links to social collaboration sites associated with the course.
- Student Whiteboard for sketches and notes.
- Link to virtual labs, which provide access to the technical environment for your course.
- Checklists with useful post-class reference information, such as procedure references.
- Notices from the LogicalCHOICE administrator.
- Course files.

Other resources provided through the LogicalCHOICE Platform include:

- The course assessment.
- Newsletters and other communications from the learning provider to learners.
- Mentoring services.
- A link to the training provider web site.

- The LogicalCHOICE store.

The following resources are considered core to the course, and will always be available to students, regardless of customization: ebook, LearnTOs, course files, course assessment, and checklists.

The LogicalCHOICE Student Edition Manual

The student manual documents the training experience for the learner, providing a complete reference of the material covered in the course. The student manual is available to the learner in both ebook and print form so the learner can use whatever format she or he finds most convenient. The student manual:

- Is divided into lessons and topics that cover the subject matter, with lessons typically arranged in order of increasing proficiency.
- Covers results-oriented topics that include relevant and supporting information you need to master the content.
- Provides extensive activities and discussions designed to enable you to practice the guidelines and procedures as well as to solidify your understanding of the informational material presented in the course.
- Provides concise procedure references and guidelines.
- Includes reflective activities and practice labs to reinforce learning and promote higher order thinking.
- Includes data files and supporting files for this course, available by download from the LogicalCHOICE Course Screen, which is accessible via an access key provided in an email for each unit ordered.
- Includes a glossary of the definitions of the terms and concepts used throughout the course.
- Includes a table of contents and index to assist in locating information within the instructional components of the book.

The LogicalCHOICE Instructor's Edition

The LogicalCHOICE Instructor's Edition is essentially the Student Edition extended with features that support the class facilitator and others involved in setting up and delivering the course. The additional features of a LogicalCHOICE Facilitator's Edition include:

Course setup Course setup notes describe hardware and software requirements for the facilitator and student computers, and include step by step instructions to prepare computers for the class.

Course-specific delivery tips Supplemental content and delivery tips provide the facilitator with additional insights to deliver the course successfully.

Presentation planners Presentation planners help the facilitator plan the class schedule, and include examples of a continuous schedule (e.g. over one day) and an extended schedule (e.g. four, 90-minute sessions).

Facilitator notes In the page margins, the facilitator edition includes notes and tips to the facilitator. The following types of notes are provided.



A **display slide** note provides a prompt to the instructor to display a specific slide from the provided PowerPoint files.



Content delivery tips provide guidance for specific delivery techniques you may want to utilize at particular points in the course, such as lectures, whiteboard sketching, or performing your own demonstrations for the class.



Managing learning interactions provides notes on suggested places to interact with the class as a whole. You might poll the class with close-ended questions, check comprehension with open-ended questions, conduct planned discussion activities, or take notes and questions from the group to “park” and address at a later point in the class.



Monitoring learner progress notes suggest when you might want to monitor individual students as they perform activities, or have private “sidebar” conversations with specific individual participants.



Engaging learners notes suggest opportunities to involve the students in active ways with the course presentation, such as enabling them to demonstrate their work to the class as a whole, or checking in on the logistics of the presentation.



Incorporating other assets notes suggest when and how to include other types of media, such as visiting LogicalCHOICE social media sites, accessing specific web resources, or utilizing media assets provided with the course, such as Logical Operations' LearnTOs.



Additional notes show where, on occasion, there may be instructor notes or tips that appear in a separate section at the back of the courseware and not in the margins.

Delivering a LogicalCHOICE Course

Although LogicalCHOICE provides a thorough, tested solution for delivering instructor-led training, as with any course, the key to delivering the best possible training is to be prepared. Not only should you understand the subject matter, but you should be thoroughly familiar with the course content and materials. You should take steps to ensure that everything is ready for the day of class. Following are guidelines to help you prepare.

Before You Deliver the Class the First Time

Before you teach the class for the first time, perform the following steps to prepare.

Learn to Use Your Web Conferencing Tool

Be sure to also plan for any additional time it will take you to learn new tools. For example, if you're using a web conferencing system to present to remote learners, make sure that you know how to:

- Schedule a web conferencing session and invite other participants.
- Display a slide presentation.
- Display the whiteboard.
- Share an application or the desktop.
- Pass control of the application to another participant in the session.
- Point at and mark up the display so that remote learners can see what you're pointing at or talking about. (You should be able to do this with slides, whiteboard, or a shared application screen.)
- Exchange files with remote learners.

Get to the Know the Course and Make It Your Own

Before you teach a class for the first time, perform the following steps:

1. Become thoroughly familiar with the course materials.
 - At your desk (or wherever you work best), work through the course like a student, keying through each activity in sequence so you know what students will do in class as you lead them.
 - Note any lead-ins, transitions, and storylines you will use in delivery, and add or enhance with your own ideas.
 - Make notes in your Instructor's Edition to add your own ideas.
 - If you will have remote learners, consider how you will keep them engaged and note any accommodations you may need to make to create the best possible learning experience.
 - Read and consider the notes provided in the Instructor's Edition to call out areas where you may want to focus on aspects of content delivery, managing learner interactions, engaging learners, monitoring learner progress, and incorporating other assets. Add your own notes in the margins as needed.
2. Prepare your slides. As you work through the course materials, each time you encounter an instructor note calling for a slide, look at the corresponding slide in the course materials so you know how to present it. Add your own slides where you think they are appropriate. Add margin notes to indicate where your new slides appear, and mark existing slide notes to indicate those you don't wish to use.
3. Prepare question postings. If you're using a web conferencing system to deliver the course, you may wish to create import files containing any polling and quiz questions you plan to use during the session.
4. Prepare a file set. Although you may the classroom and web conferencing environment set up before class, it's always good to plan for contingencies. Keep an extra set of files handy for anything you may need to retrieve during class. Put your files in a network location, cloud storage, thumb drive, or some other location where you will able to access them in class. Files you may wish to have on hand include PowerPoint files for the course, student data, any web shortcuts you plan to use, additional media and example documents you may wish to share, polling questions, and anything else you might want to refer to.

Before You Deliver the Class Each Time

Each time you deliver the class, perform the following steps before the class begins. Be sure to do this well ahead of the class start time so you can deal with any technical problems that might occur.

1. Perform the course setup. Course setup requirements and instructions are provided in the Instructor's Edition of the course. If you delegate this task to someone else, you may wish to verify the class setup has been done correctly. If you have remote learners, send them (or whoever will set up their system) with the course setup instructions well enough in advance of class that they have time to do so.

2. Set up the conferencing session. If you will use web conferencing software, send a session invitation to all participants. If you know you will have students who have not used a web conferencing system before, plan this into the class schedule.
3. Book the classroom. If you're using shared space for your local classroom, verify that the room is available and assigned to your class.

The Day of the Class

Perform the following steps on the day of the class.

1. As you begin class, orient learners to the ground rules of working with web conferencing tools, how to participate, and so forth. For example:
 - If you're using audio/video conferencing, the slight communication delay in the call may cause learners to talk over each other. It can also be confusing for participants to know who's talking. Some conferencing tools highlight on screen the name of the participant who is currently talking. For other systems, it may be good to have learners lead in with "Hi, this is [student name]" before they begin speaking.
 - Tell students how to set up their web conferencing windows for the best experience. If you're teaching a hands-on computer training course, it may be beneficial to have them show the web conferencing tools in a window (rather than in full screen mode) so they can arrange the tools in a way that enables them to easily switch between their application window (for hands-on activities) and the conferencing windows (for discussion, interaction, and to view shared slides and demos).
2. If learners are not familiar with the web conferencing tool, provide them with an orientation to the tool. You can do this within the ongoing context of class, introducing and explaining each feature as you use it for the first time.
3. Once you have shared your screen, orient students to the LogicalCHOICE Course screen, pointing out any components you plan to use in your class:
 - eBook – The interactive electronic version of the course manual
 - LearnTOs – Supplemental course content in the form of animated multimedia
 - The virtual classroom for the course
 - Social media resources related to the course
 - Personal whiteboard for sketches and notes
 - Virtual labs, for remote access to the technical environment for your course
 - Checklists with useful post-class reference information
 - The course assessment
 - Newsletters and other communications from your training organization

- Mentoring services
 - A link to the website of your training provider
 - The LogicalCHOICE store
4. Provide learners with the URL for the LogicalCHOICE Course screen and with their login information.
5. Have learners open their ebook or print manual, and explain how the manual is laid out:
- The manual is divided into lessons and topics that cover the subject matter, with lessons typically arranged in order of increasing proficiency.
 - The results-oriented topics include relevant and supporting information you need to master the content. Each topic has various types of activities designed to enable you to practice the guidelines and procedures as well as to solidify your understanding of the informational material presented in the course. Procedures and guidelines are presented in a concise fashion along with activities and discussions. Information is provided for reference and reflection in such a way as to facilitate understanding and practice.
 - Datafiles for various activities as well as other supporting files for the course are available by download from the LogicalCHOICE Course screen.
 - At the back of the book, you will find a glossary of the definitions of the terms and concepts used throughout the course. You will also find an index to assist in locating information within the instructional components of the book.
6. As you present the course, you have many tasks to attend to. In addition to the activities any instructor must perform, you will also need to keep the classroom technology configured and ensure that all of your learners, local and remote, are engaged and operating effectively. The following Presentation Best Practices provide some guidance in this challenging process.

Presentation Best Practices

Delivering courses in the connected classroom may provide benefits beyond just providing you and your students with choice of scheduling and location. Enhancements to support better communication among participants and promote active learning, while aimed at making distance learning a better experience, will likely improve the experience for all learners, both remote and local.

Teaching in the connected classroom can take some adjustment at first. Learners bring different aptitudes, attitudes, experience, and skills. Meeting the needs of all students all the time has always been a challenge. Add to that the challenge of facilitating learning from a distance, and it becomes apparent that the attributes of a successful trainer—subject matter knowledge, planning, communication skills, learner focus, classroom time management, and so forth—are more important than ever.

Promote Active Learning

 Slides, screen images, and other presentation graphics are useful for presenting material that is abstract or that can't be easily shown in the classroom context. As needed, you can add your own slides to those included in LogicalCHOICE. However, avoid adding so many slides that your course becomes heavy on lecture and light on activity. If you want to cover supplemental content, look first for ways that you can deliver it through guided hands-on activities or demonstra-

tions. When learners *interact* with new ideas, rather than just trying to absorb and memorize them, they maintain interest and attention. When possible, enable learners to experience activities for themselves, rather than simply talking about them or demonstrating them.

Software Demonstrations



Screen sharing software enables you to share your screen with students who are remote.

There can also be benefits for students who are local. Because the image is shown directly on their own screen, visibility isn't subject to limitations in the room based on lighting, distance from the projected display, and so forth. Furthermore some screen-sharing systems provide built-in session recording so you can save a video which learners can refer to for review. If your facilitator environment provides screen markup capabilities, use this to advantage. For example, to help students attend to a particular icon or portion of the screen as you guide them through a hands-on activity, use your markup tools to highlight the portion of the screen they should attend to.

Parking Lot



As you progress through the course, learners may ask questions for which you do not have an immediate answer. Or they may ask a question for which you have an answer, but you want to wait until other topics have been covered before you address the question. In such cases, it isn't sufficient to simply put off the answer until later. To assure students that all of their questions will be addressed, you can create a parking lot, where you will note these questions to assure students they will all be addressed. Later on, when you reach the proper point to address the question (or after you have researched the answer during a break in instruction), you can circle back to the parking lot, answer the question, and check it off as having been answered. The tool you use to record parking lot questions could be as simple as notes you type into Windows Notepad or another simple text editor. You can display the parking lot questions on the presentation monitor if there is room there, or you can position it on your private monitor, and periodically move it back into view on the presentation monitor as you review existing questions and add new ones.

Use Questioning and Polling Tools



Web conferencing tools such as polling, questioning, and even the simple but effective **raise your hand** button provide you with a quick way to take a reading on how every learner is doing. Of course, as in the traditional classroom, you may ask learners how they are doing and you receive back an all-too-quick response, "we're doing fine." If you're not convinced that they really are doing fine, deliver a quick skill check: use your web conferencing tool to pose a quick question about content you just covered, or observe their performance closely in the next activity. The LogicalCHOICE Instructor's Edition provides notes with examples of where to insert questions and polls.

Level the Playing Field



Set up the classroom environment and when possible ensure that learner environments are configured so there is no disadvantage to being remote or local. As you deliver the course, try to use the same tools whenever possible for dialogue, questioning, polling, and so forth to enable all students to participate equally.

Keep aware of how you're communicating. If you're acting out visually, using your hands or facial expressions, drawing or highlighting, make sure you do these things directly on your web cam video or on the presentation monitor where they will be seen by all students. Also remember that the typically slow frame rate of web video will blur fast movements or subtle gestures, which will not translate well to online delivery.

Monitor and Adjust Frequently

 As a human being delivering training, you have an immediate advantage over e-learning and other forms of self-study. You have an innate ability to sense affect: how learners are doing, whether they are getting it, and whether they are struggling to keep up. Take advantage of your ability to sense how learners are doing, and adjust your delivery as needed. Use activities and lesson labs provided in the LogicalCHOICE course as a chance to monitor student performance.

Activities and lesson labs provide a good opportunity to check learner progress. If you suspect a learner is having trouble, you can use this time to have them share their screens privately with you so you can see how they're doing. A private chat message may also be sufficient as a way to check in with individual students.

Make Eye Contact

 Assuming your learners have face-cams (individual web cams for each student), don't forget to look into their faces as you present. Although making eye contact may be a natural thing to do when all of your learners are local, it is easy to neglect to do this when you have remote learners. Make sure the video monitor with the face-cam videos is located where you can see it while you're presenting. You can tell a lot simply by looking at the expressions on the faces of your learners.

Use Dialogue

 If you do all of the talking, you won't have a good sense of what students are thinking. In many situations, you can walk alongside students to take them where you want them to go through discussion, rather than simply carrying them there through lecture.

For example, rather than say "this Excel formula works perfectly except for this one place," you might say "Is this Excel formula meeting our needs perfectly at this point? Is there one place where we might need to do something different?"

Both approaches will get learners to the same point, but there are benefits to selectively using dialogue at times rather than always using lecture. Dialogue promotes active learning by increasing the effort of thought that learners must exert to get to the answer. Dialogue also gives you a sense of students' understanding. If nobody gets it, it will be immediately apparent through the ideas they communicate (or the silence).

Show and Tell

 Because lesson labs are not as tightly scripted as guided activities, there will likely be more variation in the results produced by learners. If time permits, you might consider having students volunteer to share their desktops to show their results and explain what they did to produce them.

Build Flexibility Into the Class Schedule

 Build flexibility into your class schedule so you can monitor students' progress through the course and adjust as needed. Just as a balloonist carries sandbags that can be released as needed when the balloon falls short on lift, consider including optional material in your class schedule that you can use or not use depending on the pace of the class. The lesson labs, LearnTOs, and ad hoc social/interaction tips provided in the LogicalCHOICE course materials are a good source of such material. You may also want to build-in your own additional optional material, such as outside resources, field trips to relevant websites, or demos that you can incorporate into the class

if there is adequate time. If you're training a homogeneous group, such as a group who all work in the same department, organization, or industry, you may use this as an opportunity to add content that has been customized for this specific group.

Schedule adequate hours to account for the time these extras will require in the best case scenario, where students move through the course quickly. In anything less than the best case scenario, you can omit the extra material to gain back some time in the schedule. Be upfront with students and tell them at the beginning of the class that you will cover all of the material in the course, but depending on the timing of the class, you may not have time during the class for students to complete every lesson lab or the other extras. The labs are designed to be completed independently and include all of the files needed. If they want, students can perform the lesson labs any time after the class if they wish to gain additional practice.

Considering directing learners to related blog sites, wikis, discussion sites, and other useful web resources. You can provide links to other resources through the LogicalCHOICE Course screen. The LogicalCHOICE Course screen also provides additional resources that you can utilize to support additional learning or reference after the class, such as LearnTO videos, procedure references, and checklists.

After Class



After you have finished the class, follow up by participating in social media sites that you have set up for the course. You have the choice to use the social media sites you and your learners prefer. Ideas for ways that you can engage learners after class include:

- Post or tweet additional tips and hints for learning beyond the course.
- Provide a forum in which learners can post follow-up questions about material covered in the course, where you and others can provide them with answers.
- Provide opportunities for learners to post examples of ways in which they are applying what was covered in the course.
- Provide information and notifications on other courses learners may be interested in.
- Promote communities of practice by providing a forum in which learners can support and share information with each other.

Conclusion: Making the Transition to Virtual

The benefits offered by the connected classroom are significant. More organizations are taking advantage of the opportunities offered by new technologies and are delivering high quality instructor led training while offering more flexibility. Take advantage of information and activities provided in your CHOICE course materials. As you teach in the connected classroom, you will develop additional techniques beyond those presented here to take advantage of the technology and to promote active learning for all of your students. Continually learn from your experience and share your ideas.

Logical Operations provides an online community and resource center where instructors from all over the world share their ideas, interact, and engage each other. Visit www.linkedin.com to join our group, the Logical Operations Instructor Community. Also, receive free access to a CHOICE instructor course made specifically for you – visit www.lo-choice.com and enter access key **LCSYB24YEF**.

Again, congratulations on your choice – the right choice – the Logical CHOICE!

**Microsoft® SQL
Server® 2012:
Database
Administration
(Exam 70-462)**

Microsoft® SQL Server® 2012: Database Administration (Exam 70-462)

Part Number: 093108

Course Edition: 1.1

Acknowledgements

PROJECT TEAM

<i>Author</i>	<i>Media Designer</i>	<i>Content Editor</i>
Senthilkumar M	Alex Tong	Tricia Murphy

Notices

DISCLAIMER

While Logical Operations, Inc. takes care to ensure the accuracy and quality of these materials, we cannot guarantee their accuracy, and all materials are provided without any warranty whatsoever, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. The name used in the data files for this course is that of a fictitious company. Any resemblance to current or future companies is purely coincidental. We do not believe we have used anyone's name in creating this course, but if we have, please notify us and we will change the name in the next revision of the course. Logical Operations is an independent provider of integrated training solutions for individuals, businesses, educational institutions, and government agencies. Use of screenshots, photographs of another entity's products, or another entity's product name or service in this book is for editorial purposes only. No such use should be construed to imply sponsorship or endorsement of the book by, nor any affiliation of such entity with Logical Operations. This courseware may contain links to sites on the internet that are owned and operated by third parties (the "External Sites"). Logical Operations is not responsible for the availability of, or the content located on or through, any External Site. Please contact Logical Operations if you have any concerns regarding such links or External Sites.

TRADEMARK NOTICES

Logical Operations and the Logical Operations logo are trademarks of Logical Operations, Inc. and its affiliates.

SQL Server® 2012 is a registered trademark of Microsoft Corporation in the U.S. and other countries. The other Microsoft products and services discussed or described may be trademarks or registered trademarks of Microsoft Corporation. All other product and service names used may be common law or registered trademarks of their respective proprietors.

Copyright © 2013 Logical Operations, Inc. All rights reserved. Screenshots used for illustrative purposes are the property of the software proprietor. This publication, or any part thereof, may not be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, storage in an information retrieval system, or otherwise, without express written permission of Logical Operations, 3535 Winton Place, Rochester, NY 14623, 1-800-456-4677 in the United States and Canada, 1-585-350-7000 in all other countries. Logical Operations' World Wide Web site is located at www.logicaloperations.com.

This book conveys no rights in the software or other products about which it was written; all use or licensing of such software or other products is the responsibility of the user according to terms and conditions of the owner. Do not make illegal copies of books or software. If you believe that this book, related materials, or any other Logical Operations materials are being reproduced or transmitted without permission, please call 1-800-456-4677 in the United States and Canada, 1-585-350-7000 in all other countries.

Microsoft® SQL Server® 2012: Database Administration (Exam 70-462)

Identifying the SQL Server Platform and Tools.....	1
Overview of SQL Server Features.....	2
Create Filegroups and Partitions.....	9
Deploying SQL Server.....	21
Install SQL Server	22
Migrate SQL Server From a Previous Version.....	34
Configuring SQL Server.....	39
Configure Server Properties.....	40
Configure Memory and CPU.....	46
Configure Database Mail.....	59
Managing Databases in SQL Server 2012.....	73
Create a Contained Database.....	74
Configure Database Options.....	82
Managing SQL Server Security.....	95
Configure Server Security.....	96
Create Server Roles	109
Configure Database Security.....	116

Implementing Advanced Security Settings.....	131
Create Certificate-Based Logins.....	132
Apply Auditing and Policy-Based Management Strategies.....	141
Applying Encryption and Compression.....	167
Apply Encryption on Databases.....	168
Compress a Database Table.....	180
Working with Indexes and Log Files.....	187
Work with Full Text Indexes.....	188
Configure Advanced Storage and Indexing Settings.....	199
Configure Log Files.....	215
Working with Backup and Restore.....	229
Perform a Backup.....	230
Restore a Database.....	240
Import and Export Data.....	247
Configure the SQL Server Agent.....	259
Implementing High Availability.....	277
Work with Clustering.....	278
Work with Replication.....	295
Work with Mirroring and AlwaysOn.....	307
Work with Central Management Servers.....	319
Optimizing Server Performance.....	325
Manage Server Performance.....	326
Optimize Indexes.....	351
Troubleshooting Issues and Recovering Databases.....	361
Troubleshoot Database Issues.....	362
Perform a Database Recovery.....	376
Performing Advanced Database Management Tasks.....	383
Manage Database Infrastructure.....	384
Configure SSIS.....	412

Appendix A:

Exam 70-462: Administering Microsoft SQL Server 2012 Databases

Lesson Labs.....	435
Glossary.....	451
Index.....	459

Using the Microsoft® SQL Server® 2012: Database Administration (Exam 70-462) Instructor's Edition

Welcome to the Instructor

Welcome and congratulations on your choice to use the finest materials available on the market today for expert-facilitated learning in any presentation modality. You can utilize the *Microsoft® SQL Server® 2012: Database Administration (Exam 70-462)* curriculum to present world-class instructional experiences whether:

- Your students are with you in the classroom, or participating virtually.
- You are presenting in a continuous event, or in an extended teaching plan, such as an academic semester.
- Your presentation takes place synchronously with the students, or asynchronously.
- Your students have physical courseware, or are using digital materials.
- You have any combination of these instructional dimensions.

To make the best use of the *Microsoft® SQL Server® 2012: Database Administration (Exam 70-462)* materials in any or all of these dimensions, be sure to review the contents of the LogicalCHOICE Facilitator's Guide for an orientation to all of the components of the LogicalCHOICE experience.

Preparing to Present the LogicalCHOICE Experience

Effectively presenting the information and skills in this course requires adequate preparation, no matter your presentation modality. As such, as an instructor, you should familiarize yourself with the content of the entire course, including its organization and instructional approaches. You should review each of the student activities and exercises so you can facilitate them during the learning event. Make sure you review all of the instructor tips for presenting in the different dimensions that are available in the margins of your Instructor's Edition.

In addition to the curriculum itself, Microsoft® PowerPoint® slides, data files, and other course-specific support material may be available by downloading the files from the

LogicalCHOICE Course screen. Be sure to obtain the course files prior to your learning event and make sure you distribute them to your students.

Course Facilitator Icons

Throughout the Instructor's Edition, you may see various instructor-focused icons that provide suggestions, answers to problems, and supplemental information for you, the instructor.

Icon	Description
	A display slide note provides a prompt to the instructor to display a specific slide from the provided PowerPoint files.
	Content delivery tips provide guidance for specific delivery techniques you may want to utilize at particular points in the course, such as lectures, whiteboard sketching, or performing your own demonstrations for the class.
	Managing learning interactions provides notes on suggested places to interact with the class as a whole. You might poll the class with closed-ended questions, check comprehension with open-ended questions, conduct planned discussion activities, or take notes and questions from the group to "park" and address at a later point in the class.
	Monitoring learner progress notes suggest when you might want to monitor individual students as they perform activities, or have private "sidebar" conversations with specific individual participants.
	Engaging learners notes suggest opportunities to involve the students in active ways with the course presentation, such as enabling them to demonstrate their work to the class as a whole, or checking in on the logistics of the presentation.
	Incorporating other assets notes suggest when and how to include other types of media, such as visiting LogicalCHOICE social media sites, accessing specific web resources, or utilizing media assets provided with the course, such as Logical Operations' LearnTOs.
	Additional notes show where, on occasion, there may be instructor notes or tips that appear in a separate section at the back of the courseware and not in the margins.

Course-Specific Technical Requirements

Hardware

For this course, you will need one computer for each student and one for the instructor. Each computer will need the following minimum hardware configurations:

- Processor: 1.4 GHz 64-bit processor capable of running Hyper-V for hosting 64-bit VMs
- Hard disk: 500 GB or above
- RAM: 8 GB or above
- Network adapter: Compatible with Windows Server 2012
- DVD-ROM drive
- Keyboard and mouse (or other pointing device)
- 1024 x 768 resolution monitor recommended
- Projection system to display the instructor's computer screen

Software

- Microsoft® Windows Server® 2012 Standard Edition 64-bit operating system
- Microsoft® SQL Server® 2012 Enterprise Edition 64-bit



Note: As the Standard Edition of Windows Server 2012 contains only two VM licenses, you need to procure one more VM license to run the three VMs on the same physical system.

Brief Description of Course Setup

For each student and the instructor, you will set up a single physical computer. Next, on these physical computers, you will set up three virtual machines per student.

Base Operating System

The base operating system is Windows Server 2012 Standard edition. On this system, the Hyper-V role needs to be installed and configured.

Virtual Systems

You will need three Hyper-V virtual machines running Windows Server 2012.

Installation of Virtual Machines

For Windows Server 2012, the virtual machines have to be installed with basic options.

Install Windows Server 2012 on the physical system. Rename the system as SQL##.



Note: Assign SQL100 for instructor system. For each student system, assign a different name. For example, for student1 system assign SQL01, for student2 assign SQL02, and so on. This naming is needed to ensure that the name of the physical systems are unique on the network and there are no system name conflicts.

The activities in the course refer to the physical system as SQL01. Replace this name with the actual name assigned to the system.

- *For Windows Server 2012 (for the domain controller)*

Create a Hyper-V VM with 2 GB RAM and 60 GB disk space. Install Windows Server 2012 using the Install wizard. After the server is installed, change the server name to "Server2."

- *For Windows Server 2012 (Two Secondary Server VMs)*

Create two more Hyper-V VMs with 1 GB RAM and 60 GB disk space. Install Windows Server 2012 using the Install wizard. After the server is installed, change the server names to "Server3" and "Server4."

Setting Up the Course

For each student and instructor system:

1. Install Windows Server 2012 on the Physical System (SQL01).
 - a. Boot the computer with the Windows Server 2012 installation DVD or USB. Make sure that you select the appropriate media in the BIOS or boot selection screen.
 - b. In the **Windows Setup** dialog box, on the **Windows Server 2012** page, verify the settings.
 - a. In the **Language to install** drop-down list, verify that English (United States) language is selected.
 - b. In the **Time and currency format** drop-down list, verify that the **English (UnitedStates)** format is selected.
 - c. In the **Keyboard or input method** drop-down list, verify that the **US** method is selected and select **Next**.
 - d. On the **Windows Server 2012** page, select **Install now**.
 - e. On the **Enter the product key to activate Windows** page, in the text box, type the product key and select **Next**.
 - f. On the **Select the operating system you want to install** page, in the **Operating system** list, select the second option, **Windows Server 2012 Standard (Server with a GUI)**, and select **Next**.

- g. On the **License terms** page, check the **I accept the license terms** check box and select **Next**.
- h. On the **Which type of installation do you want** page, select the **Custom: Install Windows only (advanced)** option.
- i. On the **Where do you want to install Windows** page, delete the existing partitions, if any.
 - a. Select the partitions.
 - b. Select **Drive Options (advanced)**.
 - c. Select **Delete**.
 - d. In the **Windows Setup** message box, select **OK**.
 - e. Repeat the above four steps to remove all your existing partitions.
 - f. Select **Disk 0 Unallocated Space** and select **New**.
- j. In the **Size** text box, type 307200 and select **Apply** to allocate 300 GB as the size of the partition.
- k. In the **Windows Setup** dialog box, select **OK** to allow Windows to create new partitions for system files.
- l. On the **Where do you want to install Windows** page, with **Drive 0 Partition 2** selected, select **Next**.
- m. Wait for the installation to complete. Note the details displayed on the **Installing Windows** page.
- n. Windows will automatically restart the system. Wait for Windows to set up the Devices and show the Getting ready status. The system will automatically restart.
- o. On the **Settings** page, in the **Password** text box, type the password for the administrator.



Note: The password used for the administrator in all activities of this course is **server@2012**. You can change the password as desired. However, remember to use the updated password for all the activities.

- p. In the **Reenter password** text box, type the same password to confirm the password for the administrator. Select **Finish**.
 - q. Wait for Windows to finalize the settings and display the Windows Login screen.
2. Specify a new network driver and rename the Server as SQL01.
 - a. Press **Ctrl+Alt+Delete** to log in.
 - b. In the Login screen, in the **Password** text box, enter the password of the administrator.
 - c. If necessary, manually update your network driver.
 - d. Specify the remaining space on the hard drive as a new drive (D:).
 - a. In **Server Manager**, select **Tools-->Computer Management**.
 - b. In the Computer Management window, in the left pane, select the **Disk Management** option.
 - c. In **Disk 0** section, select the **Unallocated** space, right-click, and select **New Simple Volume**.
 - d. On the **Welcome to the New Simple Volume Wizard** page, select **Next**.
 - e. On the **Specify Volume Size** page, verify that the full disk space is listed in the **Simple volume size in MB** section and select **Next**.
 - f. On the **Assign Drive Letter or Path** page, with the **Assign the following drive letter** option selected, ensure that **D:** is selected, and select **Next**.
 - g. On the **Format Partition** page, observe that the **Format this volume with the following settings** option is selected.
 - h. Verify that the **File system** is **NTFS**, **Allocation unit size** is **Default** and the **Perform a quick format** check box is checked and then select **Next**.
 - i. On the **Completing the New Simple Volume Wizard** page, select **Finish**.



Note: If your DVD drive is already assigned the drive letter, change the drive letter, assign E: for your DVD drive, and assign D to the new partition you created.

- e. Wait for the Browser customization to complete. In the Server Manager window, in the left pane, select the **Local Server** option.
- f. In the Server Manager window, in the right pane, in the **Computer name** section, select the existing computer name link.
- g. In the **System Properties** dialog box, select **Change**.
- h. In the **Computer Name/Domain Changes** dialog box, in the **Computer name** text box, with the existing computer name selected, type **SQL01** as the new name and select **OK**.
- i. Close the **System Properties** dialog box.
- j. In the **Microsoft Windows** dialog box, select **Restart Now**.
3. Install the Hyper-V Role on SQL01.
 - a. Press **Ctrl+Alt+Delete** to log in.
 - b. In the Login screen, with the **User Name** as **Administrator**, enter the password of the administrator.
 - c. In the Server Manager window, on the **Dashboard** page, in the **Configure this local server** section, select the **Add roles and features** link.
 - d. In the **Add Roles and Features** wizard, on the **Before You Begin** page, select **Next**.
 - e. On the **Select installation type** page, verify that the **Role-based or feature-based installation** option is selected and select **Next**.
 - f. On the **Select destination server** page, with the **Select a server from the server pool** option selected, with **SQL01** as the server name, select **Next**.
 - g. On the **Select server roles** page, in the **Roles** section, check the **Hyper-V** check box to install Hyper-V role on SQL01.
 - h. In the **Add Roles and Features Wizard** dialog box, select **Add Features** to add the Hyper-V feature to SQL01.
 - i. On the **Select server roles** page, select **Next**.
 - j. On the **Select features** page, select **Next**.
 - k. On the **Hyper-V** page, select **Next**.
 - l. On the **Create Virtual Switches** page, select **Next**.
 - m. On the **Virtual Machine Migration** page, leave the default options unchanged and select **Next**.
 - n. On the **Default Stores** page, to accept the default locations for virtual hard disk files and virtual machine configuration files, select **Next**.
 - o. On the **Confirm installation selections** page, observe the Hyper-V roles and features that will be installed and select **Install**.
 - p. Wait for the installation to complete and select **Close** to close the Add Roles and Features wizard.
 - q. In the Server Manager window, in the left pane, observe that the **Hyper-V** feature appears.
 - r. Select **Local Server** and select **TASKS→ShutDown Local Server**.
 - s. In the **Shut Down Windows** dialog box, in the **Windows Server 2012** section, from the **What do you want the computer to do** drop-down list, select **Restart**.
 - t. From the **Option** drop-down list, select the **Operating System: Reconfiguration(Planned)** option and select **OK**.
 - u. Wait for Windows to restart.
4. Create a new virtual internal switch named **Internal** for connecting the VMs.
 - a. Press **Ctrl+Alt+Delete** to log in.
 - b. In the Login screen, with the **User Name** as **Administrator**, enter the password of the administrator.
 - c. In the Server Manager window, select **Tools→Hyper-V Manager**.
 - d. In the Hyper-V Manager window, in the left pane, select **SQL01**.

- e. In the **Actions** pane, in the **SQL01** section, select **Virtual Switch Manager**.
 - f. In the Virtual Switch Manager for SQL01 window, in the **Create virtual switch** section, from the list, select **Internal** and select **Create Virtual Switch**.
 - g. In the **Virtual Switch Properties** section, in the **Name** text box, type **Internal**.
 - h. In the **Connection type** section, verify that the **Internal network** option is selected and select **OK**.
5. Specify the IP address for SQL01.
- a. On SQL01, press **[Windows]** and select **Control Panel**.
 - b. In the Control Panel, in the **Network and Internet** section, select **View network status and tasks**.
 - c. In the Network and Sharing Center window, in the left pane, select **Change adapter settings**.
 - d. Select the **vEthernet (Internal)** adapter, right-click, and select **Properties**.
 - e. In the **vEthernet (Internal) Properties** dialog box, on the **Networking** tab, scroll down and select the **Internet Protocol Version 4 (TCP/IPv4)**, and select **Properties**.
 - f. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, select the **Use the following IP address** option.
 - g. In the **IP address** text box, type **192.168.1.230** as the IP address and press **Tab**.
 - h. In the **Default gateway** text box, type **192.168.1.226** as the IP address.
 - i. With the **Use the following DNS sever addresses** option selected, in the **Preferred DNS server** text box, type **192.168.1.226** as the IP address and select **OK**.
 - j. In the **vEthernet (Internal) Properties** dialog box, uncheck the **Internet Protocol Version6 (TCP/IPv6)** check box and select **Close**.
 - k. Close the Network Connections and Network and Sharing Center windows.
6. Enable .Net Framework 3.5.
- a. In the Server Manager window, on the **Dashboard** page, in the **Configure this local server** section, select the **Add roles and features** link.
 - b. In the **Add Roles and Features** wizard, on the **Before You Begin** page, select **Next**.
 - c. On the **Select installation type** page, verify that the **Role-based or feature-based installation** option is selected and select **Next**.
 - d. On the **Select destination server** page, with the **Select a server from the server pool** option selected, with **SQL01** as the server name, select **Next**.
 - e. On the **Select server roles** page, accept the default settings and select **Next**.
 - f. On the **Select features** page, in the **Features** list box, check the **.NET Framework 3.5 Features** check box.
 - g. On the **Confirm installation selections** page, select the **Specify an alternate source path** link.
 - h. In the **Specify Alternate Source Path** dialog box, in the **Path** text box, specify the drive letter of your DVD ROM drive followed by **:\\Sources\\SxS** and select **OK**.
 - i. On the **Confirmation installation** page, select **Install**.
 - j. On the **Installation progress** page, view the progress of the installation.
 - k. Wait for the installation to complete and select **Close**.
7. Install a default instance of SQL Server 2012.
- a. Open Windows Explorer and create a folder named **SQLServer2012 Installation** in C: drive.
 - b. Place an ISO image of the SQL Server 2012 64-bit installation media in the **C:\\SQLServer2012 Installation** folder.
 - c. Double-click the ISO image to access the setup file.
 - d. Double-click the **setup** file to launch **SQL Server Installation Center**.
 - e. In the SQL Server Installation Center window, in the left pane, select **Installation**.
 - f. In the right pane, select the **New SQL Server stand-alone installation or add features to an existing installation** link.

- g. In the **SQL Server 2012 Setup** dialog box, select **OK**.
 - h. On the **Installation Type** page, select the **Perform a new installation of SQL Server 2012** option and select **Next**.
 - i. On the **Product Key** page, select the **Specify a free edition** to install Evaluation edition or select the **Enter the product key** option and specify a product key and select **Next**.
 - j. On the **License Terms** page, check the **I accept the license terms** check box and select **Next**.
 - k. On the **Product Updates** page, uncheck the **Include SQL Server product updates** check box, and select **Next**.
 - l. In the **SQL Server 2012 Setup** dialog box, on the **Setup Support Rules Page**, ensure that no possible problems are identified and select **Next**.
 - m. On the **Setup Role** page, with the **SQL Server Feature Installation** option selected, select **Next**.
 - n. On the **Feature Selection** page, in the list of features, select **Select All** to select all the features.
 - o. On the **Installation Rules** page, check whether the installation rules identify any blocking issues and select **Next**.
 - p. On the **Instance Configuration** page, verify that the **Default instance** option is selected, and select **Next**.
 - q. On the **Disk Space Requirements** page, review the storage space requirements, and select **Next**.
 - r. On the **Server Configuration** page, select **Next** to configure the server by using default settings.
 - s. On the **Analysis Services Configuration** page, on the **Server Configuration** tab, select **Add Current User** to add **SQL01\Administrator** as user with administrative privileges for Analysis Services, and select **Next**.
 - t. On the **Database Engine Configuration** page, in the **Authentication Mode** section, select **Mixed Mode** option. In the **Enter password** and **Confirm password** text boxes, set the desired password for **sa** account. Select **Add Current User**, and then select **Next**.
 - u. On the **Reporting Services Configuration** page, verify that in the **Reporting Services Native Mode** section, the **Install and configure** option is selected and select **Next**. If you see the **Distributed Relay Controller** page, select **Add Current User** and then select **Next**.
 - v. On the **Error Reporting** page, choose whether to send error reports to Microsoft and the reporting server, and select **Next**.
 - w. On the **Installation Configuration Rules** page, select **Show details** and check whether the installation configuration rules identified any blocking issues that may halt the installation, and select **Next**.
 - x. On the **Ready to Install** page, view a summary of the options you selected and select **Install**. Wait for the installation to complete.
 - y. On the **Complete** page, select **Close**.
 - z. Close the SQL Server Installation Center window.
8. Set up Server2 in Hyper-V Manager on SQL01.
 - a. Create a new VM named Server2.
 - a. In the Hyper-V Manager window, in the right pane, in the **SQL01** section, select **New→Virtual Machine**.
 - b. In the **New Virtual Machine** wizard, on the **Before You Begin** page, select **Next**.
 - c. On the **Specify Name and Location** page, in the **Name** text box, type **Server2** as the name and select **Next**.
 - d. On the **Assign Memory** page, in the **Startup memory** text box, type **2048** to allocate 2 GB RAM.
 - e. Check the **Use Dynamic Memory for this virtual machine** check box and select **Next**.
 - f. On the **Configure Networking** page, from the **Connection** drop-down list, select **Internal** and select **Next**.

- g. On the **Connect Virtual Hard Disk** page, with the **Create a virtual hard disk** option selected, note the name and location that appear by default. In the **Size** text box, type **60** and select **Next**.
- h. On the **Installation Options** page, select the **Install an operating system from a boot CD/DVD-ROM** option. In the **Physical CD/DVD drive** drop-down list, select the drive-letter of your physical DVD ROM drive. If needed, insert the Windows Server 2012 DVD into the DVD drive.



Note: Alternatively, if you have the .ISO image of Windows Server 2012 stored in a separate folder on your system, in this step, you need to select **Browse** and in the **Open** dialog box, navigate to the folder containing the Windows 2012 ISO installation file. Select the .iso file and select **Open**.

- i. On the **Installation Options** page, select **Next**.
- j. On the **Completing the New Virtual Machine** wizard page, select **Finish** to complete the New Virtual Machine wizard steps.
- b. Install Windows 2012 on Server2 in Hyper-V Manager.
 - a. In the Hyper-V Manager window, in the **Virtual Machines** section, select the **Server2** VM, right-click, and select **Start** to start the virtual machine.
 - b. Wait for **State** to change from **Off** to **Running**. Right-click the **Server2** VM and select **Connect**.
 - c. In the Virtual Machine Connection window, the message “Connecting to Server2” is displayed first, followed by the **Windows Setup** dialog box.
 - d. In the **Windows Setup** dialog box, in the **Windows Server 2012** section, verify the settings.
 - a. In the **Language to install** drop-down list, verify that **English (United States)** language is selected.
 - b. In the **Time and currency format** drop-down list, verify that the **English (United States)** format is selected.
 - c. In the **Keyboard or input method** drop-down list, verify that the **US** method is selected. Select **Next** to proceed with the installation.
 - d. In the Windows Server 2012 section, select **Install now** to start the installation.
 - e. In the **Windows Setup** dialog box, on the **Enter the product key to activate Windows** page, in the product key text box, type the product key and select **Next**.
 - f. On the **Select the operating system you want to install** page, select the second option **Windows Server 2012 Standard (Server with a GUI)**, and select **Next**.
 - g. On the **License terms** page, check the **I accept the license terms** check box and select **Next**.
 - h. On the **Which type of installation do you want** page, select the **Custom: Install Windows only (advanced)** option.
 - i. On the **Where do you want to install Windows** page, verify that **Drive 0 Unallocated Space** is selected, and select **Next**.
 - j. Wait for the installation to complete. Note the details displayed on the **Installing Windows** page.
 - k. Wait for Windows to automatically restart the system.
 - l. Wait for Windows to set up devices and display the Getting ready status. Allow Windows to restart the system.
 - m. On the **Settings** page, in the **Password** text box, type the password for the administrator.
 - n. In the **Reenter password** text box, type the same password to confirm the password for the administrator. Select **Finish**.
 - o. Wait for Windows to finalize the settings and display the Windows Login screen.
 - c. For Server2 VM, specify the IP address as 192.168.1.226 and modify the other settings such as System Name, Firewall Properties, and Internet Security Enhanced Configuration.

- a. Press **Ctrl+Alt+End** to log in.
- b. In the Login screen, with the **User name** as **Administrator**, enter the password of the administrator.
- c. In the Server Manager window, in the left pane, select the **Local Server** option. In the Properties for Server section, in the **Ethernet** section, select the existing Ethernet link.
- d. In the Network Connections window, right-click the **Ethernet** network adapter and select **Properties**.
- e. In the **Ethernet Properties** dialog box, select the **Internet Protocol Version 4 (TCP/IPv4)** option and select **Properties**.
- f. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, select the **Use the following IP address** option.
- g. In the **IP address** text box, type **192.168.1.226** as the IP address and press **Tab**.
- h. In the **Default gateway** text box, type **192.168.1.226** as the IP address.
- i. With the **Use the following DNS** sever addresses option selected, in the **Preferred DNS** server text box, type **192.168.1.226** as the IP address and select **OK**.
- j. In the **Ethernet Properties** dialog box, uncheck the **Internet Protocol Version 6(TCP/IPv6)** check box, and select **Close**.
- k. Close the Network Connections window.
- l. In the Server Manager window, in the right pane, in the **IE Enhanced Security Configuration** section, select the **On** link.
- m. In the **Internet Explorer Enhanced Security Configuration** dialog box, in the **Administrators** section, select **Off**.
- n. In the **Users** section, select **Off** and select **OK**.
- o. In the Server Manager window, in the **Windows Firewall** section, select the **Public: On** link. In the WindowsFirewall window, in the left pane, select **Advanced settings**.
- p. In the Windows Firewall with Advanced Security window, select **Windows Firewall Properties**.
- q. In the resultant dialog box, on the **Domain Profile**, **Private Profile**, and **Public Profile** tabs, from the **Firewall state** drop-down list, select **Off** and select **OK**.
- r. If necessary, in the **Windows Firewall with Advanced Security** message box, select **OK** to skip the warning.
- s. In the Server Manager window, in the left pane, in the **Computer name** section, select the existing computer name link.
- t. In the **System Properties** dialog box, select **Change**.
- u. In the **Computer Name/Domain Changes** dialog box, in the **Computer name** text box, with the existing computer name selected, type **Server2** as the new name and select **OK**.
- v. In the **Computer Name/Domain Changes** message box, select **OK**.
- w. In the **System Properties** dialog box, select **Close**. In the **Microsoft Windows** message box, select **Restart Now** to restart the system.
- d. Set up AD DS on Server2.
 - a. Press **Ctrl+Alt+End** to log in.
 - b. In the Login screen, with the **User name** as **Administrator**, enter the password of the administrator.
 - c. In the Server Manager window, on the **Dashboard** page, in the **Configure this local server** section, select the **Add roles and features** link.
 - d. In the Add Roles and Features Wizard window, on the **Before you begin** page, select **Next**.
 - e. On the **Select installation type** page, observe that the **Role-based or feature-based installation** option is selected and select **Next**.
 - f. On the **Select destination server** page, in the **Server Pool** section, observe that **Server2** is selected as the server name and select **Next**.

- g. On the **Select server roles** page, in the **Roles** section, check the **Active Directory Domain Services** check box to install the AD DS role on Server2.
- h. In the **Add Roles and Features Wizard** dialog box, select **Add Features** to add the AD DS feature to Server2.
 - i. On the **Select server roles** page, select **Next**.
 - j. On the **Select features** page, select **Next**.
 - k. On the **Active Directory Domain Services** page, select **Next**.
 - l. On the **Confirm installation selections** page, observe the AD DS roles and features that will be installed and select **Install**.
 - m. Wait for the installation to complete and select **Close** to close the Add Roles and Features wizard.
 - n. In the Server Manager window, in the left pane, observe that the AD DS feature appears.
- e. Promote Server2 as the domain controller for the adwcycles.local AD forest.
 - a. In the Server Manager window, select the notification that appears and then select the **Promote this server to a domain controller** link.
 - b. In the Active Directory Domain Services Configuration wizard, on the **Deployment Configuration** page, in the **Select the deployment operation** section, select the **Add a new forest** option.
 - c. In the **Root domain name** text box, type **adwcycles.local** and select **Next**.
 - d. On the **Domain Controller Options** page, in the **Type the Directory Services Restore Mode (DSRM)** password section, in the **Password** text box, type **server@2012** as the password.
 - e. In the **Confirm password** text box, type **server@2012** as the password and select **Next**.
 - f. On the **DNS Options** page, select the **Show more** link.
 - g. In the **DNS Options** message box, select **OK**.
 - h. On the **DNS Options** page, select **Next**.
 - i. On the **Additional Options** page, in the **The NetBIOS domain name** text box, observe that the NetBIOS domain name is **ADWCYCLES** and select **Next**.
 - j. On the **Paths** page, accept the default path values and select **Next**.
 - k. On the **Review Options** page, observe the options that will be set up and select **Next**.
 - l. On the **Prerequisites Check** page, observe the message that indicates that all prerequisite checks have passed successfully. Select **Install** to begin installation. Wait for the installation to complete.
 - m. On the **Results** page, observe the message that indicates that the server was successfully configured as a domain controller.
 - n. Wait for Windows to restart and display the **Windows Login** screen. Press **Ctrl+Alt +End** to log in. In the Login screen, with the **User name** as **ADWCYCLES \Administrator** and enter the password of the administrator.
9. Configure DNS on Server2.
 - a. In the Server Manager window, select **Tools→DNS**. In the DNS Manager window, expand **SERVER2**, select and right-click **Reverse Lookup Zones** and select **New Zone**.
 - b. In the New Zone wizard, on the **Welcome to the New Zone Wizard** page, select **Next**.
 - c. On the **Zone Type** page, with the **Primary zone** option selected, select **Next**.
 - d. On the **Active Directory Zone Replication Scope** page, with the **To all DNS servers running on domain controllers in this domain: adwcycles.local** option selected, select **Next**.
 - e. On the **Reverse Lookup Zone Name** page, with the **IPv4 Reverse Lookup Zone** option selected, select **Next**.
 - f. On the **Reverse Lookup Zone Name** page, in the **Network ID** section, type **192.168.1** and select **Next**.
 - g. On the **Dynamic Update** page, with the **Allow only secure dynamic updates(recommended for Active Directory)** option selected, select **Next**.

- h. On the **Completing the New Zone Wizard** page, select **Finish**. Observe that the **1.168.192.in-addr.arpa** reverse lookup zone is added.
 - i. In the DNS Manager window, expand the **Forward Lookup Zones** folder and select the **adwcycles.local** zone.
 - j. Select **Action→New Host (A or AAAA)** and add A records with associated PTR record for **SQL01** with IP Address **192.168.1.230**, **Server3** with IP Address **192.168.1.227**, and **Server4** with IP Address **192.168.1.228**, and **mail** with IP Address **192.168.1.100**. Close the **DNSManager** window.
 - k. In the Network Connections window, right-click the **Ethernet** network adapter and select **Properties**.
 - l. In the **Ethernet Properties** dialog box, select the **Internet Protocol Version 4 (TCP/IPv4)** option and select **Properties**.
 - m. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, in the **Default gateway** text box, type **192.168.1.226** as the IP address.
 - n. With the **Use the following DNS sever addresses** option selected, in the **Preferred DNS server** text box, type **192.168.1.226** as the IP address, and select **OK**.
 - o. In the **Ethernet Properties** dialog box, select **Close**.
 - p. Close the Network Connections window.
 - q. In the Server Manager window, in the left pane, select **Local Server**.
 - r. Select **TASKS→Shut Down Local Server**.
 - s. In the **Shut Down Windows** dialog box, from the action drop-down list, select **Shut down** and from the **Option** drop-down list, select **Operating System: Reconfiguration(Planned)** and select **OK**.
 - t. Close the Server2 on SQL01 window.
 - u. In the Hyper-V Manager window, with the **Server2** VM selected, right-click and select **Settings**.
 - v. In the **IDE Controller 1** section, select **DVD Drive** and in the right pane, in the **Media** section, select the **None** option and select **OK**.
 - w. In the Hyper-V Manager window, with **Server2** VM selected, right-click and select **Start** and wait for the server to display a "Running" status. Right-click and select **Connect**.
10. Set up the Server3 VM in Hyper-V Manager on SQL01.
- a. Create a new VM named Server3.
 - a. On SQL01, in the Hyper-V Manager window, select **SQL01** and in the right pane, select the **New→Virtual Machine**.
 - b. In the New Virtual Machine wizard, on the **Before You Begin** page, select **Next**.
 - c. On the **Specify Name and Location** page, in the **Name** text box, type **Server3** as the name and select **Next**.
 - d. On the **Assign Memory** page, in the **Startup memory** section, type **1024** to allocate 1GB RAM.
 - e. Check the **Use Dynamic Memory for this virtual machine** check box and select **Next**.
 - f. On the **Configure Networking** page, from the **Connection** drop-down list, select **Internal**, and select **Next**.
 - g. On the **Connect Virtual Hard Disk** page, with the **Create a virtual hard disk** option selected, note the name and location that appear by default. In the **Size** text box, type **60 GB** and select **Next**.
 - h. On the **Installation Options** page, select the **Install an operating system from a bootCD/DVD-ROM** option. In the **Physical CD/DVD drive** drop-down list, select the drive-letter of your physical DVD ROM drive. If needed, insert the Windows Server 2012DVD into the DVD drive.



Note: Alternatively, if you have the .ISO image of Windows Server 2012 stored in a separate folder on your system, in this step, you need to select **Browse** and in the **Open** dialog box. Navigate to the folder containing the Windows 2012 ISO installation file. Select the .iso file and select **Open**.

- i. On the **Installation Options** page, select **Next**.
- j. On the **Completing the New Virtual Machine Wizard** page, select **Finish** to complete the New Virtual Machine wizard steps.
- b. Install Windows 2012 on Server3 in Hyper-V Manager.
 - a. In the Hyper-V Manager window, in the **Virtual Machines** section, select the **Server3** VM, right-click and select **Start** to start the virtual machine. Wait for **State** to change from **Off** to **Running** and right-click and select **Connect**.
 - b. In the Virtual Machine Connection window, the message “Connecting to Server3” is displayed first, followed by the **Windows Setup** dialog box.
 - c. In the **Windows Setup** dialog box, in the **Windows Server 2012** section, verify the settings.
 - a. In the **Language to install** drop-down list, verify that **English (United States)** language is selected.
 - b. In the **Time and currency format** drop-down list, verify that the **English (United States)** format is selected.
 - c. In the **Keyboard or input method** drop-down list, verify that the **US** method is selected. Select **Next** to proceed with the installation.
 - d. In the Windows Server 2012 section, select **Install now** to start the installation.
 - e. In the **Windows Setup** dialog box, on the **Enter the product key to activate Windows** dialog box, in the product key text box, type the product key and select **Next**.
 - f. On the **Select the operating system you want to install** page, select the second option **Windows Server 2012 Standard (Server with a GUI)** and select **Next**.
 - g. On the **License terms** page, check the **I accept the license terms** check box and select **Next**.
 - h. On the **Which type of installation do you want** page, select the **Custom: Install Windows only (advanced)** option.
 - i. On the **Where do you want to install Windows** page, verify that **Drive 0 unallocated space** is selected and select **Next**.
 - j. Wait for the installation to complete. Note the details displayed on the **Installing Windows** page.
 - k. Wait for Windows to automatically restart the system.
 - l. Wait for Windows to set up devices and display the Getting ready status. Allow Windows to restart the system.
 - m. On the **Settings** page, in the **Password** text box, type the password for the administrator.
 - n. In the **Reenter password** text box, type the same password to confirm the password for the administrator and select **Finish**.
 - o. Wait for Windows to finalize the settings and display the Windows Login screen.
 - c. For the Server3 VM, specify the IP address as 192.168.1.227 and modify the other settings such as System Name, Firewall Properties, and Internet Security Enhanced Configuration.
 - a. Press **Ctrl+Alt+End** to log in.
 - b. In the Login screen, type the **User name** as **Administrator** and type the **Password** as **pass@word1**.
 - c. In the Server Manager window, in the left pane, select the **Local Server** option. In the Properties for Server section, in the **Ethernet** section, select the existing Ethernet link.
 - d. In the Network Connections window, right-click the **Ethernet** network adapter and select **Properties**.

- e. In the **Ethernet Properties** dialog box, on the **Networking** tab, select the **Internet Protocol Version 4 (TCP/IPv4)** option and select **Properties**.
 - f. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, select the **Use the following IP address** option.
 - g. In the **IP address** text box, type **192.168.1.227** as the IP address and press **Tab**.
 - h. In the **Default gateway** text box, type **192.168.1.226** as the IP address.
 - i. With the **Use the following DNS sever addresses** option selected, in the **Preferred DNS server** text box, type **192.168.1.226** as the IP address and select **OK**.
 - j. In the **Ethernet Properties** dialog box, uncheck the **Internet Protocol Version 6 (TCP/IPv6)** check box and select **Close**.
 - k. Close the Network Connections window.
 - l. In the Server Manager window, verify that Local Server is selected in the left pane. In the right pane, in the **IE Enhanced Security Configuration** section, select the **On** link. In the **Internet Explorer Enhanced Security Configuration** dialog box, in the **Administrators** section, select **Off**. In the **Users** section, select **Off** and select **OK**.
 - m. In the Server Manager window, in the **Windows Firewall** section, select the **Public: On** link. In the Windows Firewall window, in the left pane, select **Advanced settings**. In the Windows Firewall with Advanced Security window, select **Windows Firewall Properties**. In the resultant dialog box, on the **Domain Profile**, **Private Profile**, and **Public Profile** tabs, from the **Firewall** state drop-down list, select **Off** and select **OK**. Close the Windows Firewall with Advanced Security window and Windows Firewall window.
 - n. In the Server Manager window, in the left pane, in the **Computer name** section, select the existing computer name link.
 - o. In the **System Properties** dialog box, select **Change**.
 - p. In the **Computer Name/Domain Changes** dialog box, in the **Computer name** text box, with the existing computer name selected and type **Server3** as the new name.
 - q. In the **Computer Name/Domain Changes** dialog box, in the **Member of** section, select the **Domain** option and type **adwcycles**.
 - r. Select **OK** to join the system to the **adwcycles.local** domain.
 - s. In the **Window Security** dialog box, enter the name and password as **Administrator** and **server@2012** and select **OK**.
 - t. In the **Computer Name/Domain Changes** message box, observe that the system is successfully joined to the domain and select **OK**.
 - u. In the **Computer Name/Domain Changes** message box, select **OK** to close the message box and restart the system to join the system to the adwcycles.local domain.
 - v. In the **System Properties** dialog box, select **Close**. In the **Microsoft Windows** dialog box, select **Restart Now** to restart the system.
 - w. Press **Ctrl+Alt+End** to log in, and in the Login screen, specify the **User name** as **adwcycles\Administrator** and enter the password of the administrator.
 - x. In the Server Manager window, in the left pane, select **Local Server** and select **TASKS→Shut Down Local Server**. In the **Shut Down Windows** dialog box, from the action drop-down list, select **Shut down** and from the **Option** drop-down list, select **Operating System: Reconfiguration (Planned)** and select **OK**.
 - y. Close Server3 on the SQL01 window.
 - z. In the Hyper-V Manager window, with the **Server3** VM selected, right-click and select **Settings**. In the **IDE Controller 1** section, select **DVD Drive** and in the right pane, in the **Media** section, select the **None** option and select **OK**.
11. Set up the Server4 VM in Hyper-V Manager on SQL01.
- a. Create a new VM named Server4.
 - b. Install Windows 2012 on Server4 in Hyper-V Manager.
 - c. Specify the IP address as 192.168.1.228, connect it to the 192.168.1.226 default gateway and DNS server.

- d. Modify the other settings such as System Name, Firewall Properties, and Internet Security Enhanced Configuration similar to settings you specified for the Server 3 VM.
 - e. Change the computer name to Server4 and connect it to the adwcycles.local domain using the Administrator account.
 - f. Shut down the server and remove the DVD drive from the Server4 IDE Controller.
12. After logging in as Administrator, shut down the Server2 VM.

Install the Course Data Files

From the location you downloaded the course data files to, open the 093108 folder. Then, open the Data folder. Run the 093108dd.exe file in that folder, which will install a folder named 093108Data at the root of your C drive. This folder contains all the data files you will need to run this course. There is a separate folder with the starter files for each lesson or lab, and there may be a Solutions folder with completed files students can use to check their results.

Presentation Planners

The lesson durations given in the course content are estimates based on a typical class experience. The following planners show examples of how the content could be presented in either a continuous one-day flow or separately across a multi-session seminar series. Your presentation flow may vary based on a number of factors, including the size of the class, whether students are in specialized job roles, whether you plan to incorporate LearnTOs or other assets from the LogicalCHOICE Course screen into the course, and so on. Use the samples and blank planners to determine how you will conduct the class to meet the needs of your own situation.

Continuous Presentation: Model Class Flow

This planner provides a sample presentation flow based on one 6.5-hour day of training, with breaks and lunch factored in.

Section	Duration	Day Planner
Day 1		
Welcome and Introductions	0:15	8:30 - 8:45
Lesson 1: Identifying the SQL Server Platform and Tools	1:30	8:45 - 10:15
BREAK	0:15	10:15 - 10:30
Lesson 2: Deploying SQL Server	1:30	10:30 - 12:00
LUNCH	0:30	12:00 - 12:30
Lesson 3: Configuring SQL Server	1:30	12:30 - 2:00
BREAK	0:15	2:00 - 2:15
Lesson 4: Managing Databases in SQL Server 2012	1:15	2:15 - 3:30
Day 2		
REVIEW	0:15	8:30 - 8:45
Lesson 5: Managing SQL Server Security	1:30	8:45 - 10:15
BREAK	0:15	10:15 - 10:30
Lesson 5: Managing SQL Server Security (Cont.)	1:00	10:30 - 11:30
Lesson 6: Implementing Advanced Security Settings	1:00	11:30 - 12:30

Section	Duration	Day Planner
LUNCH	0:30	12:30 - 1:00
Lesson 6: Implementing Advanced Security Settings (Cont.)	1:30	1:00 - 2:30
BREAK	0:15	2:30 - 2:45
Lesson 7: Applying Encryption and Compression	1:15	2:45 - 4:00
Day 3		
REVIEW	0:15	8:30 - 8:45
Lesson 8: Working with Indexes and Log Files	1:45	8:45 - 10:30
BREAK	0:15	10:30 - 10:45
Lesson 8: Working with Indexes and Log Files (Cont.)	1:00	10:45 - 11:45
LUNCH	0:30	11:45 - 12:15
Lesson 9: Working with Backup and Restore	2:00	12:15 - 2:15
BREAK	0:15	2:15 - 2:30
Lesson 10: Implementing High Availability	1:30	2:30 - 4:00
Day 4		
REVIEW	0:15	8:30 - 8:45
Lesson 10: Implementing High Availability (Cont.)	1:30	8:45 - 10:15
BREAK	0:15	10:15 - 10:30
Lesson 10: Implementing High Availability (Cont.)	1:30	10:30 - 12:00
LUNCH	0:30	12:00 - 12:30
Lesson 10: Implementing High Availability (Cont.)	1:30	12:30 - 2:00
BREAK	0:15	2:00 - 2:15
Lesson 11: Optimizing Server Performance	1:45	2:15 - 4:00
Day 5		
REVIEW	0:15	8:30 - 8:45
Lesson 11: Optimizing Server Performance (Cont.)	1:00	8:45 - 9:45
BREAK	0:15	9:45 - 10:00
Lesson 12: Troubleshooting Issues and Recovering Databases	2:15	10:00 - 12:15
LUNCH	0:30	12:15 - 12:45
Lesson 13: Performing Advanced Database Management Tasks	1:30	12:45 - 2:15

Section	Duration	Day Planner
BREAK	0:15	2:15 - 2:30
Lesson 13: Performing Advanced Database Management Tasks (Cont.)	1:15	2:30 - 3:45
REVIEW & QUESTIONS	0:30	3:45 - 4:15

Continuous Presentation: Your Class Flow

Use this planner to plan the flow of your own training day, based on the needs of your students, the schedule for your own day, and/or any other modifications you choose.

Section	Duration	Day Planner
Welcome and Introductions		
Lesson 1: Identifying the SQL Server Platform and Tools		
Lesson 2: Deploying SQL Server		
Lesson 3: Configuring SQL Server		
Lesson 4: Managing Databases in SQL Server 2012		
Lesson 5: Managing SQL Server Security		
Lesson 6: Implementing Advanced Security Settings		
Lesson 7: Applying Encryption and Compression		
Lesson 8: Working with Indexes and Log Files		

Section	Duration	Day Planner
Lesson 9: Working with Backup and Restore		
Lesson 10: Implementing High Availability		
Lesson 11: Optimizing Server Performance		
Lesson 12: Troubleshooting Issues and Recovering Databases		
Lesson 13: Performing Advanced Database Management Tasks		

Non-continuous Presentation: Model Class Flow

This planner provides a sample presentation flow based on separate sessions presented over multiple days or weeks.

Session Number	Material Covered	Session Duration
One	Lesson 1: Identifying the SQL Server Platform and Tools Lesson 2: Deploying SQL Server	3:00
Two	Lesson 3: Configuring SQL Server Lesson 4: Managing Databases in SQL Server 2012	2:45
Three	Lesson 5: Managing SQL Server Security	2:30
Four	Lesson 6: Implementing Advanced Security Settings	2:30
Five	Lesson 7: Applying Encryption and Compression Lesson 8: Working with Indexes and Log Files	3:00
Six	Lesson 8: Working with Indexes and Log Files (Cont.) Lesson 9: Working with Backup and Restore	3:00
Seven	Lesson 10: Implementing High Availability	3:00
Eight	Lesson 10: Implementing High Availability (Cont.)	3:00

<i>Session Number</i>	<i>Material Covered</i>	<i>Session Duration</i>
Nine	Lesson 11: Optimizing Server Performance	2:45
Ten	Lesson 12: Troubleshooting Issues and Recovering Databases	2:15
Eleven	Lesson 13: Performing Advanced Database Management Tasks Follow-Up and Questions	3:00

Non-continuous Presentation: Your Class Flow

Use this planner to plan how you will present the course content, based on the needs of your students, your conventions for the number and length of sessions, and any other modifications you choose.

<i>Session Number</i>	<i>Material Covered</i>	<i>Session Duration</i>

About This Course

You have created databases and executed queries on them. You are now ready to perform database administration tasks such as installation of SQL Server instances, configuration of SQL Server databases and settings, monitoring database performance, controlling user access to databases, and database maintenance. In addition to these tasks, SQL Server 2012 provides many advanced features that enable you to manage databases efficiently. By familiarizing yourself with these tasks and features, you can plan, deploy, administer, and maintain SQL Server databases efficiently. In this course, you will perform database administration tasks.

Course Description

Target Student

This course is designed for Database Administrators who administer and maintain SQL Server databases. This course will be attended by database professionals who are familiar with SQL Server databases and who are responsible for administering and maintaining SQL Server databases as their primary job responsibility. In addition, this course can be attended by individuals who develop applications that access SQL Server database.

Course Prerequisites

To ensure your success in this course you should have some working knowledge of relational databases. You should also be comfortable working with the Microsoft Windows Server 2012 Operating System, and be able to use the core functionalities of Windows Server 2012. Also, working knowledge of Transact SQL queries is essential.

You can obtain this level of skills and knowledge by taking the following Logical Operations courses:

- Relational database design courses
- Windows Server 2012 courses
- *Microsoft® SQL Server® 2012: Database Querying (Exam 70-461)*

Course Objectives

Upon successful completion of this course, students will be able to administer SQL Server 2012 databases.

You will:

- Identify the SQL Server Platform and Tools
- Deploy SQL Server
- Configure SQL Server
- Manage Databases

- Implement Advanced Security Settings
- Apply Encryption and Compression
- Work with Indexes and Log Files
- Work with Backup and Restore
- Implement High Availability
- Optimize Server Performance
- Troubleshoot Issues and Recover Databases
- Perform Advanced Management Tasks

The LogicalCHOICE Home Screen



<http://www.lo-choice.com>

The LogicalCHOICE Home screen is your entry point to the LogicalCHOICE learning experience, of which this course manual is only one part. Visit the LogicalCHOICE Course screen both during and after class to make use of the world of support and instructional resources that make up the LogicalCHOICE experience.

Log-on and access information for your LogicalCHOICE environment will be provided with your class experience. On the LogicalCHOICE Home screen, you can access the LogicalCHOICE Course screens for your specific courses.

Each LogicalCHOICE Course screen will give you access to the following resources:

- eBook: an interactive electronic version of the printed book for your course.
- LearnTOs: brief animated components that enhance and extend the classroom learning experience.

Depending on the nature of your course and the choices of your learning provider, the LogicalCHOICE Course screen may also include access to elements such as:

- The interactive eBook.
- Social media resources that enable you to collaborate with others in the learning community using professional communications sites such as LinkedIn or microblogging tools such as Twitter.
- Checklists with useful post-class reference information.
- Any course files you will download.
- The course assessment.
- Notices from the LogicalCHOICE administrator.
- Virtual labs, for remote access to the technical environment for your course.
- Your personal whiteboard for sketches and notes.
- Newsletters and other communications from your learning provider.
- Mentoring services.
- A link to the website of your training provider.
- The LogicalCHOICE store.

Visit your LogicalCHOICE Home screen often to connect, communicate, and extend your learning experience!

How to Use This Book

As You Learn

This book is divided into lessons and topics, covering a subject or a set of related subjects. In most cases, lessons are arranged in order of increasing proficiency.

The results-oriented topics include relevant and supporting information you need to master the content. Each topic has various types of activities designed to enable you to practice the guidelines and procedures as well as to solidify your understanding of the informational material presented in the course. Procedures and guidelines are presented in a concise fashion along with activities and

discussions. Information is provided for reference and reflection in such a way as to facilitate understanding and practice.

Data files for various activities as well as other supporting files for the course are available by download from the LogicalCHOICE Course screen. In addition to sample data for the course exercises, the course files may contain media components to enhance your learning and additional reference materials for use both during and after the course.

At the back of the book, you will find a glossary of the definitions of the terms and concepts used throughout the course. You will also find an index to assist in locating information within the instructional components of the book.

As You Review

Any method of instruction is only as effective as the time and effort you, the student, are willing to invest in it. In addition, some of the information that you learn in class may not be important to you immediately, but it may become important later. For this reason, we encourage you to spend some time reviewing the content of the course after your time in the classroom.

As a Reference

The organization and layout of this book make it an easy-to-use resource for future reference. Taking advantage of the glossary, index, and table of contents, you can use this book as a first source of definitions, background information, and summaries.

Course Icons

Watch throughout the material for these visual cues:

<i>Icon</i>	<i>Description</i>
	A Note provides additional information, guidance, or hints about a topic or task.
	A Caution helps make you aware of places where you need to be particularly careful with your actions, settings, or decisions so that you can be sure to get the desired results of an activity or task.
	LearnTO notes show you where an associated LearnTO is particularly relevant to the content. Access LearnTOs from your LogicalCHOICE Course screen.
	Checklists provide job aids you can use after class as a reference to performing skills back on the job. Access checklists from your LogicalCHOICE Course screen.
	Social notes remind you to check your LogicalCHOICE Course screen for opportunities to interact with the LogicalCHOICE community using social media.
	Notes Pages are intentionally left blank for you to write on.

1

Identifying the SQL Server Platform and Tools

Lesson Time: 1 hour, 30 minutes

Lesson Objectives

In this lesson, you will:

- Describe the SQL Server 2012 features that are required to plan for an SQL Server installation.
- Create filegroups and partitions.

Lesson Introduction

You are familiar with the basics of database querying. Imagine that you are a database administrator of a large organization and you want to implement SQL Server® 2012 as the database management system for your organization. SQL Server 2012 provides a mission-critical data platform to administer databases in your IT infrastructure.

Before installing the SQL Server 2012 database management system, you have to identify the different components of SQL Server 2012 that you need, determine the requirements of your setup, select the right hardware platform, prepare for scalability, and plan for optimal performance and storage. In this lesson, you will identify the SQL Server platform and tools.

TOPIC A

Overview of SQL Server Features

Before deploying SQL Server 2012, you need to prepare for the installation. Preparing for an installation involves determining the present requirements and planning for the future. This enables you to estimate the load on the server and provides a way to implement a scalable architecture that meets the performance requirements. In this topic, you will describe the features required to plan for an SQL Server installation.

Platform Components

SQL Server 2012 involves a number of components that are used to store, convert, analyze, and report data. While some of these components are new in SQL Server 2012, others have minor and major enhancements.

Component	Description
Database Engine	Acts as a Relational Database Management System (RDBMS) and provides core services for storing, processing, and protecting data. SQL Server 2012 includes several enhancements to the database engine's core services.
Integration Services	Transforms data from SQL Server or from any other data sources into formats required for analysis and report generation.
Reporting Services	Enables you to create, manage, and deliver reports.
Analysis Services	Supports quick data analysis and data mining operations on cleaned and formatted data. Enables you to perform drill downs on database data.
Master Data Services	Provides master data management capabilities to deal with changes in master data.
SQL Server Management Studio	Provides an integrated interface for managing and querying SQL Server databases.
SQL Server Profiler	Enables you to profile queries, identify performance issues, and optimize queries to enhance performance.
Database Engine Tuning Advisor	Suggests options to improve performance by using benchmarks created by the SQL Server Profiler.
Data Quality Services	Enables you to build a knowledge base of data based on the quality and integrity of the data source and create rules to clean up data.
Data Tools	Includes a collection of tools that enable you to perform several data management operations such as data import and export, database back up, and restore.
Connectivity Components	Includes components and SDKs that aid communication between the server and the clients and facilitate application development.

The Service Broker



Service Broker

The *service broker* is an SQL Server 2012 component that provides queuing and messaging capabilities to the database engine. It enables client applications to send and receive asynchronous messages by

using extensions to T-SQL Data Manipulation Language (DML). Client applications can send messages to the same database, to a different database on the same SQL Server instance, or to another SQL Server instance.

A sender places messages in a queue and the service broker delivers messages in the queue to the destination. Once the messages reach their intended destination, the service broker activates queue readers on the receivers and sends a confirmation message to the sender. By using asynchronous messaging, the service broker enables applications to send messages to multiple receivers eliminating the need for developing solutions to handle data communication.

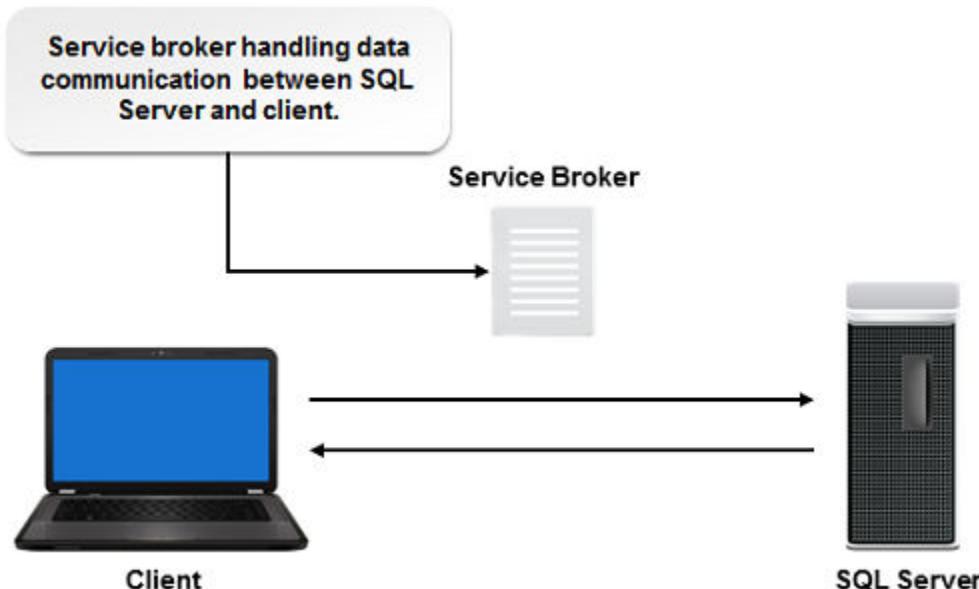


Figure 1-1: The service broker enables client applications to send and receive asynchronous messages.

Full Text Search

Full Text Search is an SQL Server 2012 feature that enables users to execute complex text-based queries on character-based data in tables. It executes queries on unstructured data stored in char, varchar, and nvarchar data types. Full Text Search queries can:

- Range from simple searches to more comprehensive linguistic ones that search for phrases or multiple forms of a word.
- Match terms that begin with a prefix.
- Match words that are closer to another word.
- Identify synonyms.
- Search for words using weighted values.

Before using Full Text Search on a table, a full-text index must be created on the table. Database administrators create full-text indexes on one or more character-based columns in a table. These indexes allow users to perform linguistic queries on text-based data depending on the rules of a specific language.

A Full Text Search returns a list of documents that have at least one match of all the terms specified in the query. Database administrators can also select the way through which results are retrieved and returned. They can also optimize the performance of Full Text Search queries depending on speed and accuracy requirements.



Ask the students to identify situations in which they would optimize a Full Text Search for speed and situations in which they would optimize the search for accuracy.

SQL Server Installation Requirements

To install SQL Server 2012 successfully, you need to ensure that your system meets the minimum installation requirements listed in the table.

Component	Minimum Requirements
Processor speed	1.0 - 1.5 GHz processor depending on whether it is a 32-bit or 64-bit processor. For modern relational database systems, a 64-bit processor is highly recommended.
Memory capacity	6-8 GB hard disk space and 1GB RAM. Depending on the number of databases, size of the databases, and the number of users, the memory capacity has to be increased.
Operating system	Windows Server 2008 R2 SP1, Windows Server 2012, Windows 7 SP1, or Windows 8.
.NET Framework	.Net 3.5 SP1 and .Net 4.0 are required to start the product installation.
PowerShell	Windows PowerShell 2.0 or 3.0 on Windows Server 2012 or Windows 8 is necessary to perform management tasks.
Networking	TCP/IP is recommended for remote communication.
Browser	Internet Explorer 7 or higher is recommended because it is used by many components such as Books Online Help, Microsoft Management Console (MMC), and Reporting Services.

In addition to the typical installation requirements, you can also enable Hyper-V or install any other hyper-virtualization software to run SQL Server on virtual machines. However, before enabling virtualization, it is necessary to check the compatibility of the virtualization package and verify whether it can support SQL Server virtually.



Ask the students to refer to the minimum requirements of SQL Server 2012 installation on Microsoft's website. Discuss the advantages of using 64-bit processors and a 64-bit operating system.

Scalability Methods

Scalability is a factor that determines the performance of applications when their capacity is increased by increasing the number of resources the server uses. There are two scalability methods.

Scalability Method	Description
Scale up	Involves adding resources such as memory, processors, and storage to a server to increase the capacity at which it runs. For instance, if a system has 4GB of RAM and the RAM capacity is increased to 16GB, you can add databases and users to SQL Server by changing the settings within the server. This method is also called <i>Vertical Scaling</i> .
Scale out	Involves adding hardware to distribute load and increase the capabilities of each server in the setup. For instance, if 1000 users are able to connect simultaneously to one server, by adding the server to a 3-node cluster, you can scale the capacity of the server to three times its original capacity. This method is also called <i>Horizontal Scaling</i> .



Ask the students to identify the differences between the two scalability methods. Provide instances and ask them to identify the scalability method they would use in a given scenario.

Virtualization involves both the scalability methods. The scale up method is required to support the virtualization host that handles multiple guests by increasing the server's processing and storage capacities. The scale out method is required to increase the capacities of the physical hosts.

Guidelines to Design an SQL Server Installation

SQL Server 2012 is made up of several components that you can install on different servers. Designing the SQL Server setup will enable you to develop and deploy a scalable architecture of SQL Server components based on your requirements. Some of the best practices for designing an SQL Server installation include:

- Selecting the components that you want to install and determine the servers on which you want to install them.
- Deciding whether to install the database engine on a separate server and provide references to access the components.
- Deciding which services need to be installed on the different servers involved in your setup. If you decide to install services on different servers, determine mechanisms that enable communication and resource sharing between the services.
- Selecting the scalability method that meets your requirements.
- Determining the hardware requirements and plan for future needs.

Guidelines for Capacity Planning

Capacity planning is a task that database administrators perform to estimate the expected load on a database or server and plan a server deployment according to the estimation. In addition to the requirements of the current deployment, database administrators need to estimate capacity requirements based on projections for capacity increases in the future. Some of the guidelines for capacity planning include:

- Determining the number and size of the databases, processor and memory requirements, number of simultaneous users, and number of physical and virtual servers.
- Determining the speed of the IO and storage devices.
- Considering other aspects such as network speed and bandwidth and security requirements.
- Monitoring the performance of the system in a test environment over a specific period of time, extrapolating the findings to a real world scenario, and planning for capacity based on the results of your findings.
- Determining the number of instances to be installed on each server and finalizing the databases to be created within each instance.



Note: SQL Server can support 32,767 databases per instance and 50 instances per server.

Default and Named Instances

An instance is one installation of the SQL Server database engine. You can set up multiple instances of the database engine on the same physical system. Each instance is independent of the others. Whenever a client application connects to an instance, appropriate T-SQL commands are passed on to it.

The *default instance* is the default installation of the database engine that is set up when SQL Server is installed. It is always named "MSSQLSERVER" and is accessible by either directly using the machine name of the system for remote access or by using a period to refer to it locally. There can be only one default instance of SQL Server on a machine.

Named instances are additional instances that you can set up on the same system on which you installed the default instance. You can specify a custom name for a named instance and refer to it using that name. For example, if you named an instance as "INSTANCE1," you can refer to it as "SERVER1\INSTANCE1." You can install different editions or versions of SQL Server on each named instance. You can also assign different administrators and management policies to each instance to segregate databases and control the way they are accessed.



Default and Named Instances

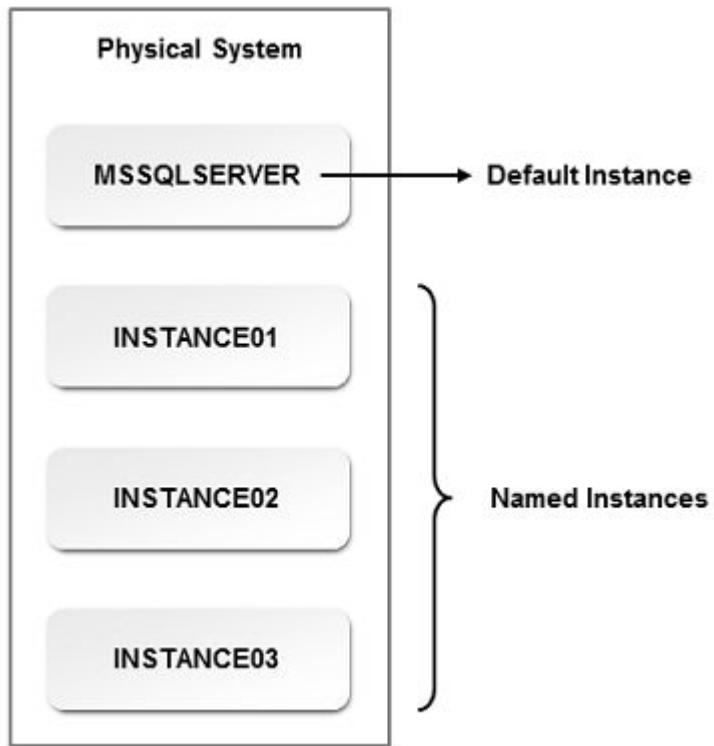


Figure 1–2: A physical system on which one default instance and multiple named instances are installed.



Ask the students to identify the differences between default and named instances. Ask them to identify the advantages of using multiple instances of the SQL Server database engine.

Guidelines to Select Hardware

Selecting the right hardware is an important step to be performed while planning for an SQL Server installation. Some of the best practices to select the right hardware include:

- Identifying the current requirements and estimate the future needs. SQL Server instances that service a high number of transactions and a large number of users require high processing power. Consider using multi-core processors or multiple small processors to achieve the required processing power.
- Determining the number of data pages of each database in an instance and the minimum, maximum, and average memory utilization for a typical query execution workload.
- Minimizing data page reads from disks by configuring SQL Server to cache a large number of data pages in memory.
- Determining the availability, reliability, and throughput requirements of the storage.
- Deciding the type of storage you require.
- Considering the query performance requirements of your databases and determining the suitable RAID level. Consider using redundancy options to prevent data loss.
- Considering power saving and budget requirements.

Guidelines to Decide the Type of Storage

Deciding the right type of storage is necessary to ensure smooth I/O operations on the databases in your servers. Some of the guidelines to decide the storage type are:

- Determining the number of servers required for the setup.
- Estimating the physical distance between the servers in the setup.

- Selecting a Direct-Attached Storage (DAS) if your setup requires a limited number of servers and the physical distance between the servers is limited. In a DAS, the disk drives are directly connected to the server.
- Selecting a Storage Area Network (SAN) if your set up requires a large number of servers that are geographically dispersed and if your future requirements need scalable storage architecture.

ACTIVITY 1–1

Identifying SQL Server Features

Scenario

You are the database administrator at Adventure Works Cycles. Before setting up the environment, you like to test your understanding on the SQL Server features.

1. Which SQL Server component enables you to build a knowledge base of data based on the quality and integrity of the data source and create rules to clean up data?
 - Data Quality Services
 - Data Tools
 - Integration Services
 - Analysis Services

2. What capabilities does the Service Broker provide to the database engine?
 - Data Manipulation capabilities
 - Querying capabilities
 - Searching capabilities
 - Querying and messaging capabilities

3. Which are the best practices you would follow while selecting hardware for your server setup? (Choose three.)
 - Identify the current requirements and estimate the future needs.
 - Determine the availability, reliability, and throughput requirements of the storage.
 - Minimize data page reads from disks by configuring SQL Server to cache a large number of data pages in memory.
 - Set up the hardware in such a way that the minimum requirements for a SQL Server installation are met.

4. Which of the following tasks can Full Text Search perform? (Choose three.)
 - Linguistic searches that find phrases or multiple forms of a word.
 - Match words that sound similar.
 - Match words that are closer to another word.
 - Identify synonyms.

5. Which are the best practices you would be following while planning for server capacity? (Choose three.)
 - Determine the hardware and software requirements and estimate the budget required to deploy the server setup.
 - Determine the number and size of the databases, processor and memory requirements, number of simultaneous users, and number of physical and virtual servers.
 - Monitor the performance of the system in a test environment over a specific period of time, extrapolate the findings to a real world scenario, and plan for capacity based on the results of your findings.
 - Determine the number of instances to be installed on each server and finalize the databases to be created within each instance.

TOPIC B

Create Filegroups and Partitions

You identified SQL Server features and examined their requirements. Before installing SQL Server, you need to design the physical implementation of databases and determine how data in the databases are to be stored. You can perform these tasks by creating filegroups and partitions. Familiarizing yourself with filegroups and partitions enables you to optimize the storage of data across multiple storage locations and improve the IO performance. In this topic, you will create filegroups and partitions.

Drives

Drives are the physical devices on which data in SQL Server databases is stored. The amount of storage capacity you require depends on the number and size of the databases. In addition to the databases, space is required to store indexes, system files, work spaces, virtual memory, and log files. You also need to consider allocating space for backups. Some of the best practices for designing storage using drives include:

- Analyzing the workload and estimating the number of read and write operations.
- Estimating the throughput necessary to support IO operations. Instead of using a single large drive, distribute data into several small drives. Use a RAID array of multiple disks to prevent data loss and enhance performance.
- Considering the use of a DAS if your implementation is small and involves a low budget. In case of geographically dispersed networks, use SANs to store data.
- Considering the need for data replication and planning for scalability, availability, and performance.
- Considering the use of Solid State Drives (SSD). These drives are faster and more reliable because they do not have moving parts.
- Supporting the storage set up with the right disk controllers and IO devices.

Solid State Drives

A *solid state drive* is a storage device that uses flash memory to store data. It is different from the traditional hard disk drives because it does not contain electromechanical parts or spinning disks. Solid state drives provide higher performance than hard disk drives because they have lower access times.

Files

Files store the data in SQL Server databases. SQL Server uses three types of files — primary data files, secondary files, and log files.



Files

Type of File	Description
Primary data files	Serves as the starting point of a database and points to other files those constitute the database. Primary data files have the extension .mdf. Every database must have one primary data file.
Secondary data files	Comprises of all the data files other than the primary data files in a database. Secondary data files are not mandatory. They have the extension .ndf.
Log files	Contains log information necessary to restore a database. A database must have atleast one log file. Log files have the extension .ldf.

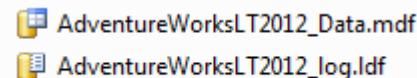


Figure 1–3: A primary data file and log data file associated with the AdventureWorksLT2012 database.



Filegroups

Filegroups are collections of files that are organized based on storage allocation and administration considerations. They allow database administrators to manage the set of files of a database on different drives. Every database has a default filegroup that stores data whenever tables and indexes are created without specifying a filegroup. SQL Server uses two types of filegroups—PRIMARY and user-defined.

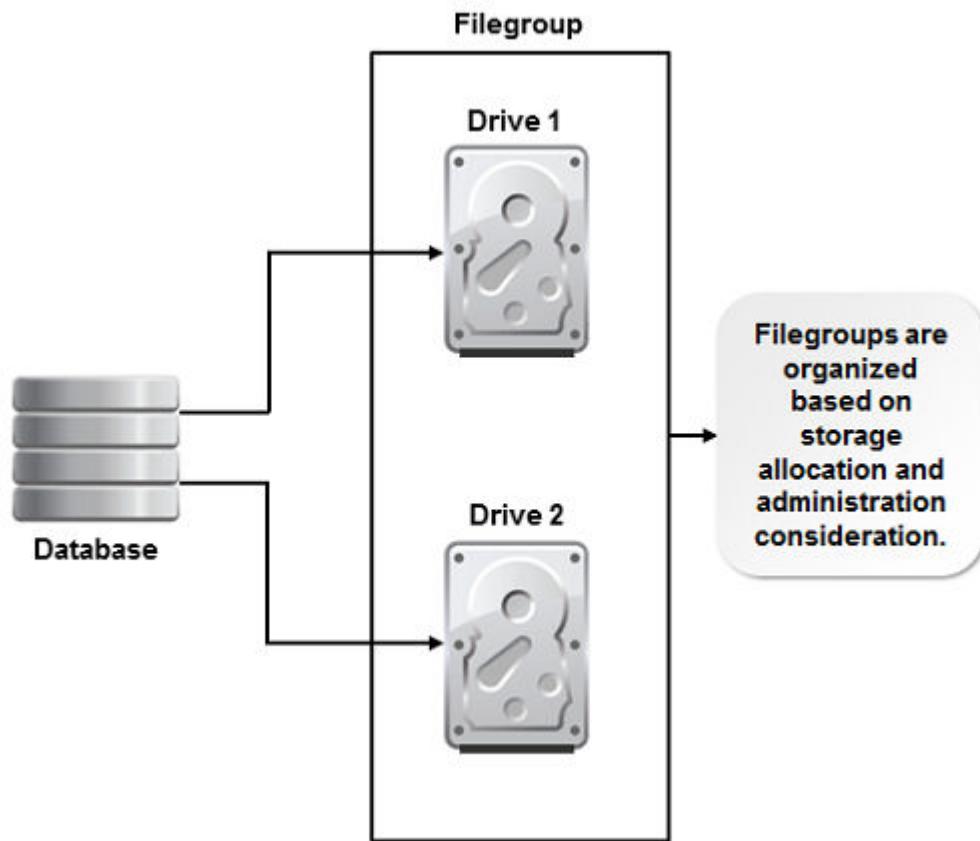


Figure 1–4: A database stored in two drives that belong to one filegroup.

Type of Filegroup	Description
Primary	Contains the primary data file and other files that are not assigned to any other filegroup. This filegroup hosts all pages of system tables and serves as the default filegroup if no filegroup is designated as the default filegroup.
User-defined	Contains secondary data files. You can designate a user-defined filegroup as the default filegroup.



Note: A file can be a member of one filegroup only.

Advantages of Using Filegroups

Using filegroups enhances the performance of tables in your databases. While creating a table, you can specify on which filegroup the table has to be created. For instance, you can create a table called "SalesOrders" on the primary filegroup. This table can be stored in different files on different drives. By using filegroups, you can distribute this table across multiple locations. Distributing the table across different filegroups enhances the performance of queries to this table.



Note: For more information about filegroups, check out the LearnTO **Administer Database Storage using Filegroups** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.

Partitions

Partitions form the basic unit of organization in the physical implementation of tables and indexes. Partitioning distributes tables and indexes across different filegroups. It enables you to manage subsets of data by mapping groups of table rows to individual partitions. By default, tables in SQL Server are stored in a single partition. However, as the number of rows in a table increases, the number of transactions also increases leading to degradation in database performance. Partitioning helps enhance database performance by splitting large tables into multiple partitions based on usage patterns.

For instance, consider a SalesOrders table that has a large number of rows and involves a huge number of transactions. There is also a usage pattern, which indicates that users access the current year's records more frequently than the previous year's records. You can improve the performance of this table by partitioning the table in such a way that the current year and the previous year's data are easily accessible without scanning the entire table. You can do this by creating a partition based on the values in the date column.



You may want to show LearnTO **Administer Database Storage using Filegroups** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



Partitions

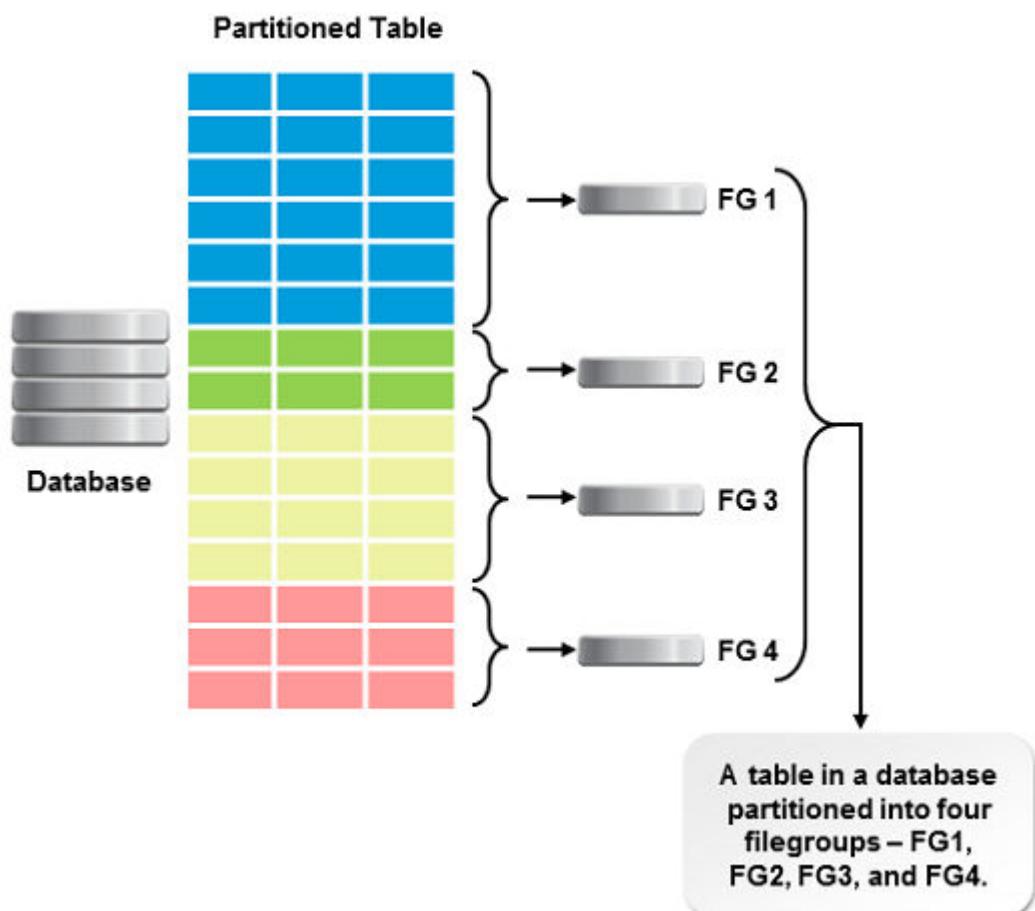


Figure 1–5: Partitions enable you to manage subsets of data.



Ask the students to explain how partitioning helps enhance database performance in large organization and also how partitioning helps design databases to meet the demands of future expansions.

The CREATE PARTITION FUNCTION Statement

The CREATE PARTITION FUNCTION statement is used to create a partition function that maps the rows of a table into partitions. You can use the CREATE PARTITION FUNCTION statement as the first step to create a partitioned table.

The syntax of the CREATE PARTITION FUNCTION statement is as follows.

```
CREATE PARTITION FUNCTION <partition_function_name> (parameter_type)
AS RANGE [ LEFT | RIGHT ]
FOR VALUES (boundary_value1, boundary_value2,...boundary_value n)
```

In this syntax, `<partition_function_name>` is the name of the partition function which is unique within a database. The `(parameter_type)` is the data type of the column used for partitioning. Boundary values specify the boundaries at which a new partition is created. If boundary value is empty, then all the rows are mapped into one partition. `[LEFT | RIGHT]` specifies whether rows are sorted in ascending or descending order of boundaries.

Example

A CREATE PARTITION FUNCTION statement to create a partition named date_partfunc is as follows.

```
CREATE PARTITION FUNCTION date_partfunc(datetime)
AS RANGE LEFT
```

```
FOR VALUES ('20101231 23:59:59.999', '20111231 23:59:59.999', '20121231  
23:59:59.999');
```

This statement creates a partition function with datetime parameter type that creates four partitions in which data from the starting date in the table until "12/31/2010" is mapped to the first partition, data from "12/31/2010" to "12/31/2011" is mapped to the second partition, data from "12/31/2011" to "12/31/2012" is mapped to the third partition, and the rest of the data is mapped to the fourth partition.

The CREATE PARTITION SCHEME Statement

The CREATE PARTITION SCHEME statement is used to create a scheme that maps the partitions of a partitioned table to one or more filegroups. You can use the CREATE PARTITION SCHEME statement only after using the CREATE PARTITION FUNCTION statement.

The syntax of the CREATE PARTITION SCHEME statement is as follows.

```
CREATE PARTITION SCHEME <partition_scheme_name>  
AS PARTITION partition_function_name  
[ALL] TO (filegroup_name | PRIMARY)
```

In this syntax, <partition_scheme_name> is the name of the partition function which is unique within a database. "AS PARTITION" refers to the partitions to be created based on a partition function. The partition_function_name is the name of the partition function that uses the partition scheme. ALL specifies that all partitions map to a single filegroup specified by the filegroup_name or the PRIMARY filegroup. The filegroup_name specifies the name of the file group that holds the partitions.

Example

A CREATE PARTITION SCHEME statement to create a partition scheme named data_partscheme is as follows.

```
CREATE PARTITION SCHEME data_partscheme  
AS PARTITION data_partfunc  
TO (fgrp1, fgrp2, fgrp3, fgrp4);
```

This statement creates a partition scheme named data_partscheme that uses the data_partfunc partition function to create a scheme and stores each partition to the specified filegroups as fgrp1, fgrp2, fgrp3, and fgrp4 respectively.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Create Filegroups and Partitions

ACTIVITY 1–2

Creating Files and Filegroups in a Database

Scenario

As the database administrator in a bicycle manufacturing company named Adventure Works Cycles, you are asked to create a Sales database for Adventure Works Cycles. Because you expect this database to contain several tables with hundreds of rows, you expect huge volumes of transactions. To improve query performance you wish to distribute query load across multiple files and drives for certain parts of the database. To achieve this, you decide to add multiple files and place them in a different file group so that the load is distributed.

1. Launch the **SQL Server Management Studio**.

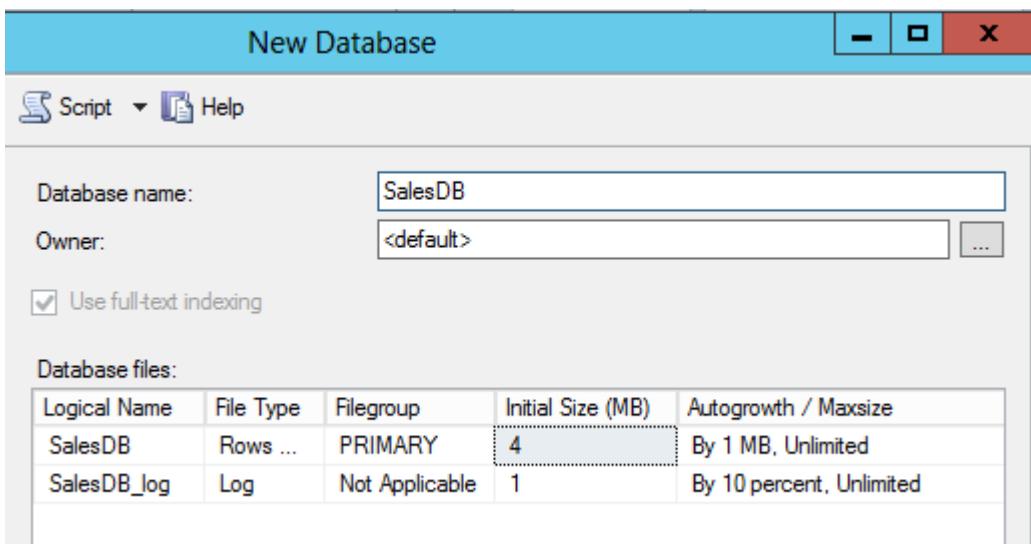
- In the desktop interface of Windows Server 2012, move your mouse pointer to the bottom-left corner of the screen and select **Metro Interface** to view the Start screen.
- In the Start screen, navigate to SQL Server Management Studio 2012.
- Select **SQL Server Management Studio** to launch it.
- In the **Connect to Server** dialog box, in the **Server Type** drop-down list, verify that **Database Engine** is selected and in the **Server name** drop-down list, verify that the name of your server is selected.
- In the **Authentication** drop-down list, verify that **Windows Authentication** is selected and select **Connect** to connect to the server.



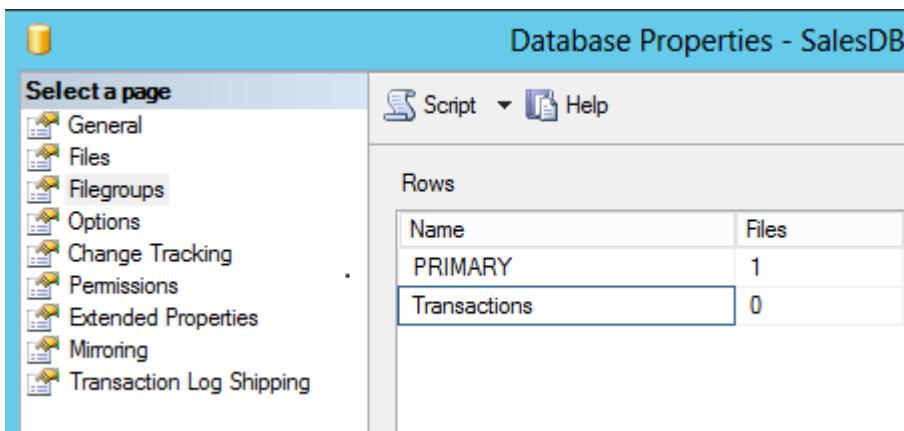
Verify that all the participants have been able to connect to the server before you move on to the next milestone.

2. Create a database.

- In the **Object Explorer** pane, right-click **Databases** folder and select **New Database**.
- In the **New Database** dialog box, in the **Database name** text box, type **SalesDB**.
- Observe that a new database with the name "SalesDB" is added to the PRIMARY filegroup.



- d) Select **OK** to create the database.
3. Create a filegroup for the "SalesDB" database.
- In Windows Explorer, create two folders, "Drive1" and "Drive2" under C: drive to simulate different storage locations.
 - In the **Object Explorer** pane, expand the **Databases** folder.
 - Right-click **Sales DB** and select **Properties**.
 - In the **Database Properties - SalesDB** dialog box, in the **Select a page** section, select **Filegroups**.
 - In the right pane, select the **Add** button to add a new filegroup.
 - Observe that in the **Rows** section, a blank row is added.
 - In the newly added row, in the **Name** column, type **Transactions**



4. Create a file in the "Transactions" filegroup.
- In the left pane, in the **Select a page** section, select **Files**.
 - In the right pane, select **Add** to add a new file.
 - In the newly added row, in the **Logical Name** column, type **Trans-01**
 - Select the **Filegroup** column and from the drop-down list, select **Transactions**.

Database files:				
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
SalesDB	Rows ...	PRIMARY	4	By 1 MB, Unlimited
SalesDB_log	Log	Not Applicable	1	By 10 percent, Limited to 209..
Trans-01	Rows ...	Transacti... ▾	4	By 1 MB, Unlimited
		PRIMARY		
		Transactions		
		<new filegroup>		

- e) Scroll right, in the **Path** column, select the ellipsis and in the dialog box that appears, navigate to C:\ and choose **Drive1** and select **OK**.
- f) In the **File Name** column, type **Trans-01** and click outside the row.
5. Create another file in the Transactions filegroup.
- Select **Add** to add another file.
 - In the newly added row, in the **Logical Name** column, type **Trans-02** and in the **Filegroup** column, select **Transactions**.
 - Scroll right and in the **Path** column, select the ellipsis and in the dialog box that appears, ensure that the dialog box navigates to C:\ and choose **Drive2**.
 - In the **File Name** column, type **Trans-02**, click outside the row and select **OK** to save the changes to the database.
 - In Windows Explorer, navigate to the C:\Drive1 and C:\Drive2 folders to verify that the two files are created.

ACTIVITY 1–3

Creating a Partitioned Table

Scenario

You have planned to create a SalesOrders table on the SalesDB database. You wish to store the data in the SalesOrders table into multiple partitions. By analyzing the usage patterns, you found that the current year's data is accessed most frequently, while the data of the past years is rarely accessed. Therefore, you wish to partition the data in such a way that sales orders are stored in three partitions. To maximize storage and query performance, you have planned to store each of these partitions in three different filegroups.



Note: You have already associated the PRIMARY filegroup and the Transactions filegroup with the SalesDB database. Now, you need to create a third filegroup to store the third partition.

1. Create a filegroup and create files within the newly created filegroup.
 - a) In the **Object Explorer** pane, right-click **SalesDB** and then select **Properties**.
 - b) In the left pane, in the **Select a page** section, select **Filegroups**.
 - c) In the right pane, select the **Add** button to add a new filegroup.
 - d) In the newly added row, in the **Name** column, type **LYTransactions**.
 - e) In the left pane, in the **Select a Page** section, select **Files**.
 - f) In the right pane, select **Add** to add a new file.
 - g) In the newly added row, in the **Logical Name** column, type **LYTrans**.
 - h) Select the **Filegroup** column and from the drop-down list that appears, select **LYTransactions**.

Database files:				
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
SalesDB	Rows ...	PRIMARY	4	By 1 MB, Unlimited
Trans-01	Rows ...	Transactions	4	By 1 MB, Unlimited
Trans-02	Rows ...	Transactions	4	By 1 MB, Unlimited
SalesDB_log	Log	Not Applicable	1	By 10 percent, Limited
LYTrans	Rows ...	LYTransactions	4	By 1 MB, Unlimited

- i) Set the path of the new filegroup to C:\Drive2 and name it as **LYTrans**.
 - j) Select **OK** to save the changes to the database.
2. Create a partition function.
 - a) In the **Object Explorer** pane, ensure that the **SalesDB** database is selected.
 - b) On the **Standard** toolbar, select the **New Query** button to create a new query.
 - c) In the **Query Editor** pane, type the CREATE PARTITION FUNCTION statement followed by the partition name Year_PartFunc with the DATE parameter type followed by the AS RANGE keyword.


```
CREATE PARTITION FUNCTION Year_PartFunc(DATE) AS RANGE
```
 - d) Enter the RIGHT keyword to specify that the rows are arranged in descending order, and then type the FOR VALUES keyword followed by the boundary values 20120101 and 20130101 to create a partition function that splits the data in the SalesDB database into three partitions based on values in the DATE column.

```
CREATE PARTITION FUNCTION Year_PartFunc(DATE) AS RANGE
    RIGHT FOR VALUES('20120101', '20130101');
```

- e) On the **SQL Editor** toolbar, select **Execute** to execute the query.
- f) On the **Messages** tab, observe that the message "Command(s) completed successfully" is displayed.
- g) Close the **Query Editor** pane without saving the query.

3. Create a partition scheme.

- a) On the **Standard** toolbar, select the **New Query** button to create a new query.
- b) In the **Query Editor** pane, type the CREATE PARTITION SCHEME statement followed by the partition scheme name Year_PartFunc with the DATE parameter type followed by the AS PARTITION keyword.

```
CREATE PARTITION SCHEME Year_PartScheme AS PARTITION
```

- c) Enter Year_PartFunc to specify the name of the partition function used by the partition scheme.

```
CREATE PARTITION SCHEME Year_PartScheme AS PARTITION
    Year_PartFunc
```

- d) Enter the TO keyword followed by the names of the filegroups on which the partitions are to be stored.

```
CREATE PARTITION SCHEME Year_PartScheme AS PARTITION
    Year_PartFunc
    To([PRIMARY], LYTransactions, Transactions);
```

- e) On the **SQL Editor** toolbar, select **Execute** to execute the query and partition the table.
- f) On the **Messages** tab, observe that the message "Command(s) completed successfully" is displayed.
- g) Close the **Query Editor** pane without saving the query.

4. Type a query to create a partitioned table.

- a) On the **Standard** toolbar, select the **New Query** button to create a new query.
- b) Type CREATE TABLE statement to create a table named SalesOrders.

```
CREATE TABLE SalesOrders
```

- c) Enter the code to add the SalesOrderID column with the integer data type and the NOT NULL constraint. Add the IDENTITY property to this column to automatically increment values starting from the number 1.

```
CREATE TABLE SalesOrders (SalesOrderID int IDENTITY (1,1) NOT NULL,
```

- d) Enter the code to add the SalesDate column with the date data type and NOT NULL constraint.

```
CREATE TABLE SalesOrders (SalesOrderID int IDENTITY (1,1) NOT NULL,
    SalesDate date NOT NULL,
```

- e) Enter the code to add the ProductID column with the integer data type and NOT NULL constraint.

```
CREATE TABLE SalesOrders(SalesOrderID int IDENTITY (1,1) NOT NULL,
    SalesDate date NOT NULL, ProductID int NOT NULL,
```

- f) Enter the code to add the ListPrice column with the float data type and NOT NULL constraint.

```
"...
    SalesDate date NOT NULL, ProductID int NOT NULL,
    ListPrice float NOT NULL,
```

- g) Type the code to add the Quantity column with the integer data type and NOT NULL constraint.

```
"...
    SalesDate date NOT NULL, ProductID int NOT NULL,
    ListPrice float NOT NULL,Quantity int NOT NULL,
```

- h) Enter the code to add the TotalSalesAmount column with the float data type and NOT NULL constraint.

```
"...
    SalesDate date NOT NULL, ProductID int NOT NULL,
```

```
ListPrice float NOT NULL,Quantity int NOT NULL,
TotalSalesAmount float NOT NULL
```

- i) Enter the code to create the table on the Year_PartScheme partition scheme and partition it based on the values in the SalesDate column.

```
"...
TotalSalesAmount float NOT NULL)
ON Year_PartScheme(SalesDate);
```

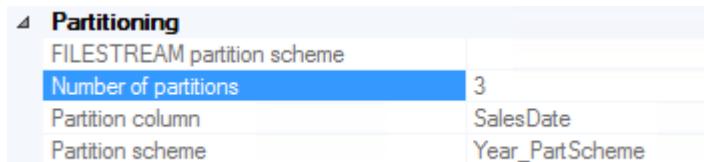
- j) On the **SQL Editor** toolbar, select **Execute** to execute the CREATE TABLE statement.

5. Execute the code to add primary key to the partitioned table.

- a) Open a **Query Editor** pane, enter ALTER TABLE statement followed by the table name SalesOrders along with CHECK ADD CONSTRAINT to add the PRIMARY KEY constraint on the SalesOrderID and the sorted SalesDate columns.

```
ALTER TABLE SalesOrders WITH CHECK ADD CONSTRAINT PK_SalesOrders
PRIMARY KEY (SalesOrderID, SalesDate ASC);
```

- b) On the **SQL Editor** toolbar, select **Execute** to execute the ALTER TABLE statement.
 c) In **Object Explorer** pane, expand the **SalesDB** database and then expand the **Tables** folder.
 d) Right-click **dbo.SalesOrders** and select **Properties**.
 e) In the **Table Properties - SalesOrders** dialog box, in the **Select a Page** section, select **Storage**.
 f) In the right pane, in the **Partitioning** section, verify that the number of partitions is 3 and select **OK**.



- g) Close the **Query Editor** panes without saving the queries.

Summary

In this lesson, you identified the features of SQL Server and created filegroups and partitions. This will enable you to determine the components of SQL Server that you need to install, plan for the physical implementation of databases, and organize your databases into filegroups and partitions.



Use the review questions provided to generate discussion among the participants.

What factors would you consider when selecting hardware for an SQL Server setup?

A: Answers will vary but may include: the current requirements and an estimate of the future needs, the number of transactions and number of users to be, the availability, reliability, and throughout requirements of the storage, query performance requirements, and the need for a 64-bit installation.

What are the benefits of storing data in multiple partitions in different storage locations?

A: Answers will vary but may include: enhances query performance, reduces time taken by eliminating the need to scan the entire table, provides quick access to frequently accessed data, supports multiple users effectively, and improves the performance of storage and IO.



Note: Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

2

Deploying SQL Server

Lesson Time: 1 hour, 30 minutes

Lesson Objectives

In this lesson, you will:

- Install SQL Server.
- Migrate SQL Server from a previous version.

Lesson Introduction

You have identified the SQL Server® set up requirements. You have also created the filegroups and partitions necessary to store the SQL Server databases. Now that you are all set, the next step is to implement the SQL Server set up. Before you begin the installation process, you need to decide whether you need to install SQL server as a new installation or whether you need to migrate data from an existing system to SQL Server 2012. In this lesson, you will deploy SQL Server.

TOPIC A

Install SQL Server

You are familiar with the SQL Server platform and tools. You are now ready to begin the SQL Server installation process. Careful examination of the steps in the installation process will help you select the right approach that suits your requirements and complete the installation successfully. In this topic, you will install SQL Server.

Guidelines for SQL Server Installation

Installing SQL Server involves several important tasks that require careful planning. There are some guidelines you can follow to perform a successful installation of SQL Server:

- Before starting the installation, configure the disk or storage volume that hosts the operating system. You can mirror the disk to protect the operating system in case the disk suffers a failure.
- Determine the security model you need to implement, select a suitable authentication method, and identify the required settings.
- Identify the features to be enabled during the installation.
- Determine the IO capacity required for the setup.

Guidelines to Configure an OS Disk for SQL Server 2012

Before installing SQL Server 2012, it is important to check and configure the basic configuration settings in the operating system (OS) disk.

- Ensure that your system has an NTFS formatted disk for installing SQL Server.
- If you are not installing SQL Server on the same disk as the OS, ensure that at least 4GB of empty disk space is available for temporary files on the OS disk before starting the installation of SQL Server.
- You can also use BitLocker on the drive that stores SQL Server and the database files. This ensures that even in the case that the disk is misplaced or stolen, unauthorized users will not be able to retrieve any data from the disk.
- Perform Redundant Array of Independent Disks (RAID) setups such as RAID 1, 5, 6, or 10 to ensure the reliability of the OS and SQL Server setup.

Guidelines to Implement Windows and Service Level Security

Before installing SQL Server, you need to consider the Windows and Service level security settings to be implemented for each server instance. You can implement multiple levels of security settings to protect SQL Server data. Here are some guidelines you can follow to implement a security model at multiple levels:

- Because SQL Server runs as a set of Windows services, you can configure each service to use its own account. At the time of installation, you can create service accounts for each service and assign appropriate access privileges to each service account. You can also select an authentication mode to determine how user access to the server is validated.
- Once installation is complete, create login accounts for users. You can create either a Windows login account or a SQL Server login account for each user.
- Associate user login accounts to databases. While a user with a Windows or SQL Server login account might be able to access the server, access to each individual database is maintained within the database. For example, user A with a Windows login account and access rights to database "DB1" may not have access rights to another database "DB2", while a user B with an SQL Server login account might have access to both DB1 and DB2.

Service Account for the Database Engine

The SQL Server Database Engine is a service that needs an account to run. Because it is important to protect the Database Engine from unauthorized access, it is preferable to create a service account with the least set of privileges that grants rights to access the service only. During the installation, the Installation wizard automatically suggests a new low-privilege account that it creates and maintains. You can accept the default account or create your own. For security purposes, care must be taken to ensure that this account is not used as a user login account.

Types of User Login Accounts

The Windows login account is a login account that is associated with a Windows or Active Directory user account. In this type of login account, authentication is taken care of by Windows or Active Directory Services and not by SQL Server. The SQL Server login account is a login account that is created and maintained by SQL Server.

Authentication Modes

SQL Server uses two authentication modes to enable users to log in to the server instance—**Windows Authentication mode** and **SQL Server and Windows Authentication mode**.



Authentication Modes

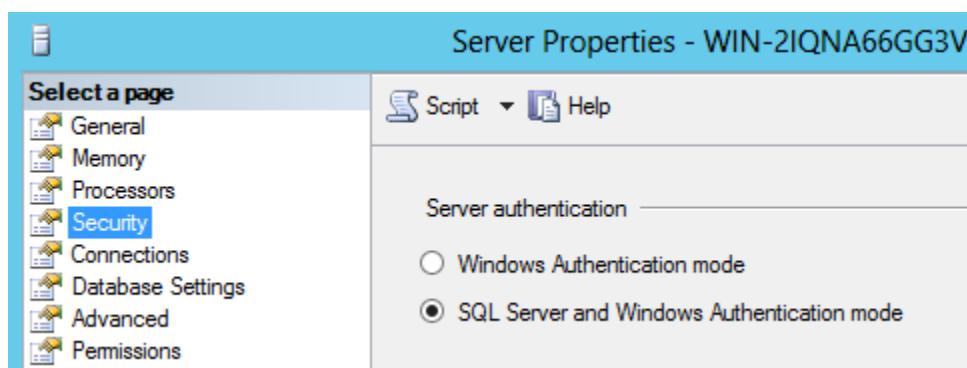


Figure 2-1: The Server Properties dialog box displaying two options for selecting the server authentication mode.

Authentication Mode	Description
Windows Authentication mode	Uses only Windows accounts or Active Directory accounts to log into SQL Server. In this mode, the operating system handles authentication to access the SQL Server instance. Each user who needs to access SQL Server can do so only by using a valid Windows/Active Directory account. This mode is preferred for accessing SQL Server because it provides a single sign-on access and simplifies login administration.
SQL Server and Windows Authentication mode	Uses both Windows accounts or Active Directory accounts and SQL Server accounts to log into SQL Server. In this mode, each user can access SQL Server using either a Windows account or an SQL Server account. This mode is suited where backward compatibility with an earlier version of SQL Server is necessary or when there is a specific Internet-based scenario that requires this type of authentication.



Note: SQL Server 2012 does not support an independent SQL Server authentication method. A Windows login is always required to access the server.



SQL Server Database Engine Instance Configuration



Caution: While using the SQL Server and Windows Authentication mode, you need to ensure that each SQL Server login account uses strong passwords because the passwords may be transmitted across the network to the server for authentication.

SQL Server Database Engine Instance Configuration

While installing SQL Server, you can install an instance of the SQL Server Database Engine by selecting it as one of the features to be installed. Once the Database Engine feature is selected, you can specify whether it has to be installed as a default or a named instance. If you install SQL Server for the first time, by default, the Installation Wizard selects the **Default instance** option for installation. However, you can change the default settings and select the **Named instance** option to install the Database Engine as a named instance.

Once all the required options are selected, the Installation Wizard starts the installation of the Database Engine service instance, creates a low-privilege user account for the instance, and maps the location of the databases for the instance.



Note: You can re-run the Installation wizard multiple times, select the **Named instance** option, and create multiple instances of the Database Engine. You can install a maximum of 50 instances of the Database Engine on a single machine

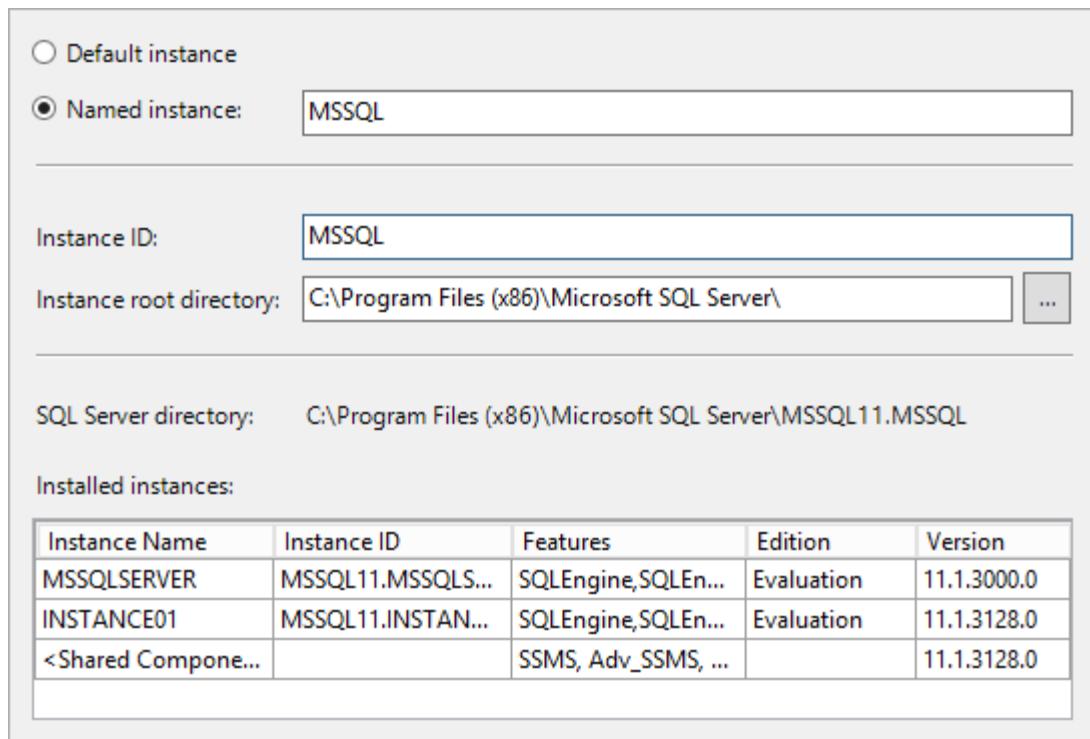


Figure 2-2: The *Instance Configuration* page of the SQL Server 2012 dialog box with options to select the type of instance to be installed.

SSIS

SQL Server Integration Services (SSIS) is an SQL Server component that you can install along with the other SQL Server components. When you select SSIS for installation, you can also install three shared features — SQL Server Data Tools, Management Tools, and Client Tools SDK.

Shared Features	Installs
SQL Server Data Tools	Tools that can be used for designing packages.
Management Tools	The SQL Server Management Studio.
Client Tools SDK	Managed assemblies to program Integrated Services.



Note: You can install only one instance of SSIS on a single machine.

The Server Core Mode

The *Server Core mode* is a minimal installation mode available in the Windows Server 2008 R2 and Windows Server 2012 operating systems. In this mode, only the least number of components required for the functioning of the server are installed. Performing an installation on the Server Core mode saves space and resources and provides a smaller attacker space. A Windows Server Core mode installation installs the command prompt and the Power Shell for server management.

SQL Server Installation in Server Core Mode

You can install SQL Server 2012 on Window Server in the Server Core mode. However, before installing SQL Server on the Server Core mode, you need to consider the following factors:

- While most SQL Server services can be installed and configured on the Server Core mode, there are a number of features of SQL Server 2012 that cannot be installed. These features are Reporting Services, SQL Server Data Tools, Client Tools, Management Tools, Distributed Replay Controller and Client, Master Data Services, and the Data Quality Client.
- Because SQL Server does not support a GUI option for the Server Core mode installation, you need to install SQL Server and its features by using the command prompt.

To install SQL Server in the Server Core mode, you need to provide values for each configuration option by using the switches listed in the table.

Switch	Description
/QS	Performs a simple and quiet installation.
/IACCEPTSQLSERVERLICENSETERMS	Confirms acceptance of EULA terms.
/ACTION=Install	Installs the features specified in the command.
/FEATURES=<featureset>	Specifies a list of comma separated values that represent the features to be installed. The values are SQLENGINE, REPLICATION, FULLTEXT, AS, IS, and CONN.
/INSTANCENAME=<name>	Specifies the name of the Database Engine instance.
/TCPENABLED=[0 1]	Determines whether TCP/IP is enabled. A 1 specifies that TCP/IP has to be enabled. TCP/IP is recommended for remote management.
/<OPTION>SVCACCOUNT=<account>	Specifies a service account for SQL and Analysis Services features.
/<OPTION>SVCPASSWORD=<pwd>	Specifies the password for the service account.
/<OPTION>ADMINACCOUNTS=<account>	Specifies the admin account for a feature.

- You may need to use a few more sub-switches and options depending on the SQL Server features you selected for the installation.
- All the options and switches for the Server Core mode installation can be saved as a configuration file. Using the configuration file you created, you can install multiple SQL Server instances on different servers.

Example

The following command installs the SQL Server Database Engine and SSIS on the Server Core mode.

```
SETUP.EXE /QS /IACCEPTSQLSERVERLICENSETERMS /ACTION=Install
FEATURES=SqlEngine, Is /INSTANCENAME=MSSQLSERVER /SQLSVCCOUNT='Fuller
\SQLService' SQLSVCPASSWORD='kk32c%12@dd' SQLSYSADMINACCOUNTS='Fuller
\SQLAdmin' /TCPENABLED=1
```

Remote Access

You can manage an SQL Server through remote access by using the Management Tools on a different machine. To access SQL Server remotely, you need to enable the remote access settings on SQL Server. As SSMS is not available on the Server Core mode, you need to enable remote access by executing commands through the command prompt. To do this, you need to execute the RUN SQLCMD.EXE command and then enter the following command:

```
EXEC sys.sp_configure N'remote access', N'1'
GO
RECONFIGURE WITH OVERRIDE
GO
```

After completing the remote access configuration, you can start the SQL Browser service using the following command:

```
SC CONFIG SQLBROWSER start=auto
NET START SQLBROWSER
```



Note: If the firewall is configured to allow connections through the default 1433 port, you can use the remote to connect to and manage SQL server from a different system.

SQLIO Tools

SQLIO tools are a set of free tools from Microsoft that enable you to test the IO performance of your storage system. These tools are used to determine the capacity of the IO subsystem. Testing the performance of the IO subsystem prior to deploying SQL Server is necessary to validate the functioning of the IO subsystem and identify performance problems. There are two main tools that you can use to test the IO performance—*SQLIO* and *SQLIOSim*.

Tool	Description
SQLIO	General purpose IO benchmarking tool used to test the storage system.
SQLIOSim	A tool that simulates SQL Server data access patterns and tests the IO performance of the storage system. This tool can be used to perform stress tests on the storage system.



Note: The SQLIO tools are available for free download from Microsoft's website. You can refer to the latest release notes in the documentation and download the latest version of these tools.

Installation Methods

You can install an SQL Server instance using one of the following methods:

- **Installing as a new instance:** While installing SQL Server, if you select the Database Engine for installation, you can install a new instance of SQL Server. If you are installing the Database Engine for the first time, by default, the Installation wizard selects the **Default Instance** option for installation. However, if you want to install the Database Engine as a named instance, you can change this setting.
- **Enabling features on an existing instance:** If you have already installed an SQL Server instance but need to add a feature that you missed installing earlier, you can easily do this by running the Installation wizard again. When the wizard starts up, it checks the system for existing installations. If it finds an instance, on the **Installation Type** screen, you can select the **Add features to an existing instance of SQL Server 2012** option to add more features to the existing instance. Once you select this option, you can select the desired instance or multiple instances on which you want to add features. The wizard determines the features already installed on the selected instances and lists the features you can add. While you cannot remove any installed component, you can add features by selecting them from the list.
- **Upgrading from an earlier version:** Upgrading is the process of updating the program files from a previous version to SQL Server 2012. SQL Server provides tools such as the **Upgrade Advisor** and the **Upgrade Assistant** that help you transition databases from an old version of SQL Server to SQL Server 2012 easily. You can perform an in-place upgrade to SQL Server 2012 from the following previous versions of SQL Server.
 - SQL Server 2005 with SP4
 - SQL Server 2008 with SP2
 - SQL Server 2008R2 with SP1



Installation Methods

To perform an upgrade, you need to run the **Upgrade Assistant** available in the SQL Server 2012 installation media. Depending on the Upgrade Assistant's report, you can complete the in-place upgrade. The upgrade process cannot add features to the installation. However, you can add features once the upgrade is completed.

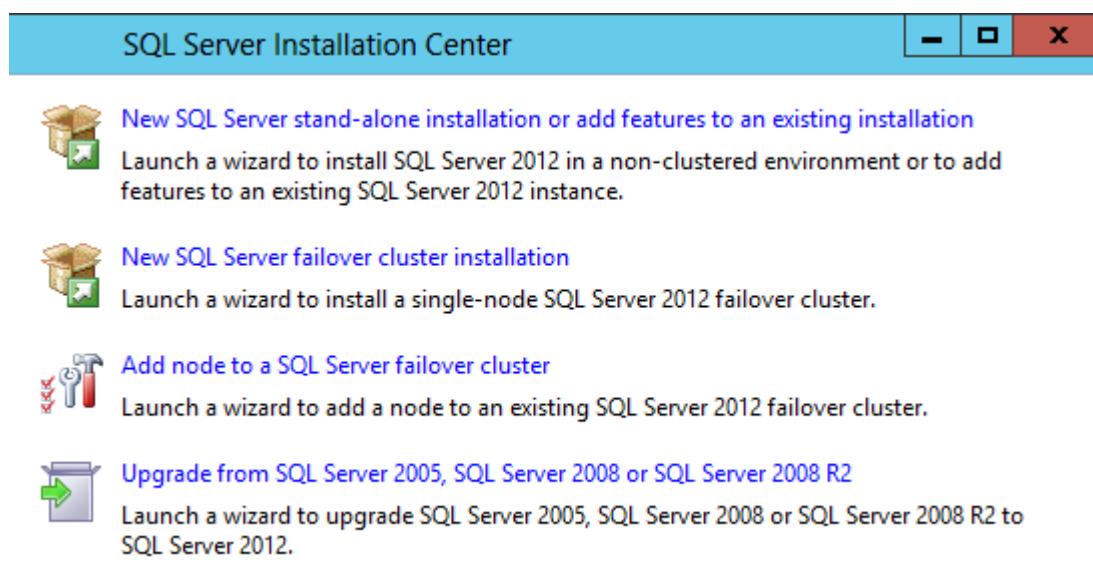


Figure 2-3: The SQL Server Installation Center displaying different options to deploy SQL server 2012.



Caution: You cannot upgrade a 32-bit version of SQL Server to a 64-bit version.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Install SQL Server

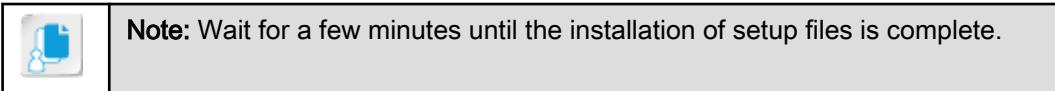
ACTIVITY 2–1

Installing a New Instance of SQL Server

Scenario

You have already installed a default instance of SQL Server on your system. However, you want to create another instance that runs only the Database Engine service. You want to name this instance as “INSTANCE01” so that you can switch to this instance whenever required.

1. Launch the Installation wizard.
 - a) Open Windows Explorer and navigate to the folder that contains the SQL Server installation components (C:\SQL Server 2012 Installation).
 - b) Double-click the **setup** file to launch the **SQL Server Installation Center**.
 - c) In the SQL Server Installation Center window, in the left pane, select **Installation**.
 - d) In the right pane, select the **New SQL Server stand-alone installation or add features to an existing installation** link.
-  **Note:** In the **SQL Server Setup** dialog box, in the **Setup Support Rules** identify problems that might occur during the installation. This operation might take a few minutes to complete. Once the **Setup Support Rules** page completes problem detection, a status indicating that the operation is complete appears.
- e) In the **SQL Server 2012 Setup** dialog box, select **OK**.
 - f) On the **Product Updates** page, select **Next**. Once you select **Next** on the **Product Updates** page, the installation process starts.



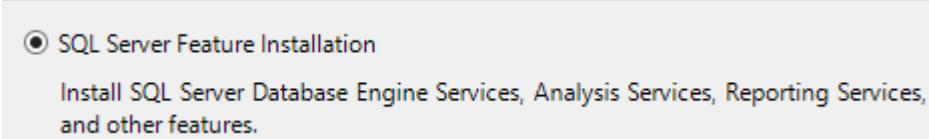
- g) In the **SQL Server 2012 Setup** dialog box, on the **Setup Support Rules** page, verify that no problems are detected and select **Next**.

2. Specify settings to start the installation of a named instance.

- a) On the **Installation Type** page, verify that the **Perform a new installation of SQL Server 2012** option is selected by default and select **Next**.



- b) On the **Product Key** page, verify that the **Specify a free edition** option is selected and select **Next**.
- c) On the **License Terms** page, check the **I accept the license terms** check box and select **Next**.
- d) On the **Setup Role** page, verify that the **SQL Server Feature Installation** option is selected and select **Next**.



Check with participants to ensure that they were able to progress to the **Installation Type** page of the **SQL Server 2012 Setup** dialog box without encountering any problems.



Note: Once you select **Next** on the **Installation Type** page, it takes a few minutes for the features to load on the **Feature Selection** page.

- e) On the **Feature Selection** page, in the list of features, check the **Database Engine Services** check box and select **Next**.

Features:	Feature description:
Instance Features <input checked="" type="checkbox"/> Database Engine Services <input type="checkbox"/> SQL Server Replication <input type="checkbox"/> Full-Text and Semantic Extractions for Search <input type="checkbox"/> Data Quality Services	The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.



Note: Once you select **Next** on the **Feature Selection** page, wait for the wizard to complete the task and return to the **SQL Server 2012 Setup** dialog box.

- f) On the **Installation Rules** page, select **Next**.

3. Configure the named instance.

- On the **Instance Configuration** page, in the **Named Instance** text box, type **INSTANCE01** and select **Next**.
- On the **Disk Space Requirements** page, observe the summary that describes the disk usage and select **Next**.
- On the **Server Configuration** page, select **Next**.
- On the **Database Engine Configuration** page, select **Add Current User**.
- Observe the current user's name is added to the list of SQL Server administrators and select **Next**.
- On the **Error Reporting** page, select **Next**.
- On the **Installation Configuration Rules** page, select **Next**.
- On the **Ready to Install** page, select **Install**.



Note: Once you select **Install**, the installation process begins. This could take a few minutes to complete.

- i) On the **Complete** page, verify that the installation has completed successfully and select **Close**.



Note: If a message box that displays 'Computer restart required' is displayed, select **OK** to restart the system.



Check with participants to ensure that they were able to progress to the **Ready to Install** page of the **SQL Server 2012 Setup** dialog box without encountering any problems.

Complete

Your SQL Server 2012 installation completed successfully with product updates.

Setup Support Rules
Installation Type
Product Key
License Terms
Setup Role
Feature Selection
Installation Rules

Information about the Setup operation or possible

Feature
Management Tools - Complete
Client Tools Connectivity
Client Tools SDK
Client Tools Backwards Compatibility
Management Tools - Basic

- j) Close the SQL Server installation Center window.



Note: Ensure that you have closed the SQL Server Installation Center window and the Drive F: window.

- k) In the Server Manager window, in the left pane, select **Local Server** and select **TASKS**→**Shut Down Local Server**.
- l) In the **Shutdown Windows** dialog box, from the action drop-down list, select **Restart** and from the Option drop-down list, select **Operating System: Reconfiguration (Planned)** and select **OK**.
- m) Press **Ctrl+Alt+End** to log in. In the Login screen, with the **User name** as **Administrator** type the **Password** as **pass@word1**.

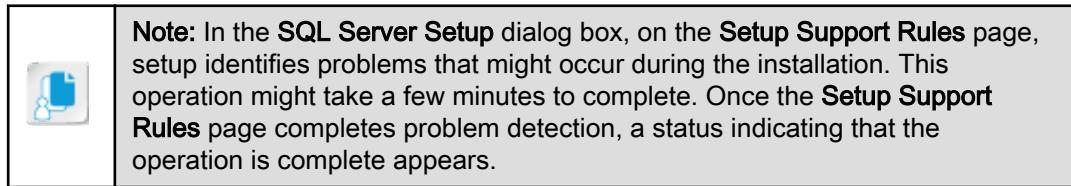
ACTIVITY 2–2

Adding Features to an Existing Instance of SQL Server

Scenario

You want to add the replication feature to the named instance of SQL Server that you created. Because you already added the Database Engine service, you want to enable the replication service on the same instance.

1. Launch the Installation wizard.
 - a) Open Windows Explorer and navigate to C:\SQL Server 2012 Installation.
 - b) Double-click the **setup** file to launch the **SQL Server Installation Center**.
 - c) In the SQL Server Installation Center window, in the left pane, select **Installation**.
 - d) In the right pane, select the **New SQL Server stand-alone installation or add features to an existing installation** link.

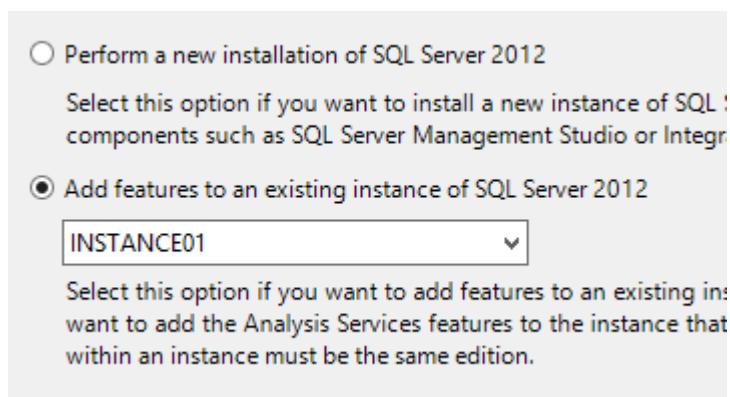


- e) In the **SQL Server 2012 Setup** dialog box, select **OK**.
 - f) On the Product Updates window, select **Next**.
- Note:** Once you select **Next** on the **Product Updates** page, the installation process starts. Wait for a few minutes until the installation of setup files is complete.
- g) In the SQL Server 2012 Setup window, in the **Setup Support Rules** page, verify that no problems are detected and select **Next**.

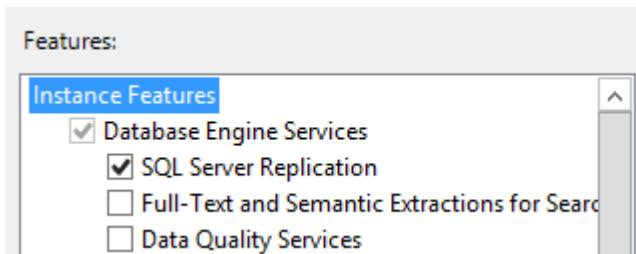


Check with participants to ensure that they were able to progress to the **Installation Type** page of the SQL Server 2012 Setup window without encountering any problems.

2. Specify settings to add features to an existing instance of SQL Server.
 - a) On the **Installation Type** page, select the **Add features to an existing instance of SQL Server 2012** option.
 - b) From the drop-down list below the **Add features to an existing instance of SQL Server 2012** option, select **INSTANCE01** and select **Next**.



- c) On the **Feature Selection** page, in the list of features, check the **SQL Server Replication** check box and select **Next**.



 **Note:** Once you select **Next** on the **Feature Selection** page, wait for the wizard to complete the task and return to the **SQL Server 2012 Setup** dialog box.

- d) On the **Installation Rules** page, select **Next**.
 e) On the **Disk Space Requirements** page, observe the summary that describes the disk usage and select **Next**.
 f) On the **Error Reporting** page, select **Next**.
 g) On the **Installation Configuration Rules** page, select **Next**.
 h) On the **Ready to Install** page, select **Install**.

 **Note:** Once you select **Install**, the installation process begins. This could take a few minutes to complete.

- i) On the **Complete** page, verify that the installation has completed successfully and select **Close**.
 j) Close the Installation Center window.
 k) Close Windows Explorer.



Check with participants to ensure that they were able to progress to the **Ready to Install** page of the **SQL Server 2012 Setup** dialog box without encountering any problems.

TOPIC B

Migrate SQL Server From a Previous Version

You have successfully installed an instance of SQL Server. Suppose you have huge volumes of data in a different database system and you need to move those data into SQL Server databases. SQL Server provides powerful solutions that help you migrate resources from an existing system to the new system in a reliable manner. In this topic, you will migrate SQL Server from a previous version.

Guidelines for Migrating Hardware from a Previous Version of SQL Server to SQL Server 2012

Migrating hardware from a previous version of SQL Server set up to SQL Server 2012 requires careful planning and implementation. There are some guidelines you can follow while migrating hardware from an old server setup to SQL Server 2012:

- Check whether you need to procure additional hardware before migrating hardware from the existing setup.
- Check whether the hardware supports the operating and SQL Server prerequisites.
- If you decide to migrate the server set up to new hardware, consider using 64-bit capable hardware with a 64-bit operating system and a 64-bit installation of SQL Server 2012. This will enhance the performance of your server setup.
- Consider future capacity expansions and provide room for RAM and storage expandability.



Discuss the benefits of migrating old servers from 32-bit to 64-bit operating systems.



You may want to show **LearnTO Recognize the Guidelines to Migrate to SQL Server 2012** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOS for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



The SQL Server Import Export Wizard

Note: For more information about migration, check out the **LearnTO Recognize the Guidelines to Migrate to SQL Server 2012** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.

Guidelines for Migrating Databases from Other Database Systems

SQL Server 2012 enables you to migrate databases from previous versions of SQL Server or from other databases systems. Migrating databases from other database systems requires adequate planning and preparation. There are some guidelines you can follow while migrating databases from other database systems to SQL Server:

- Each database system such as Oracle, MySQL, and PostgreSQL supports different data formats and programming languages. Before migrating databases from a different database system, perform adequate tests to ensure that client applications can access data without any conflicts.
- Use the **SQL Server Import and Export** wizard to connect to a different database system. This wizard enables you to connect to different database systems and migrate database objects from those systems to your SQL Server instance.
- If the required database system is not listed in the **SQL Server Import and Export** wizard, you can install the appropriate database provider or driver on the machine and run the wizard again.

The SQL Server Import and Export Wizard

The *SQL Server Import and Export wizard* is a component of Integration Services that enables you to copy data to and from different data sources. Using this wizard, you can copy data from an Oracle database to a PostgreSQL if a managed .NET database provider or an Open Database Connectivity (ODBC)/ Object Linking and Embedding Database (OLEDB) driver for those databases is installed. Typically, this wizard is used to either export from or import to an SQL Server database.

You can perform mapping between the table structures of the source and destination if they are from different providers. For example, you can map a VARCHAR column to a TEXT column if the source and destination use different providers.

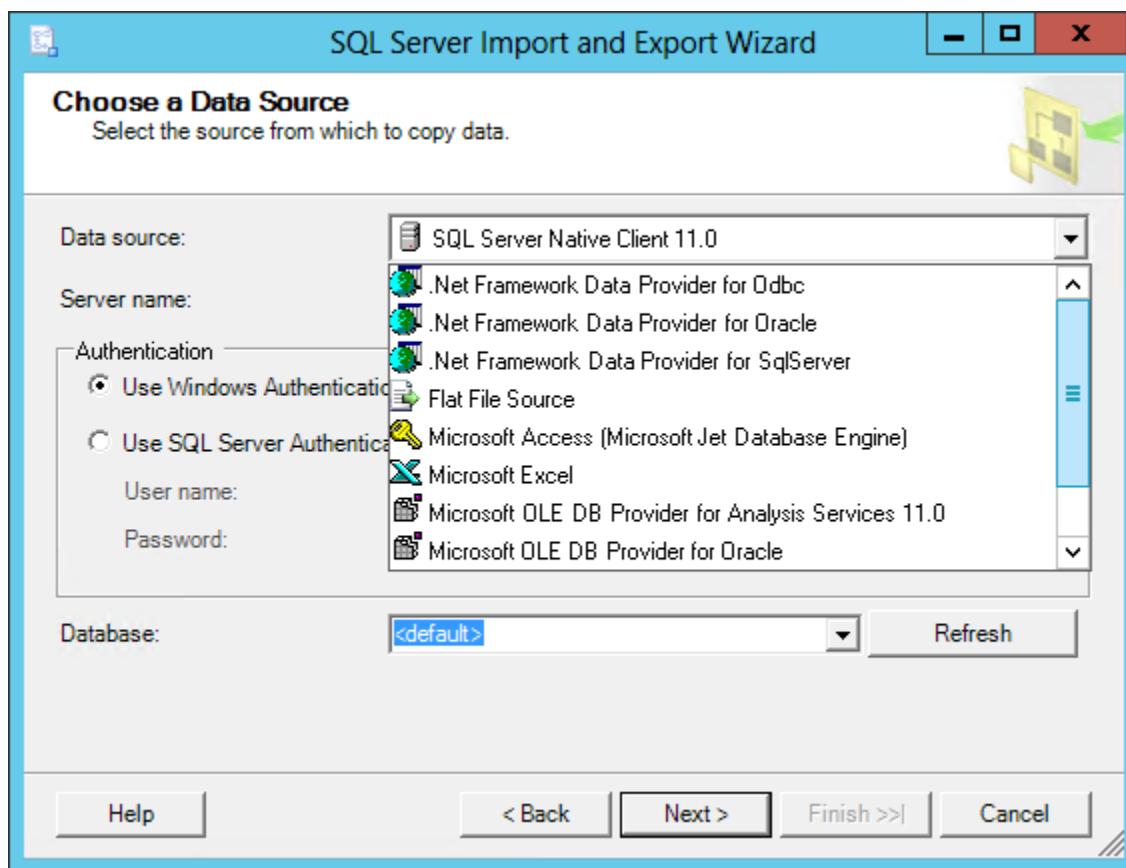


Figure 2–4: The SQL Server Import and Export wizard displaying data source names.

Permissions to Copy Data

You can use the **SQL Server Import and Export** wizard to move data only if you have appropriate read and write permissions on the source and destination databases. If you have the required permissions, then in the wizard, you can select the required database system and provide the necessary parameters required for the connection such as the connection string, access credentials, and database name. Once you are connected to the required database, you can copy the required database objects by selecting them through the wizard.



Ask the students to list the techniques they use to move data from old servers to new servers. Ask them to identify the advantages of using the Import/Export wizard to migrate data.

ACTIVITY 2–3

Identifying the Best Practices to Migrate from a Previous Version of the SQL Server to SQL Server 2012

Scenario

Your company maintains data in a previous version of SQL Server. Because management has decided to move the previous versions of SQL Server to SQL Server 2012, you want to identify the factors you need to consider before implementing the change.

1. **What steps would you perform before migrating hardware from a previous version of SQL Server to SQL Server 2012? (Choose three.)**
 - Check whether you need to procure additional hardware.
 - Verify whether hardware supports the operating and SQL Server prerequisites.
 - Buy hardware based on the lowest possible configurations and deploy SQL Server on it through trial and error.
 - Consider future capacity expansions and provide room for RAM and storage expandability.
2. **How does the SQL Server Import and Export wizard help migrate data?**
 - Connects to different data source systems and copies the required databases.
 - Verifies the connectivity between SQL Server 2012 and the old databases.
 - Moves data from SQL Server databases to storage devices.
 - Imports databases from IO devices.
3. **Which are the best practices you would follow before migrating a database from a previous version of SQL Server to SQL Server 2012? (Choose two.)**
 - Perform adequate tests to ensure that client applications can access data without any conflicts.
 - Use the SQL Server Import and Export wizard to connect to the old database system and copy the required database objects.
 - Convert the old databases to SQL Server 2012 database objects and then copy them to the new system.
 - Copy the old database files and place them within the new system.

Summary

In this lesson, you deployed SQL server. You can now install SQL Server 2012 as a new installation, upgrade it from an older version, or enable features in an existing installation.

What are the benefits of planning for an SQL Server installation?

A: Answers will vary, but may include: it helps in identifying the features to be installed, developing an estimate of the hardware and software requirements, developing an estimate of the storage requirements, estimating the future needs, and preparing for expansion.



Use the review questions provided to generate discussion among the participants.

What are the benefits of using the SQL Server Import and Export wizard for transferring data?

A: Answers may vary, but will include: copy large databases easily, export data to different systems, import data from different data sources, map table structures between the source and destination databases, and migrate databases from a previous version of SQL Server to SQL Server 2012.



Note: Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

3 | Configuring SQL Server

Lesson Time: 1 hour, 30 minutes

Lesson Objectives

In this lesson, you will:

- Configure server properties.
- Configure memory and CPU.
- Configure database mail.

Lesson Introduction

You have successfully deployed SQL Server® in your organization. To use the installed server resources efficiently, you need to configure the server based on your requirements. SQL Server provides many configuration options for the server instances. Familiarizing yourself with the server configurations will enable you to improve the performance, stability and reliability of your database system. In this lesson, you will configure SQL Server.

TOPIC A

Configure Server Properties

After installing the required server instances, you need to configure the server properties. The server properties can be configured using the server configuration options available in the SQL Server. You can access these configuration options using the T-SQL statements. A better understanding on configuring server properties will enable you to change the server properties based on your requirements. In this topic, you will configure server properties.



The `sp_configure` Command

The `sp_configure` is a system stored procedure that is used to display or modify the global configuration options for the currently connected server. It is used to view and modify the global settings for the current server. You can change the values of configuration option using the `sp_configure` command. However, to apply the changes on the server, you have to run the RECONFIGURE command immediately after changing the configuration option. The general syntax to modify the configuration options on the server is as follows.

```
sp_configure '<configuration option name>', <value>;
```

Example

The `sp_configure` stored procedure to return all the default configuration options of the server is as follows.

```
sp_configure;
```

This stored procedure will display the default configuration options of the server with their values for the following columns.

Column	Contains
<code>name</code>	The name of the configuration option.
<code>minimum</code>	The minimum value that the configuration option can accept.
<code>maximum</code>	The maximum value that the configuration option can accept.
<code>config_value</code>	The value currently configured using <code>sp_configure</code> .
<code>run_value</code>	The value that is currently running.

SQLQuery3.sql - SQ...Administrator (52))*

sp_configure

100 % < |

Results Messages

	name	minimum	maximum	config_value	run_value
1	allow updates	0	1	0	0
2	backup compression default	0	1	0	0
3	clr enabled	0	1	0	0
4	contained database authentication	0	1	0	0
5	cross db ownership chaining	0	1	0	0
6	default language	0	9999	0	0
7	filestream access level	0	2	0	0
8	max text repl size (B)	-1	2147483647	65536	65536
9	nested triggers	0	1	1	1
10	remote access	0	1	1	1
11	remote admin connections	0	1	0	0
12	remote login timeout (s)	0	2147483647	10	10
13	remote proc trans	0	1	0	0
14	remote query timeout (s)	0	2147483647	600	600
15	server trigger recursion	0	1	1	1
16	show advanced options	0	1	0	0
17	user options	0	32767	0	0

Figure 3-1: The result set of sp_configure command displaying the server configuration options.



Note: The values in the config_value and the run_value columns always need not be the same.

A sp_configure statement to alter the clr enabled option is as follows.

```
sp_configure 'clr enabled', 1;
RECONFIGURE;
```

This statement will set the clr enabled option to 1 and will enable you to execute the Common Language Runtime (CLR) assemblies.



Note: Some configuration options require the server to restart for applying the changes made.

Server-Level Settings

A server includes a number of configuration options that are hidden by default. You have to alter or modify these hidden configuration options to alter the advanced properties of the server. The advanced configuration options are displayed by executing the sp_configure stored procedure after changing the show advanced options to 1.



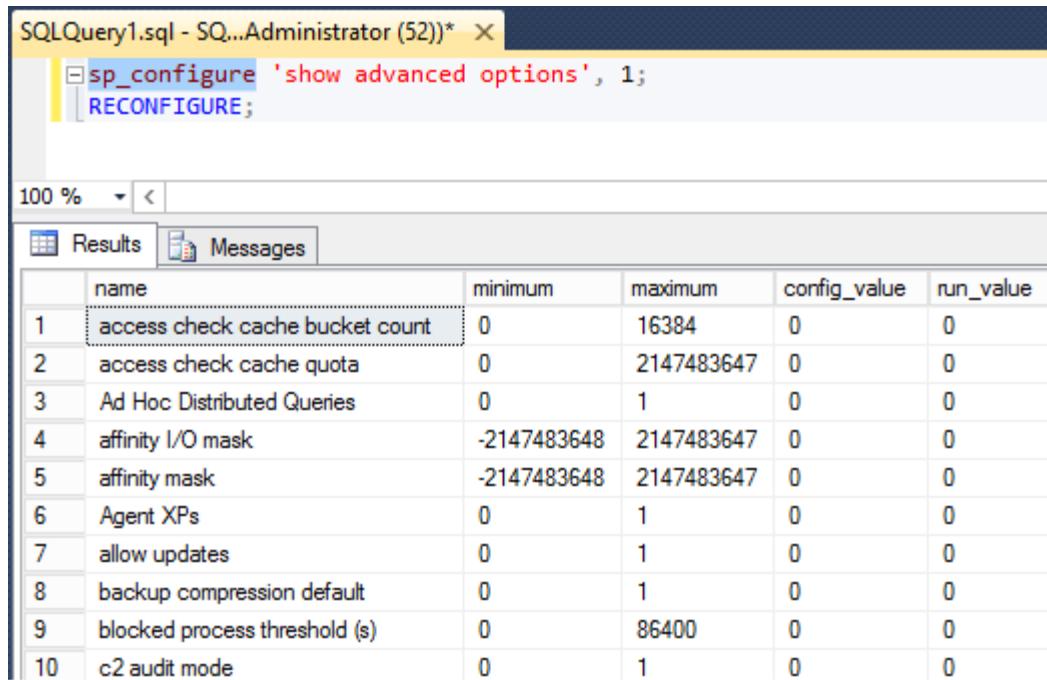
Server-Level Settings

Example

The sp_configure statement to display the advanced server configuration options is as follows.

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
```

This statement will display the hidden advanced options on the server.



The screenshot shows a SQL query window titled "SQLQuery1.sql - SQ...Administrator (52)*". The query is:

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
```

The results pane displays a table with the following data:

	name	minimum	maximum	config_value	run_value
1	access check cache bucket count	0	16384	0	0
2	access check cache quota	0	2147483647	0	0
3	Ad Hoc Distributed Queries	0	1	0	0
4	affinity I/O mask	-2147483648	2147483647	0	0
5	affinity mask	-2147483648	2147483647	0	0
6	Agent XPs	0	1	0	0
7	allow updates	0	1	0	0
8	backup compression default	0	1	0	0
9	blocked process threshold (s)	0	86400	0	0
10	c2 audit mode	0	1	0	0

Figure 3–2: The result set of the `sp_configure` command displaying the advanced server configuration options.



Note: It is recommended that you turn off the show advanced options configuration option by setting it to 0 as soon as you modify the advanced options.



Ask the students if they would prefer the "show advanced options" configuration option be turned on so that they can quickly access the advanced settings.

Default Server Options

The configuration options that are accessed by the `sp_configure` stored procedure have default values. The default values set are sufficient for non-production systems that are used for testing purposes. However, in production environments, the default values need to be changed for better performance and reliability.

Example

A `min memory per query` configuration option allows you to specify the minimum memory that needs to be allocated for the execution of a query. The option has a range from 512KB to 2147483647KB (2GB) and a default value of 1024 KB.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Configure Server Properties

ACTIVITY 3–1

Changing Advanced Settings Using the `sp_configure` Command

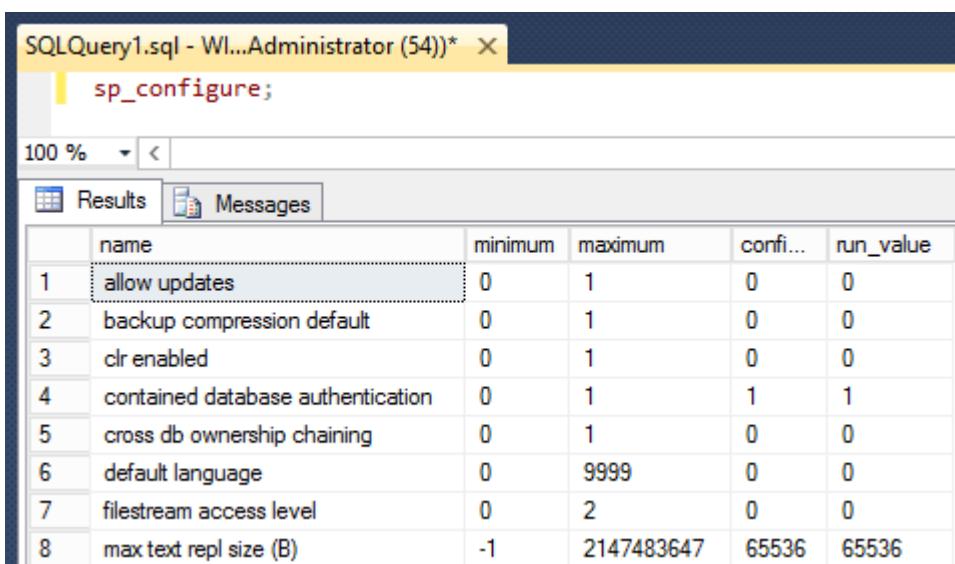
Before You Begin

Open SSMS and connect to the SQL01 instance.

Scenario

As the database administrator of Adventure Works Cycles, you are asked to configure the memory allocated for each query in the server to improve the performance of queries. You feel that increasing the minimum memory allocated to each query would help improve the query performance. You decide to configure the minimum memory for a query using the `sp_configure` procedure.

1. Execute the code to display the configuration options of the server.
 - a) In SSMS, open a **Query Editor** pane, type `sp_configure`
 - sp_configure;
 - b) Execute the `sp_configure` stored procedure.
 - c) On the **Results** tab, observe that the configuration options of the server are displayed. The result set does not include advanced options such as min memory per query.



	name	minimum	maximum	confi...	run_value
1	allow updates	0	1	0	0
2	backup compression default	0	1	0	0
3	clr enabled	0	1	0	0
4	contained database authentication	0	1	1	1
5	cross db ownership chaining	0	1	0	0
6	default language	0	9999	0	0
7	filestream access level	0	2	0	0
8	max text repl size (B)	-1	2147483647	65536	65536



Check with participants to ensure that they were able to observe that the show advanced options is set to 0 before you proceed.

2. Execute code to display the advanced server configuration options.
 - a) In the **Query Editor** pane, place the cursor after `sp_configure` command.
 - b) Type `'show advanced options'` in single quotes followed by a comma and value 1 to set the show advanced options configuration option to 1.

```
sp_configure 'show advanced options', 1;
```

- c) On the **SQL Editor** toolbar, select **Execute** to execute the configuration statement.

- d) On the **Messages** tab, observe that the message ‘Show advanced options is changed from 0 to 1’ is displayed.

SQLQuery1.sql - SQ...Administrator (52)* X
sp_configure 'show advanced options', 1;

100 % < Messages
Configuration option 'show advanced options' changed from 0 to 1.
Run the RECONFIGURE statement to install.



Select one or two participants, and ask them to summarize the show advanced options configuration option they accessed

- e) On the **Query Editor** pane, enter RECONFIGURE command to apply the changes to the server.
- ```
sp_configure 'show advanced options', 1;
RECONFIGURE;
```
- f) Select the RECONFIGURE command and on the **SQL Editor** toolbar, select **Execute** to turn on the show advanced options in the server.
- g) On the **Messages** tab, observe that the message “Command(s) executed successfully” is displayed.

SQLQuery1.sql - SQ...Administrator (52)\* X  
sp\_configure 'show advanced options', 1;  
RECONFIGURE;

100 % < Messages  
Command(s) completed successfully.

- h) On the **Query Editor** pane, select the sp\_configure command.
- i) Execute the sp\_configure command to display the updated server configuration options.
- j) On the **Results** tab, observe that the configuration options including the advanced options are displayed.

SQLQuery1.sql - WI...Administrator (54)\* X  
sp\_configure 'show advanced options',1;  
RECONFIGURE;

100 % < Results Messages

|   | name                            | minimum     | maximum    | config_value | run_value |
|---|---------------------------------|-------------|------------|--------------|-----------|
| 1 | access check cache bucket count | 0           | 65536      | 0            | 0         |
| 2 | access check cache quota        | 0           | 2147483647 | 0            | 0         |
| 3 | Ad Hoc Distributed Queries      | 0           | 1          | 0            | 0         |
| 4 | affinity I/O mask               | -2147483648 | 2147483647 | 1            | 1         |
| 5 | affinity mask                   | -2147483648 | 2147483647 | 2            | 2         |
| 6 | affinity64 I/O mask             | -2147483648 | 2147483647 | 0            | 0         |
| 7 | affinity64 mask                 | -2147483648 | 2147483647 | 0            | 0         |

3. Execute the code to change the min memory per query configuration option to 2048 KB.

- a) In the **Query Editor** pane, replace the show advanced options configuration option with min memory per query (KB) and value 1 with 2048.

```
sp_configure 'min memory per query (KB)', 2048;
RECONFIGURE;
```

- b) Execute the query to change the min memory per query configuration option.

- c) On the **Messages** tab, observe that the message stating that the min memory per query (KB) is changed from 1024 to 2048 is displayed.

The screenshot shows the SQL Query Editor with the following command:

```
sp_configure 'min memory per query (KB)', 2048;
RECONFIGURE;
```

In the Messages tab, the output is:

```
Configuration option 'min memory per query (KB)' changed from 1024 to 2048.
Run the RECONFIGURE statement to install.
```

- d) In the **Query Editor** pane, select the RECONFIGURE command.

- e) Execute the RECONFIGURE command to apply the changes to the server.

- f) On the **Messages** tab, observe that the message "Command(s) executed successfully" is displayed.

The screenshot shows the SQL Query Editor with the same command as before:

```
sp_configure 'min memory per query (KB)', 2048;
RECONFIGURE;
```

In the Messages tab, the output is:

```
Command(s) completed successfully.
```

4. Execute code to verify that the updated configuration option.

- a) In the **Query Editor** pane, select sp\_configure stored procedure.

- b) On the **SQL Editor** toolbar, select **Execute**.

- c) On the **Results** tab, scroll down to min memory per query record and verify that the value in the config\_value column is 2048.

The screenshot shows the SQL Query Editor with the command:

```
sp_configure 'min memory per query (KB)', 2048;
RECONFIGURE;
```

Switching to the Results tab, the output is:

|    | name                      | minimum | maximum    | config_value | run_value |
|----|---------------------------|---------|------------|--------------|-----------|
| 39 | min memory per query (KB) | 512     | 2147483647 | 2048         | 2048      |

- d) Close the **Query Editor** pane without saving the query.

# TOPIC B

## Configure Memory and CPU

You configured server properties of a server instance using the configuration options. While working with multiple instances of SQL Server, memory and CPU management gain more significance. SQL Server allows you to decide how much resources each instance can get from the system. You can configure the amount of memory as well as the number of CPUs that each instance can work on. Better understanding on configuring memory and CPU enables you to perform multi-tasking and use memory resources efficiently. In this topic, you will configure memory and Central Processing Unit (CPU).

### Affinity Masks

*Affinity mask* is a server configuration option that enables you to assign processes to specific CPUs. Windows Servers perform multitasking between different processes running on the system. The Windows operating system switches tasks between CPUs depending on the load on each CPU. However, in the case of SQL Server this can create a performance problem since each time a process switches to a different CPU, the processor cache has to be reloaded. In extremely high-use scenarios, this becomes a bottleneck. Affinity masks prevent this bottleneck by reducing the process switches among CPUs.

There are two types of affinity masks that you can change:

| Affinity Mask         | Description                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------|
| CPU Affinity Mask     | This defines the affinity or “stickiness” of a SQL Server computational process to a specified set of CPUs on the system. |
| CPU I/O Affinity Mask | This defines the affinity or “stickiness” of a SQL Server Input/Output process to a specified set of CPUs on the system.  |

### 64-bit Affinity Masks

In modern systems it is possible to have a large number of CPUs. In case the system has more than 32 CPUs, you need to use the 64-bit equivalent of affinity masks. The two 64-bit affinity mask configuration options are:

- Affinity64 mask
- Affinity64 I/O mask



Ask the students how assigning CPUs to SQL Server processes would help when there are multiple CPUs on a server.



SQL Server Affinity

### SQL Server Affinity

SQL Server affinity can be assigned to CPUs using the affinity mask configuration options. To change the affinity, you must assign a bit-wise value to each processor. If you have multiple CPUs on a system and you wish to assign SQL Server affinity to some of them, assign those CPUs a value of 1 and the others a value of 0. The bit-wise affinity values for an eight CPU system are calculated as a byte and represented as integer in the server configuration option.

The affinity values for assigning processor affinity to CPUs 1, 5, and 6 in an eight CPU system are as follows.

The bit positions represent the CPUs in an eight CPU system and bit values are assigned as affinity masks for the corresponding CPUs. Therefore the CPUs 1, 5, and 6 are assigned affinity mask of 1. The affinity mask values 00110001 when calculated as a byte represent integer 49.

The affinity values for assigning processor affinity to CPUs 2 and 4 in an eight CPU system are as follows. The CPUs 2 and 4 are assigned affinity mask of 1. The affinity values 00001010 when calculated as a byte represent integer 10.

| Server Affinity | C<br>P<br>U <sub>8</sub> | C<br>P<br>U <sub>7</sub> | C<br>P<br>U <sub>6</sub> | C<br>P<br>U <sub>5</sub> | C<br>P<br>U <sub>4</sub> | C<br>P<br>U <sub>3</sub> | C<br>P<br>U <sub>2</sub> | C<br>P<br>U <sub>1</sub> |
|-----------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Affinity mask   | 0                        | 0                        | 1                        | 1                        | 0                        | 0                        | 0                        | 1                        |
| Bit position    | 7                        | 6                        | 5                        | 4                        | 3                        | 2                        | 1                        | 0                        |

| Server Affinity | C<br>P<br>U <sub>8</sub> | C<br>P<br>U <sub>7</sub> | C<br>P<br>U <sub>6</sub> | C<br>P<br>U <sub>5</sub> | C<br>P<br>U <sub>4</sub> | C<br>P<br>U <sub>3</sub> | C<br>P<br>U <sub>2</sub> | C<br>P<br>U <sub>1</sub> |
|-----------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Affinity mask   | 0                        | 0                        | 0                        | 0                        | 1                        | 0                        | 1                        | 1                        |
| Bit position    | 7                        | 6                        | 5                        | 4                        | 3                        | 2                        | 1                        | 0                        |

**Figure 3-3: The affinity masks set for an eight bit CPU system.**



**Note:** You can change the affinity mask option only if the show advanced options configuration option should be set to 1.

## Example

The sp\_configure statements to change the affinity mask of CPUs 1,5, and 6 in an eight CPU system is as follows.

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
sp_configure 'affinity mask', 49;
RECONFIGURE;
GO
```



**Note:** The GO command denotes the end of the batch of T-SQL statements.

The first sp\_configure statement sets the show advanced options configuration option to 1 to modify the affinity mask advanced option. The second sp\_configure statement evaluates integer 49 to 00001010 and sets the affinity masks of CPUs 1,5, and 6 to 1.



**Note:** SQL Server 2012 deprecates the “affinity mask” and “affinity64 mask” configurations. You can use the ALTER SERVER CONFIGURATION command to assign the computational processes to CPUs.

The sp\_configure statements to change the affinity I/O mask of CPUs 2 and 4 are as follows.

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
sp_configure 'affinity I/O mask', 10;
RECONFIGURE;
GO
```

This first sp\_configure sets the show advanced options configuration option to 1 to modify the affinity I/O mask advanced option. The second sp\_configure statement evaluates integer 10 to 00001010 and sets the affinity I/O masks of CPUs 2 and 4 to 1.



**Note:** If the CPUs are not actually in the system, the masks will get reset to 0.

### Change CPU and I/O Affinity Masks from SSMS

You can also change the CPU and I/O Affinity Masks from SSMS using the **Server Properties**→**Processors** page. To set the affinity masks for the CPUs, you need to check the check boxes in the **Processor Affinity** and **I/O Affinity** columns.



**Note:** You need to uncheck the **Automatically set processor affinity mask for all processors** and **Automatically set I/O affinity mask for all processors** check boxes to change the affinity masks.



Ask the students which method they would prefer to change the server settings-using the SSMS or sp\_configure stored procedure and why.

| Processor | Processor Affinity                  | I/O Affinity                        |
|-----------|-------------------------------------|-------------------------------------|
| ALL       | <input type="checkbox"/>            | <input type="checkbox"/>            |
| NumaNode0 | <input type="checkbox"/>            | <input type="checkbox"/>            |
| CPU0      | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| CPU1      | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |

*Figure 3-4: The Processors page of Server Properties window displaying the affinity settings for CPUs in a system.*

## The ALTER SERVER CONFIGURATION Statement

CPU process affinity can also be set using the ALTER SERVER CONFIGURATION statement.

The general syntax for assigning affinity using the ALTER SERVER CONFIGURATION statement is as follows.

```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY CPU=<CPU set or Range
(0-based)>
```

The PROCESS AFFINITY keyword enables the computational processes to be assigned to the CPUs specified. The CPUs specified can be individual CPUs, range of CPUs or Non Uniform Memory Access (NUMA) enabled CPUs. The CPUs are specified as integer values starting from 0.

## Example

The ALTER SERVER CONFIGURATION statement to change the affinity mask of CPUs 0,4, and 5 is as follows.

```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY CPU=0,4,5;
```

This statement assigns the computational processes to CPUs 0, 4, and 5 by setting their process affinity masks.

The ALTER SERVER CONFIGURATION statement to change the affinity mask of NUMA enabled CPUs is as follows.

```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY NUMANODE=0,2;
```

The NUMA nodes specified in the statement represent a set of CPUs. Each NUMA set is a 64-CPU block starting from CPU 0. This statement will assign NUMANODES 0 and 2 consisting of CPUs 0-63 and 128-191 respectively, to the SQL Server processes.

## NUMA

NUMA is a multiprocessor memory architecture in which memory blocks called NUMA nodes can be attached to a set of CPUs. The CPUs can access their NUMA nodes faster than accessing other non-local memory.

## Memory Allocation

While running multiple instances on a server, *memory allocation* is done to ensure that each instance gets enough memory and does not exceed the maximum amount of memory defined. This enables you to keep the server running even if any instance tries to use too much memory in unexpected conditions such as executing an incorrect query. Unlike SQL Server 2005, 2008, and 2008R2, SQL Server 2012 has only two types of memory allocation.

| <b>Memory Allocation</b>            | <b>Description</b>                                                                                                                                   |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Single-page allocations             | SQL Server 2012 allocates memory by combining single and multi-page allocations into a single-page allocator that handles any size page allocations. |
| Common Language Runtime allocations | SQL Server 2012 allocates memory for CLR processing.                                                                                                 |

## Server Memory Options

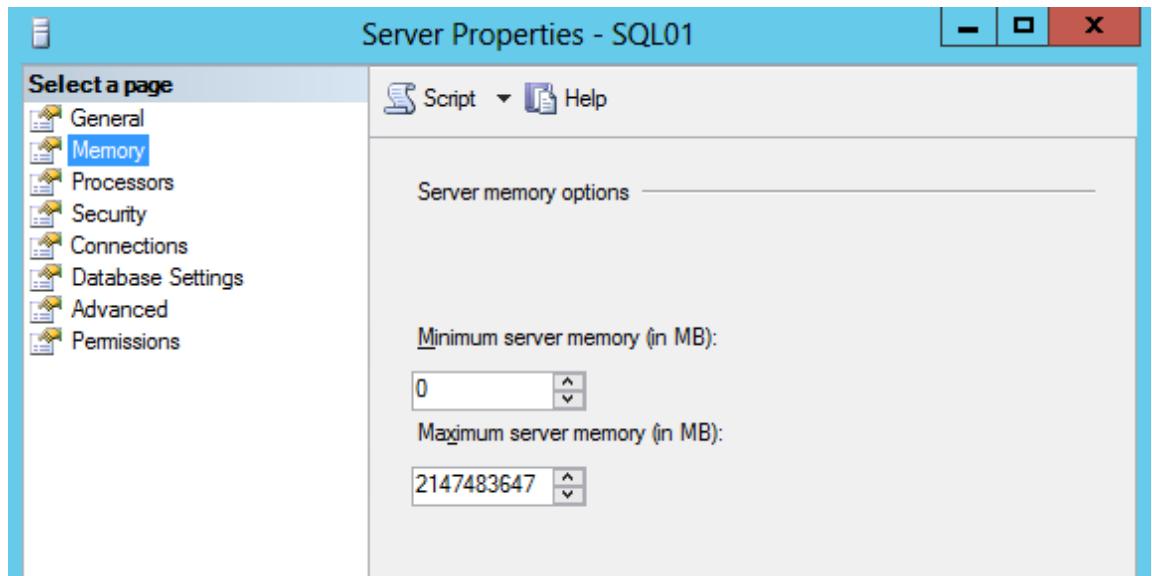
You can set the minimum and maximum memory values in megabytes for a server instance using the following options.



Server Memory Options

| <b>Memory Option</b>   | <b>Description</b>                                                                                                        |
|------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Min Server Memory (MB) | Specifies the minimum memory in megabytes allocated for a SQL Server instance. The default value is set to 0.             |
| Max Server Memory (MB) | Specifies the maximum memory in megabytes allocated for a SQL Server instance. The default value is set to 2147483647 MB. |

You can change these values either using the `sp_configure` procedure or from the **Server Properties**→**Memory** page. The SQL Server can dynamically change the memory settings depending on the requirements.



**Figure 3-5:** The Memory page of Server Properties window showing the server memory options.

## Fill Factor



Fill Factor (3 Slides)

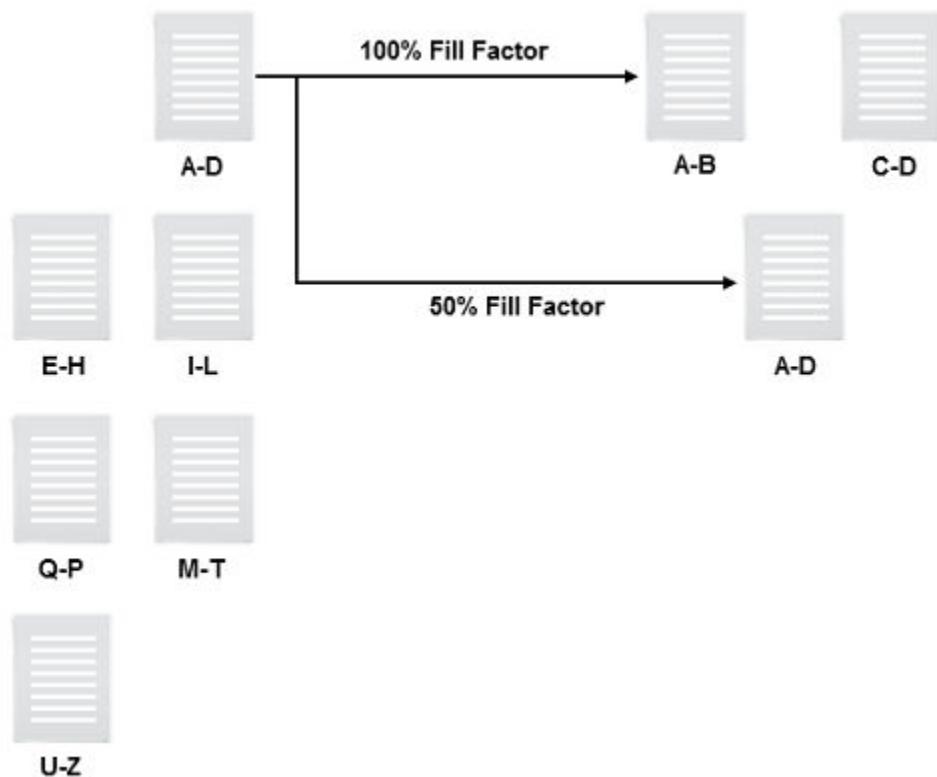
The *fill factor* setting determines the percentage of space filled with data in an index page. Setting fill factor enables you to optimize the storage of indexes while working with data such as a key column with a VARCHAR data type which changes in a non-contiguous manner.

When you add a new row in a table, the index is normally added at the end of the index or the index page. When the fill factor is 100%, a page split occurs to accommodate the new index value. When the fill factor is set below 100, for instance if the fill factor is 70%, the remaining 30% is reserved for expansion and can be used if an index has to be added in that particular page.



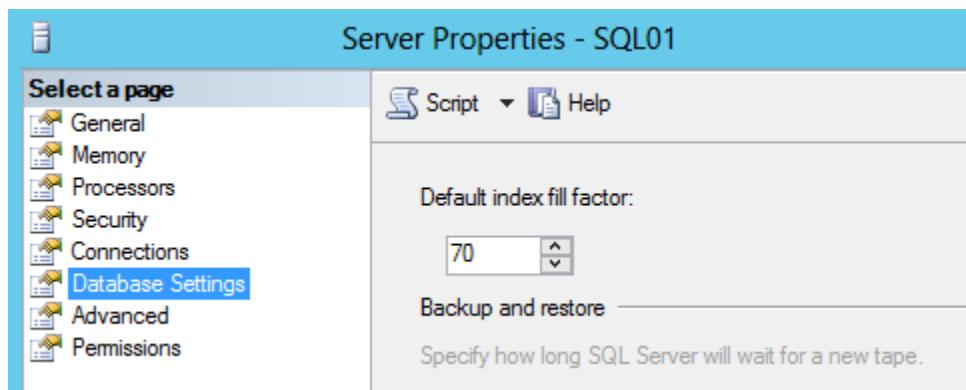
**Note:** A very low fill factor can cause unnecessary page splits, increased storage space, and performance degradation.

Fill factor can be set at two levels—one at the server and the other at index level for every database and table in the server. Fill factor on the server has a default value of 0 and that for the individual indexes is changed based on the requirements.



**Figure 3–6:** Index page A-D is split to A-B and C-D pages when the fill factor is 100%. Fill factor set to 50% avoids this page split.

You can change the server settings in the **Server Properties**→**Database Settings** page.



**Figure 3–7:** The **Database Settings** page of **Server Properties** window showing the fill factor for a server.

The fill factor for individual indexes can be changed in the **Index Properties**→**Options** page.

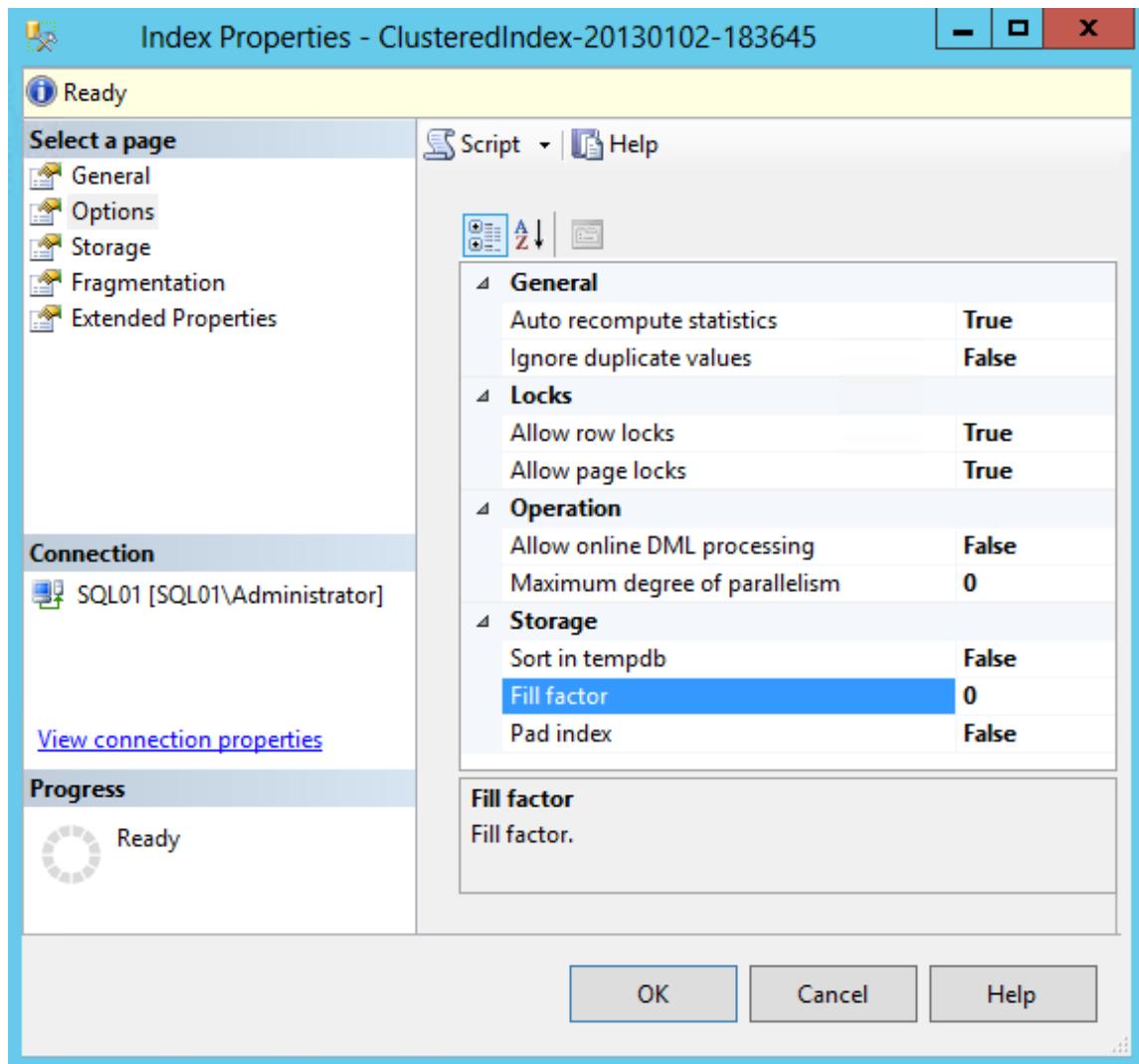


Figure 3-8: The Options page of Index Properties window showing the fill factor setting for an index.



Ask the students how fill factor can increase or decrease the performance in different scenarios.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Configure Memory and CPU

# ACTIVITY 3–2

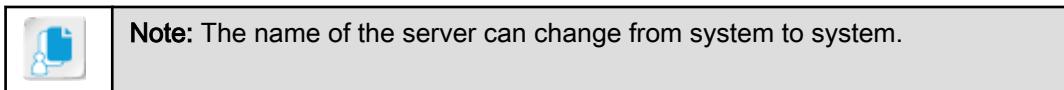
## Modifying the Affinity Mask

### Scenario

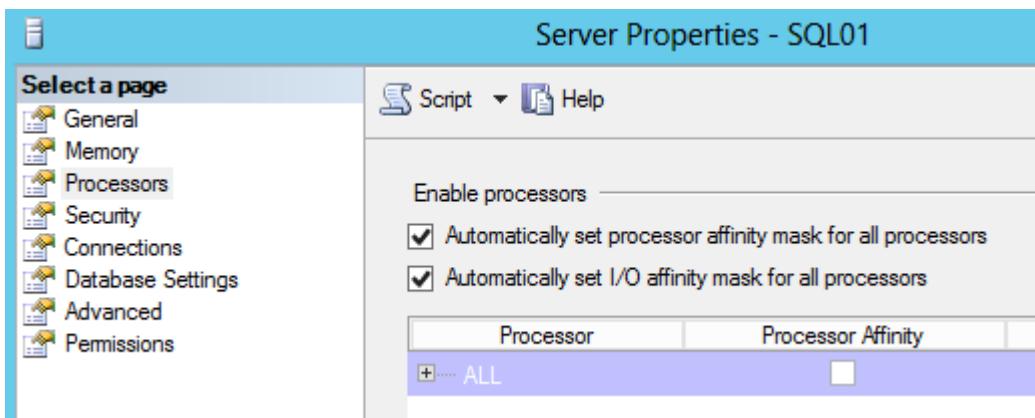
As database administrator of Adventure Works Cycles, you notice that your production server, which handles a lot of sales transactions, might have CPU bottlenecks. To increase the performance, you decide to assign the processes to specific processors by changing the affinity masks.

1. Observe the default processor affinity and I/O affinity settings for the server.

- a) In the **Object Explorer** pane, right-click **SQL01** and select **Properties**.



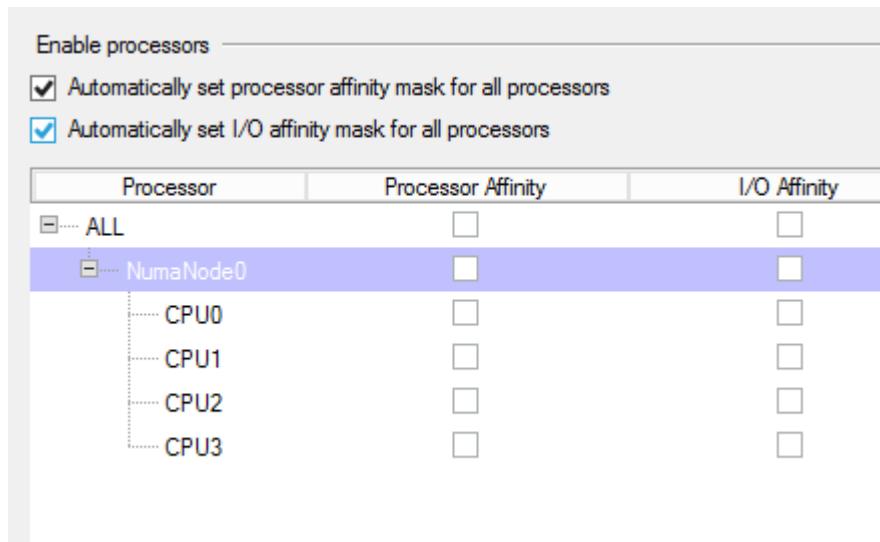
- b) In the **Server Properties** window, in the left pane, select **Processors**.
- c) In the **Processors** page, observe that the **Automatically set processor affinity mask for all processor** and **Automatically set I/O affinity mask for all processors** check boxes are checked.



- d) Below the **Processor** column, expand **ALL→ NumaNoden0** to view all the CPUs.



Remind participants that they need to uncheck that to change the affinity masks.



- e) Close the Server Properties dialog box without making any changes.
2. Execute the code to change the affinity I/O mask to 2.
- Open a **Query Editor** pane, enter the `sp_configure` command to ensure that the show advanced options configuration option is set to 1.  

```
sp_configure 'show advanced options', 1;
```
  - Enter the `RECONFIGURE` command followed by `GO` command to denote the end of the batch of statements.  

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
```
  - Enter the `sp_configure` command followed by the affinity I/O mask configuration option and the value 2.  

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
sp_configure 'affinity I/O mask', 2;
```
  - Enter the `RECONFIGURE` command followed by `GO` command to denote the end of the batch of SQL statements.  

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
sp_configure 'affinity I/O mask', 2;
RECONFIGURE;
GO
```
  - Execute the `sp_configure` statements to set the affinity mask setting.
  - On the **Messages** tab, if your server has more than two CPUs, observe that the message stating that the show advanced options is changed from 1 to 1 and affinity I/O mask is changed from 0 to 2 is displayed.

```

SQLQuery2.sql - SQ...Administrator (52)* X
sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
sp_configure 'affinity I/O mask', 2;
RECONFIGURE;
GO

```

Messages

Configuration option 'show advanced options' changed from 1 to 1.  
Run the RECONFIGURE statement to install.  
Configuration option 'affinity I/O mask' changed from 0 to 2.  
Run the RECONFIGURE statement to install.

- g) Select and execute the RECONFIGURE command to apply the changes.
- h) Open the Server Properties window and on the Processors page, expand ALL→NumaNode0. Observe that the **Automatically set I/O mask for all processors** check box is unchecked and I/O affinity mask is assigned to CPU1.

| Processor | Processor Affinity       | I/O Affinity                        |
|-----------|--------------------------|-------------------------------------|
| ALL       | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| NumaNode0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| CPU0      | <input type="checkbox"/> | <input type="checkbox"/>            |
| CPU1      | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| CPU2      | <input type="checkbox"/> | <input type="checkbox"/>            |
| CPU3      | <input type="checkbox"/> | <input type="checkbox"/>            |

- i) Close the Server Properties window without making any changes.
3. Execute the code to change the processor affinity mask to 0 using the ALTER SERVER CONFIGURATION statement.
- a) Open a **Query Editor** pane, enter an ALTER SERVER CONFIGURATION statement.
- ```
ALTER SERVER CONFIGURATION
```
- b) Type SET keyword followed by PROCESS AFFINITY to set process affinity to CPUs.
- ```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY
```
- c) Type CPU followed by equals operator and the value 0 to assign process affinity to CPU0.
- ```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY CPU=0
```
- d) Execute the ALTER SERVER CONFIGURATION statement.
 - e) On the **Messages** tab, observe that the message "Command(s) completed successfully" is displayed.

```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY CPU=0

100 % < |
Messages
Command(s) completed successfully.
```

- f) Open the Server Properties window and on the **Processors** page, expand **ALL→NumaNode0**. Observe that the **Automatically set processor affinity mask for all processors** check box is unchecked and the **affinity mask** is assigned to CPU0.



Remind participants that CPU affinities can also be changed in the Processors page without using the T-SQL statements.

Processor	Processor Affinity	I/O Affinity
ALL	<input type="checkbox"/>	<input type="checkbox"/>
NumaNode0	<input type="checkbox"/>	<input type="checkbox"/>
CPU0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CPU1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CPU2	<input type="checkbox"/>	<input type="checkbox"/>
CPU3	<input type="checkbox"/>	<input type="checkbox"/>

- g) In the Server Properties window, select **OK**.
 h) Close the **Query Editor** panes without saving any changes.

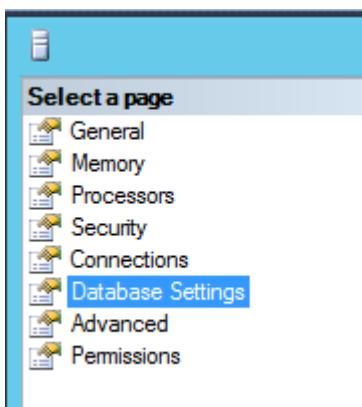
ACTIVITY 3–3

Modifying the Fill Factor

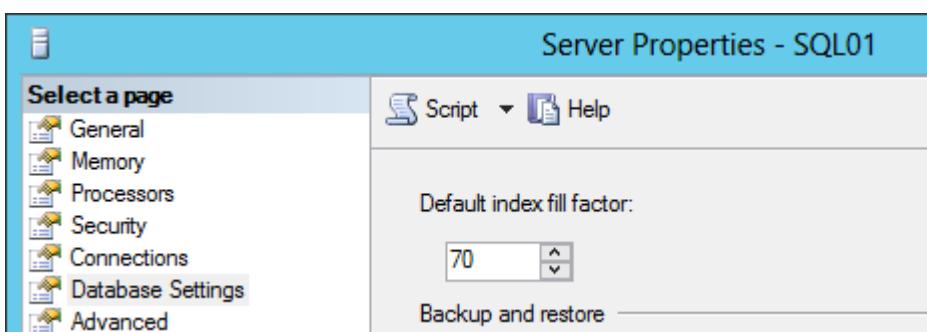
Scenario

As the business of Adventure Works Cycles grows in various countries, you expect more data to be added to the database. The addition of more rows to database tables requires reorganization of index pages as well. This results in page splits to accommodate new rows in the index pages which in turn affect the performance. To ensure minimum page splits during addition of new rows, you decide to allocate more free space to each index on the server by changing the fill factor property.

1. Set the fill factor value to 70%.
 - a) In the **Object Explorer** pane, right-click **SQL01** and select **Properties**.
 - b) In the **Server Properties** window, in the left pane, select **Database Settings**.



- c) In the **Database Settings** page, in the **Default index fill factor** box, type **70** and select **OK**.



2. Verify the fill factor setting using **sp_configure** command.
 - a) Open a **Query Editor** pane, type **sp_configure** command.
 - sp_configure;
 - b) Execute the **sp_configure** command to view the configuration options.
 - c) On the **Results** tab, observe that the **config_value** of the fill factor configuration option is 70. This denotes that only 70% of the index pages is filled and the remaining is reserved for expansion.

22	filestream access level	0	2	0	0
23	fill factor (%)	0	100	70	0
24	ft crawl bandwidth (max)	0	32767	100	100

3. Which statement is true with regard to fill factor?

- Assigns CPU affinity based on indexes.
- Avoids page splits in index pages.
- A fill factor of 50% lowers the performance of the server.
- A fill factor of 5% decreases the storage space.

4. What happens to the index pages when the fill factor server property is set to 70%?

- 70% of the index page is reserved for expansion.
- Page splits occur if the percentage of data in the index page exceeds 30%.
- 70% of the data in index page is moved to a new index page.
- 30% of the index page is reserved for expansion.

TOPIC C

Configure Database Mail

You have configured memory and CPU settings of your server. As a database administrator of an organization, you may have to communicate with your colleagues through mails, alerts and reports, which may contain tables from the database. In such situations, it would be very helpful if your database application is capable of sending emails and reports to users. SQL Server contains Database Mail which enables you to send emails from your database application using the SQL Server. In this topic, you will configure database mail.

Database Mail

SQL Server *Database Mail* feature allows you to send email messages from SQL Server Database Engine. The Database Mail uses *Simple Mail Transfer Protocol (SMTP)* to send email messages. The data in the mail messages are stored in the msdb system database. The sending of email messages is separated from the SQL Server using a process called DatabaseMail.exe. This process offers reliability and scalability. SQL Server can queue huge number of mail messages in case of delivery failures and send them once the SMTP server is ready.

SMTP

SMTP is an Internet standard that enables you to send and receive email messages through Internet. It uses the Transmission Control Protocol (TCP) port 25 for sending and receiving email messages.



Note: For more information about database mail, check out the LearnTO **Communicate Using Database Mail** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.

Mail Accounts and Profiles

When you configure the database mail feature, you need to create or configure the following components:

- Mail account consists of metadata required for a mail to be sent. It includes the server's SMTP details such as Domain name or Internet Protocol (IP) address, SMTP port, and SMTP Authentication scheme with authentication credentials if present.

The SMTP authentication scheme can be of three types:

Authentication	Description
Anonymous	No authentication credentials are required.
Basic	SMTP server authentication credentials are used.
Windows	SQL Server Database Engine Windows account is used for authentication.

- *Mail Profile* is a set of mail accounts with an order that the Database Mail follows to send email messages. Each mail account has a sequence number that determines the order. You can create as many mail accounts and mail profiles. Profiles can also be associated with a set of users or server roles to ensure that only the associated users will be able to use those profiles for sending email messages from SQL Server. This enables you to prioritize and segregate different categories of users easily.



You may want to show **LearnTO Communicate Using Database Mail** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



Mail Accounts and Profiles

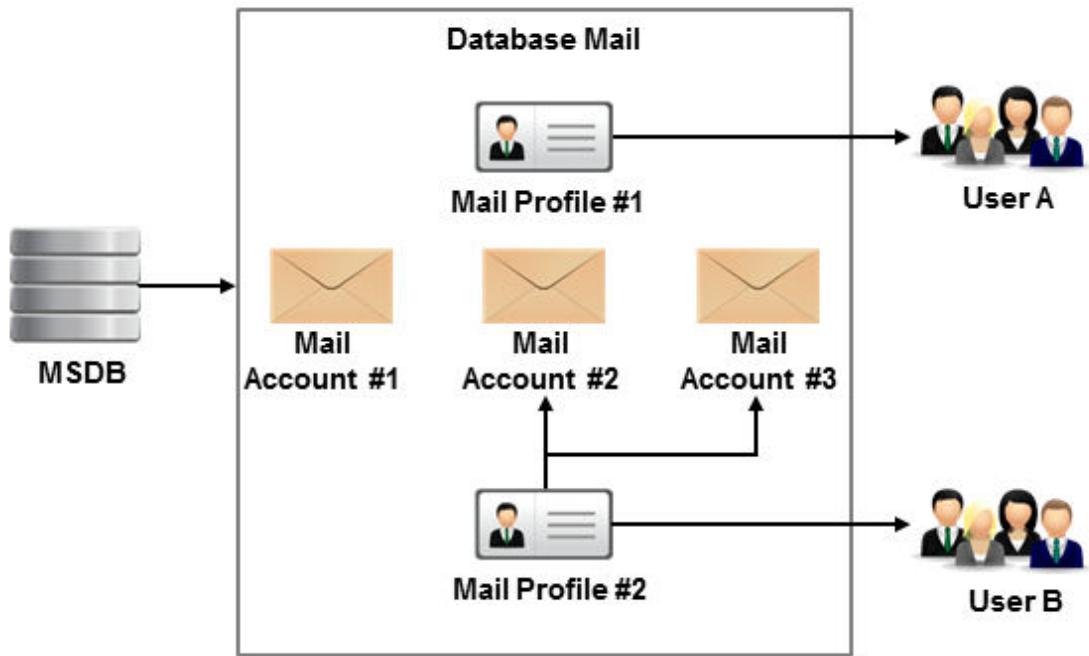


Figure 3–9: Two mail profiles assigned to users A and B.

Database Mail Configuration

Database Mail can be configured using the following ways:

- By configuring the Database Mail Configuration Wizard, which enables you to create a profile and an account for the profile the first time you run it. You can also re-run the wizard to add more accounts or profiles and even change the existing ones.
- By configuring the Database Mail XPs option to 1 using sp_configure, which enables the Database Mail extended stored procedures such as sp_send_dbmail. These stored procedures enable you to perform email operations.



Ask the students a few scenarios where they would use Database Mail extensively in database administration.

The sp_send_dbmail Stored Procedure

The sp_send_dbmail is a system stored procedure that allows you to send an email from a query.

The general syntax of sp_send_dbmail is as follows.

```
sp_send_dbmail
@profile_name = <profile>,
@recipients = <recipients>,
@subject = <subject>,
@body = <body>;
@query=<query>;
```

The sp_send_dbmail stored procedure sends email from the database profile specified in the @profile_name parameter to the recipients specified in the @recipients parameter. The @subject parameter contains the mail subject and the @body parameter contains the mail content. You can also send the result of a query as mail content by adding the @query parameter followed by the query.

Example

The SQL statement to send the result of a query to a email id is as follows.

```
EXEC msdb.dbo.sp_send_dbmail
    @profile_name='Profile 1',
    @recipients='<distributor@example.com>',
    @subject='Query email',
    @body='List of all products in our database:',
    @query='SELECT * FROM Product;';
```

This statement sends an email to ‘distributors@example.com’ from ‘Profile 1’ with subject as ‘Query email’. The email starts with a body ‘List of all products in our database:’ followed by all the rows in the Product table.



Ask the students how they would secure access to different mail profiles in their organization.

Notifications

Notifications are alerts based on some system events to database operators. You need to configure the Database Mail settings for the instance to set up notifications when the system events occur. Once the notification is set up, it will use the Database Mail settings to send the alert.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Configure Database Mail

ACTIVITY 3–4

Configuring Database Mail

Scenario

The management of Adventure Works Cycles wants to send the estimate of orders to their customers through email. You are asked to configure Database Mail on the server and send a test mail to the customer.



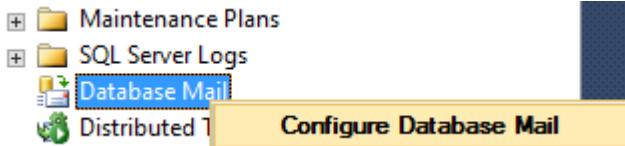
In this activity, students will configure Database Mail to use a specific email account. Be prepared to provide students with this email account along with the relevant SMTP server address for the email account.

1. Configure database mail to send email from SSMS.
- a) In the **Object Explorer** pane, expand **SQL01→Management**.



Note: The server name can vary from system to system.

- b) Right-click **Database Mail** and select **Configure Database Mail**.



- c) In the **Database Mail Configuration Wizard**, on the introduction page, select **Next**.



Welcome to Database Mail Configuration Wizard

Database Mail is a SQL Server component that uses the Simple Mail Transfer Protocol (SMTP) to send e-mail messages. The wizard helps you perform setup and maintenance tasks such as:

- Managing Database Mail accounts and profiles
- Managing Database Mail security
- Configuring Database Mail system parameters

Click **Next** to continue, or **Cancel** to exit the wizard.

- d) On the **Select Configuration Task** page, ensure that the **Set up Database Mail by performing the following tasks:** option is selected and select **Next**.

Select Configuration Task

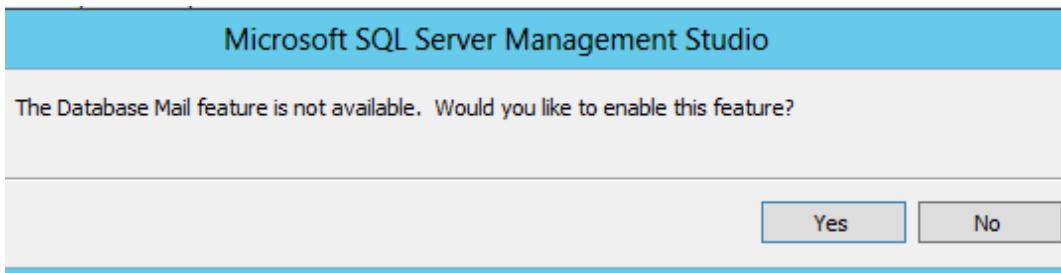
Select setup or maintenance tasks.

If you are installing Database Mail for the first time, select the setup option.

- Set up Database Mail by performing the following tasks:

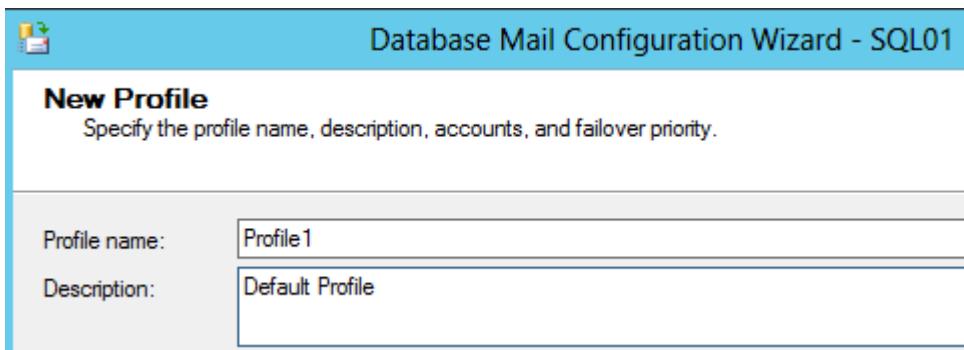
1. Create a new e-mail profile and specify its SMTP accounts
2. Specify profile security
3. Configure system parameters

- e) If prompted to enable the Database Mail feature in SSMS, select **Yes**.



- f) On the **New Profile** page, in the **Profile name** text box, type **Profile1**.

- g) In the **Description** text box, type **Default Profile**.



- h) Select **Add** to add a new account.

- i) In the **New Database Mail Account** dialog box, enter the following details and select **OK**.

- Account Name: <Hotmail>
- Description: To send email from hotmail accounts.
- Email Address: <your hotmail address>
- Display Name: SQL Email
- Server Name: smtp.live.com
- This server requires a secure connection(SSL): On
- Authentication: Basic Authentication
- Username: <your hotmail address>
- Passwords: <your hotmail password>

	Note: You need to provide a valid email address in the Email Address and Username text boxes and the corresponding password in the Password and Confirm password text boxes. Your instructor will provide you with this information.
	Note: A hotmail account is used here. If you want to use other mail accounts, provide the appropriate server name in the Server Name text box. The default port number is 25. If port 25 is blocked in your network, use port 587.

Account name:	Hotmail
Description:	To send email from hotmail accounts
Outgoing mail server (SMTP) —	
E-mail address:	jane.clayton30@hotmail.com
Display name:	SQL Email
Reply e-mail:	
Server name:	smtp.live.com
<input checked="" type="checkbox"/> This server requires a secure connection (SSL)	
SMTP Authentication —	
<input type="radio"/> Windows Authentication using Database Engine service credentials	
<input checked="" type="radio"/> Basic authentication	
User name:	jane.clayton30@hotmail.com
Password:	*****
Confirm password:	*****
<input type="radio"/> Anonymous authentication	

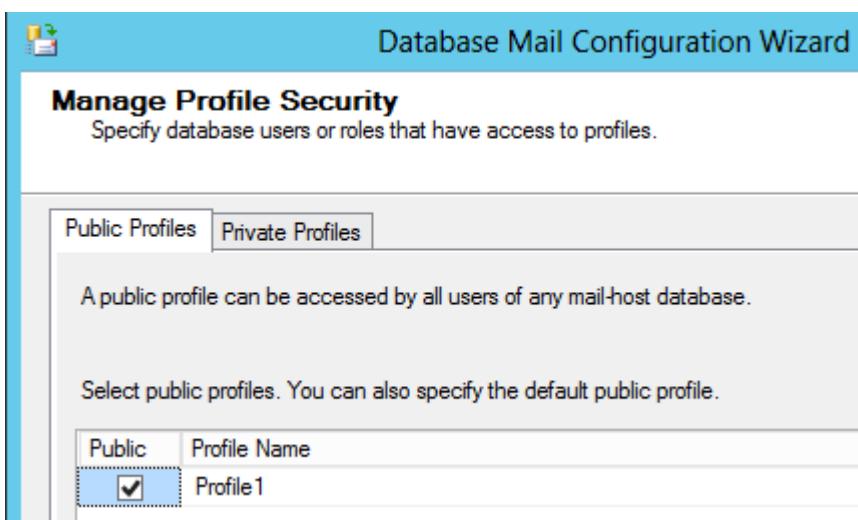
- j) On the **New Profile** page, observe that the mail account is added to the **SMTP accounts** section and select **Next**.



Remind participants that they can add more accounts in the Profile screen .

Profile name:	Profile1	
Description:	Default Profile	
A profile may be associated with multiple SMTP accounts. If an account in the priority list. Specify the accounts associated with th		
SMTP accounts:		
Priority	Account Name	E-mail Address
1	Hotmail	jane.clayton30@gmail.com

- k) On the **Manage Profile Security** page, check the check box in the **Public** column for **Profile1** and select **Next**.



- I) On the **Configure System Parameters** page, observe the default values for system parameters and select **Next**.

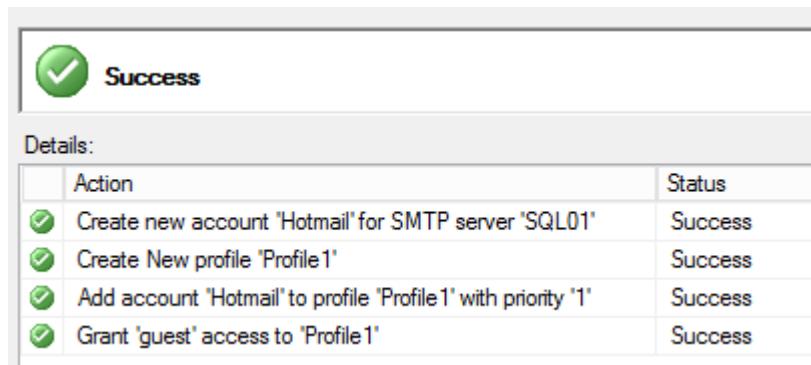
Parameter	Value
Account Retry Attempts	1
Account Retry Delay (seconds)	60
Maximum File Size (Bytes)	1000000
Prohibited Attachment File Extensions	exe,dll,vbs,js
Database Mail Executable Minimum Lifetime (seconds)	600
Logging Level	Extended

- m) On the **Complete the Wizard** page, observe the settings made by you and select **Finish**.

Click Finish to perform these actions:

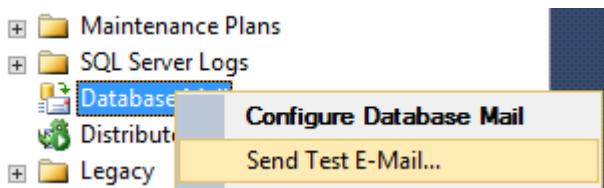
- New Accounts**
 - Create new account 'Hotmail' for SMTP server 'smtp.live.com'
- New Profiles**
 - Create New profile 'Profile1'
- Adding Accounts Profiles**
 - Add account 'Hotmail' to profile 'Profile1' with priority '1'
- Manage Profile Security**
 - Set 'Profile1' as public profile

- n) Wait for the configuration process to complete and close the Database Mail Configuration Wizard.



2. Send test Email to an email id from SSMS.

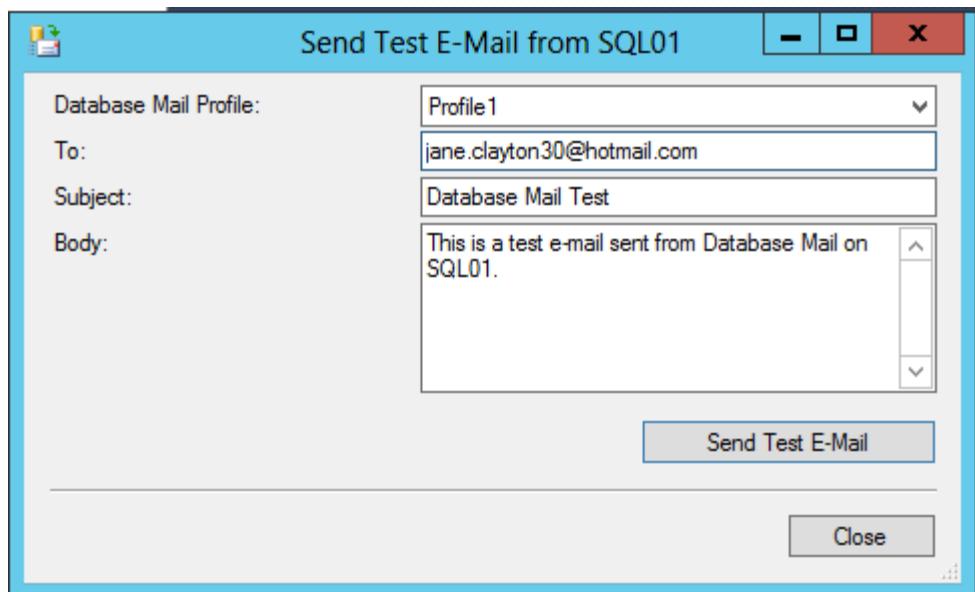
- a) In SSMS, in the **Object Explorer** pane, right-click **Database Mail** and select **Send Test E-Mail**.



- b) In the **Send Test E-Mail** dialog box, in the **To** text box, type the email id of the recipient and select **Send Test E-Mail**.



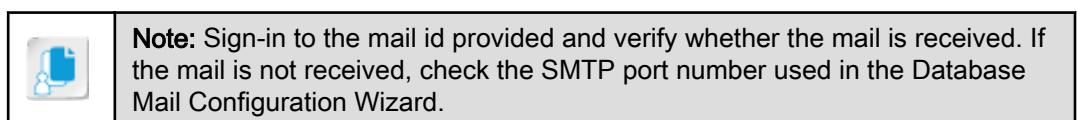
Remind participants that they need to provide their email id in the **To** text box.

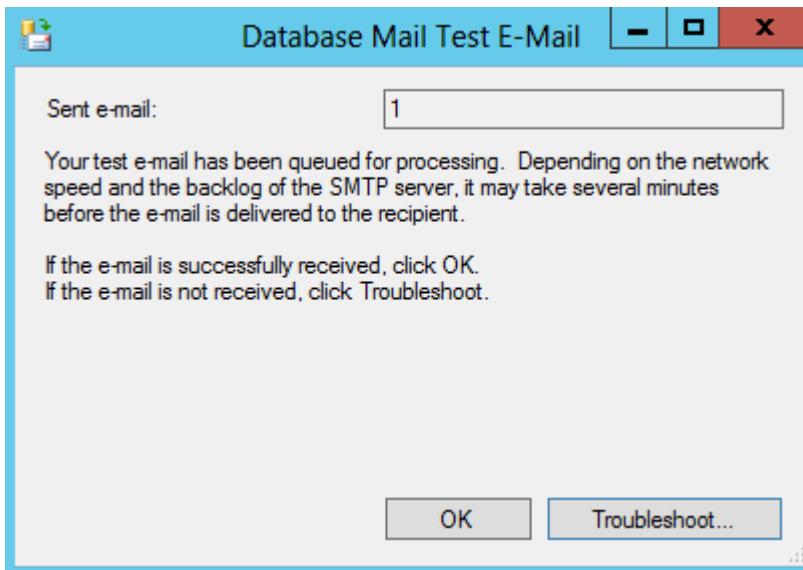


- c) In the **Database Mail Test E-Mail** dialog box, select **OK** if the test email is received by the recipients id.



Select one or two participants, and ask them to summarize the Database Mail feature they configured.





3. Execute the code to display the email ids, error logs, and sent items stored in the msdb database.

- a) Open a **Query Editor** pane, enter a USE command to use the msdb database followed by a GO keyword.

```
USE [msdb]
GO
```

- b) Enter a SELECT statement to display the rows of sysmail_mailitems table followed by a GO keyword.

```
USE [msdb]
GO
SELECT * FROM sysmail_mailitems
GO
```

- c) Enter a SELECT statement to display the rows of sysmail_log table followed by a GO keyword.

```
USE [msdb]
GO
SELECT * FROM sysmail_mailitems
GO
SELECT * FROM sysmail_log
GO
```

- d) Enter a SELECT statement to display the rows of sysmail_sentitems table followed by a GO keyword.

```
"...
SELECT * FROM sysmail_log
GO
SELECT * FROM sysmail_sentitems
GO
```

- e) On the **SQL Editor** toolbar, select **Execute** to execute the USE command and install the changed settings.

- f) On the **Results** tab, observe that the emails in the system, error logs with the detailed error message and the sent items are displayed.

The screenshot shows the SSMS interface with a query window titled "SQLQuery1.sql - SQ...Administrator (53)*". The window contains the following T-SQL code:

```
USE [msdb]
GO
SELECT * FROM sysmail_mailitems
GO
SELECT * FROM sysmail_log
GO
SELECT * FROM sysmail_sentitems
GO
```

Below the code, there are three tabs: "Results", "Messages", and another tab that is partially visible. The "Results" tab displays the output of the first query, which is a single row from the sysmail_mailitems table:

mailitem_id	profile_id	recipients	copy_recipient
24	24	3	jane.clayton30@hotmail.c... NULL

The "Messages" tab displays the output of the second query, which is a single row from the sysmail_log table:

log_id	event_type	log_date	description
.....

The third tab displays the output of the third query, which is a single row from the sysmail_sentitems table:

mailitem_id	profile_id	recipients	copy_recipients
.....

- g) Close the **Query Editor** panes without saving the queries.

ACTIVITY 3–5

Sending Email Using Database Mail

Before You Begin

Attach the AdventureWorks2012 database to the server.

1. In the **Object Explorer** pane, in the **SQL01** instance, right-click the **Databases** folder and select **Attach**.
2. In the **Attach Databases** dialog box, select **Add**.
3. In the **Locate Database Files** dialog box, navigate to the **C:\093108Data\AdventureWorks2012_Database** folder and select the **AdventureWorks2012_Data.mdf** file.
4. In the **Locate Database Files** dialog box, select **OK**.
5. In the **Attach Databases** dialog box, select **OK**.
6. In the **Object Explorer** pane, expand the **Databases** folder and right-click the **AdventureWorks2012** database and select **Properties**.
7. In the **Database Properties-AdventureWorks2012** dialog box, in the **Select a page** section, select **Options**.
8. In the right pane, from the **Recovery Model** drop-down list, select **Full** and select **OK**.

Scenario

The production team of Adventure Works Cycles is planning to send a consolidated list of the required components that are queued up in the production line-up to a distributor. You are asked to list the products that are in production and send the result through mail to the distributor.

1. Execute the SELECT statement to display all the products that are in production.
 - a) Open a **Query Editor** pane, and then enter a SELECT statement to display the rows of **AdventureWorks2012.Production.Product** table.

```
SELECT * FROM AdventureWorks2012.Production.Product;
```

 - b) On the **SQL Editor** toolbar, select **Execute** to execute the SELECT statement.
 - c) On the **Results** tab, observe that the details of the products that are in production are displayed.

The screenshot shows the SSMS interface with a query window titled "SQLQuery5.sql - SQ...Administrator (53)*". The query entered is:

```
SELECT * FROM AdventureWorks2012.Production.Product;
```

The results tab displays the following table of product data:

	ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag
1	1	Adjustable Race	AR-5381	0	0
2	2	Bearing Ball	BA-8327	0	0
3	3	BB Ball Bearing	BE-2349	1	0
4	4	Headset Ball Bearings	BE-2908	0	0
5	316	Blade	BL-2036	1	0

2. Execute the code to email the result of the SELECT statement to an email id.
- Open a **Query Editor** pane, enter an EXEC statement to execute the msdb.dbo.sp_send_dbmail stored procedure.
- ```
EXEC msdb.dbo.sp_send_dbmail
```
- Enter the parameters for the sp\_send\_dbmail stored procedure with the following details.
    - @profile\_name='Profile1',
    - @recipients='<distributor's email address>',
    - @subject='Query email',
    - @body='List of all products in the production stage:',
    - @query='SELECT ProductID, Name, ProductNumber, MakeFlag, FinishedGoodsFlag FROM AdventureWorks2012.Production.Product';'



Remind participants to enter a valid email address in the @recipients field.



**Note:** Type a valid email id in the @recipients parameter.

```
EXEC msdb.dbo.sp_send_dbmail
@profile_name='Profile1',
@recipients='<distributors email address>',
@subject='Query email',
@body='List of all products in the production stage:',
@query='SELECT ProductID, Name, ProductNumber,
MakeFlag, FinishedGoodsFlag
FROM AdventureWorks2012.Production.Product;';
```

- On the **SQL Editor** toolbar, select **Execute** to execute the system stored procedure.
- On the **Messages** tab, observe that the number of mails queued is displayed.
- Log in to the email id specified in the @recipients field and observe that an email containing the result of the query is received.



Verify that all participants have received the mail with the product list before you proceed.

List of all products in the production stage:

| ProductID | Name                  | ProductNumber | MakeFlag | FinishedGoodsFlag |
|-----------|-----------------------|---------------|----------|-------------------|
| 1         | Adjustable Race       | AR-5381       | 0        | 0                 |
| 2         | Bearing Ball          | BA-8327       | 0        | 0                 |
| 3         | BB Ball Bearing       | BE-2349       | 1        | 0                 |
| 4         | Headset Ball Bearings | BE-2908       | 0        | 0                 |
| 316       | Blade                 | BL-2036       | 1        | 0                 |
| 317       | LL Crankarm           | CA-5965       | 0        | 0                 |
| 318       | ML Crankarm           | CA-6738       | 0        | 0                 |

- Close the **Query Editor** pane without saving the queries.

# Summary

In this lesson, you configured the server properties, memory settings, CPU settings, and database mail messages. This will enable you to configure the server properties depending on your requirements and improve the performance, scalability and reliability of your database system.

## In what ways do you think is using sp\_configure advantageous over the GUI of SSMS?

**A:** Answers will vary, but may include: using the sp\_configure system stored procedure to configure all server configuration options, the queries executed can be saved for later use, all server configuration options with the values can be verified using sp\_configure stored procedure.



Use the review questions provided to generate discussion among the participants.

## In what circumstances would you use the affinity mask option?

**A:** Answers will vary, but may include: heavy load, when more than four processors are used, when performance degrades due to frequent reloading of processor cache.



**Note:** Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.



# 4

# Managing Databases in SQL Server 2012

**Lesson Time:** 1 hour, 15 minutes

## Lesson Objectives

In this lesson, you will:

- Create a contained database.
- Configure database options.

## Lesson Introduction

You configured SQL Server® by installing and migrating the required settings to SQL Server 2012. When working with organizational data, you need to ensure that the critical information is easily accessible. You can manage data accessibility by managing the database server instance using the various tools such as auto shrink, auto close, and recovery models available in SQL Server 2012. Familiarizing yourself with database management techniques will enable you to efficiently administer the databases. In this lesson, you will manage databases in SQL Server 2012.

# TOPIC A

## Create a Contained Database

Databases are the core concepts of SQL Server, it is important to understand the key aspects of a database such as physical structure of databases, database types, and user databases. A better understanding on these database infrastructure related concepts will enable you to create a contained database that stores the required data efficiently. In this topic, you will create a contained database.

### Physical Structure of Databases

The physical structure of databases refers to the location of the physical files that make up the database itself. Normally, a database defined in SQL Server is made up of a log file (.ldf) and a database file (.mdf). Both these files are stored in the same location.

However, if you design your infrastructure for performance, you might want to place both these on different locations preferably on different disks. This helps in increasing performance as the system can perform concurrent input or output (I/O) operations on both files.

To enhance the performance, you can split the database into multiple files and file groups located on different physical drives. This sort of physical structure of databases helps in maintaining a high throughput of I/O operations in the database.



**Note:** Placing the files on different logical drives does not yield performance gains.

### System Databases

The *system databases* store the metadata information of the server, database, and database objects. System databases are internally used by the server for system functions.

The system databases used in SQL Server 2012 are listed in the table below.

| <b>System Database</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                     |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Master database        | The <i>master database</i> is the system database created for each instance of SQL Server. The master database contains the settings for the server components such as logins, linked servers, encryption, and database compression. It also contains system-stored procedures and functions that you can include in your metadata queries.                            |
| Model database         | The <i>model database</i> is used as the template for any new user database that is created. Any change you make in the model database will be reflected in any new database you create from then on. You can use the model database of each instance to make settings for all databases on that instance.                                                             |
| Tempdb database        | The <i>tempdb database</i> is a temporary database and is recreated every time the SQL Server process for the instance starts. The tempdb database is used whenever the SQL Server requires to create some temporary entities due to the result of a query or transaction. You should never create object within this database unless it is meant for temporary usage. |
| Msdb database          | The <i>msdb database</i> is used to store configuration information about database components such as database mail, SQL Server agent, and jobs. It also contains configuration information about stored                                                                                                                                                               |

| <b>System Database</b> | <b>Description</b>                                                          |
|------------------------|-----------------------------------------------------------------------------|
| Resource database      | procedures for working with these database components using SQL statements. |

## User Databases

*User databases* are the databases that are created by users or the administrators. These are the standard databases that are used for business and transactional purposes. Various database objects such as tables, relationships, keys, procedures, triggers, views, and logins can be created within the user database.



Ask the students to state the circumstances during which they perform database operations on the master database.



User Databases

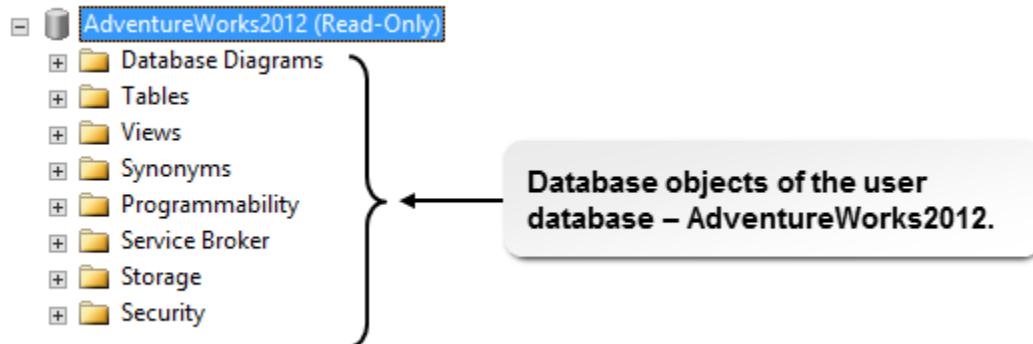


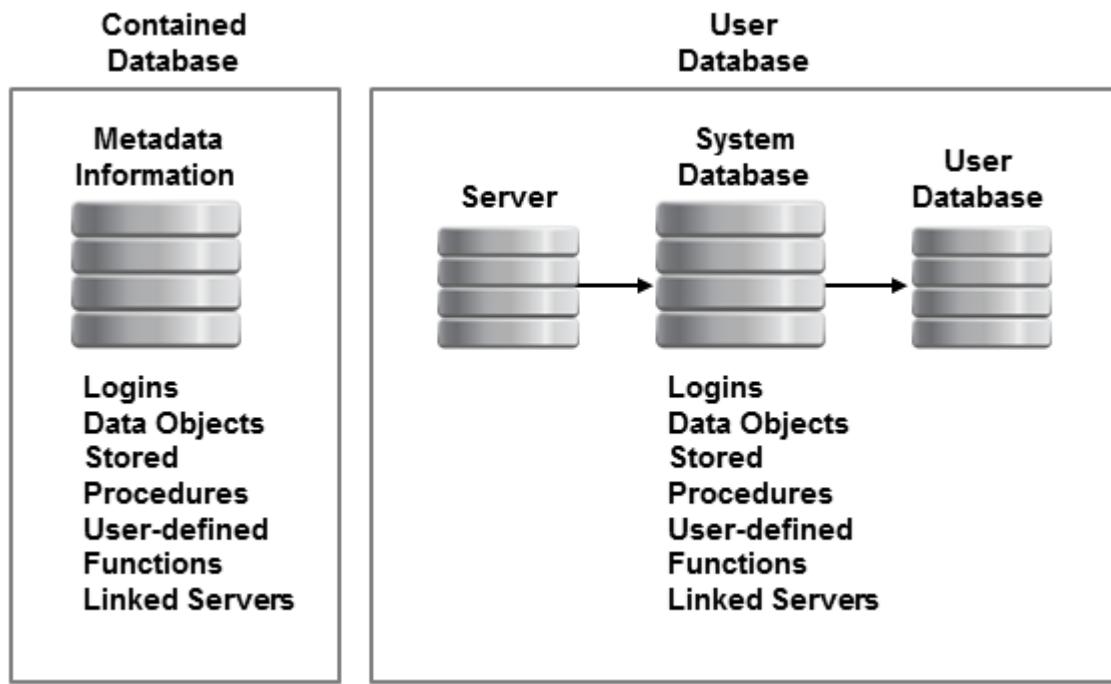
Figure 4-1: User databases are typically created by the administrator.

## Contained Databases

*Contained databases* are a new type of database in SQL Server 2012. It is a special type of user database. The contained database stores most of the metadata information within the database itself thus ensuring that it is self-contained. In case of a user database, the metadata information's are stored within the system databases, especially the master database.



Contained Databases



*Figure 4-2: Structure of contained database and user database.*

SQL Server 2012 also supports partially contained databases where many of the objects that normally reside in system databases can be moved within the contained database itself, while some objects may still reside in the system databases.

In SQL Server 2012, any object that is within the database itself is termed as contained while an uncontained database has objects that do not exist within the database boundary. All versions of SQL Server prior to 2012 have only uncontained databases. In SQL Server 2012, new user databases default to the uncontained state by having their containment property set to **None**.

To allow databases to be contained, database administrators need to enable database containment at the server instance level by setting the contained database authentication property to 1 using the `sp_configure` stored procedure.

The syntax to enable the database containment at the server instance level is as follows.

```
sp_configure 'contained database authentication', '1';
RECONFIGURE;
```

## Example

The `sp_configure` statement to enable the database containment at the server instance level is as follows.

```
USE master;
GO
EXEC sp_configure 'contained database authentication', '1';
```

You can also change the database containment level for the database the `ALTER DATABASE` statement.

The syntax for changing the database containment level is as follows.

```
ALTER DATABASE <Database Name> SET CONTAINMENT = PARTIAL;
```

An `ALTER DATABASE` statement to change the database containment level of `MASTER` database is as follows.

```
ALTER DATABASE MASTER SET CONTAINMENT = PARTIAL;
```

## Advantage and Disadvantage of Contained Databases

The advantage of using a contained database is that it is completely isolated within the instance itself that will enable you to easily move the database to a different instance or server since all the required metadata also resides within the database. In addition, you can allow database users who have full rights on the database to work with the database without providing them the sysadmin rights on the instance. The disadvantage of contained databases is that it cannot be part of advanced scenarios such as database replication or data tracking.



Ask the students to specify the instances under which they will use contained databases.

## Contained Login

A *contained login* is a login that exists only within the contained database without having any reference in the SQL Server instance. SQL Server allows you to create user login that are specific only to that database and do not exist outside of the database context. A contained login user can connect to a database by specifying the required login credentials in the connection string.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Create a Contained Database

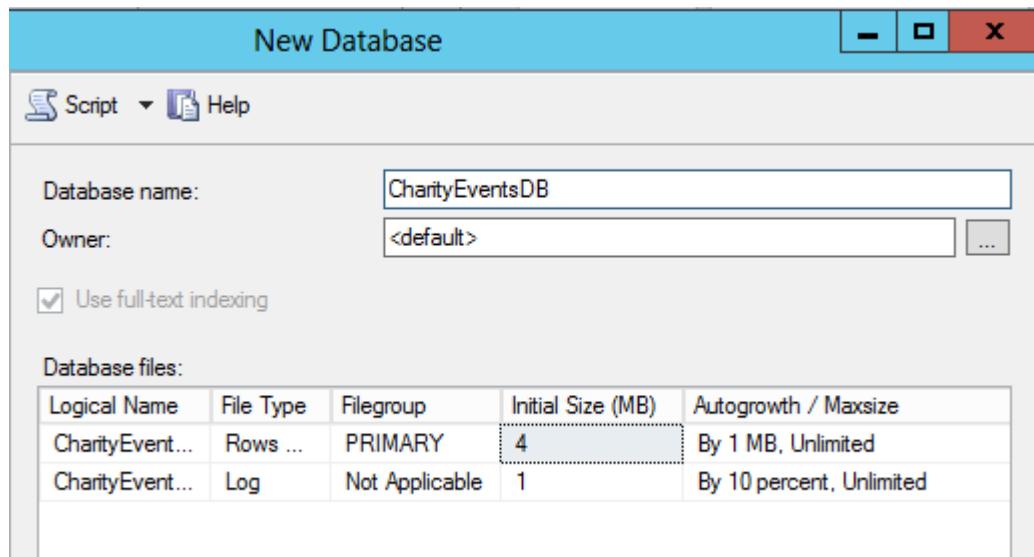
# ACTIVITY 4-1

## Creating a Contained Database

### Scenario

Management of Adventure Works Cycles had planned to create a new department to fulfill its corporate social responsibilities. You are asked to create a database named **CharityEventsDB** to store the details of the charity events that are conducted by the organization. The database should be accessed only by specific users who are part of the department handling corporate social responsibilities. Management also wants to ensure that the **CharityEventsDB** is isolated from other databases in their network. You decide to create the **CharityEventsDB** database and migrate it as a contained database.

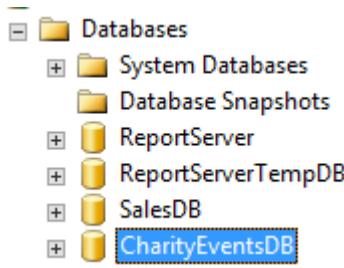
1. Create the “CharityEventsDB” database.
  - a) In the **Object Explorer** pane, right-click **Databases** and from the fly-out pane, select **New Database**.
  - b) In the New Database window, on the **General** page, in the right pane, in the **Database name** text box, type **CharityEventsDB** to enter the database name.
  - c) In the **Database files** section, observe that a data file and a log file are created.



- d) Select **OK** to create the database.
2. In the Object Explorer, verify that the database has been created.
  - a) In the **Object Explorer** pane, expand **Databases**.
  - b) Observe that the **CharityEventsDB** database is listed.



**Note:** If the **CharityEventsDB** database is not listed, right-click the **Databases** node and from the fly-out pane, select **Refresh**.



**3. Enable the Contained Databases option.**

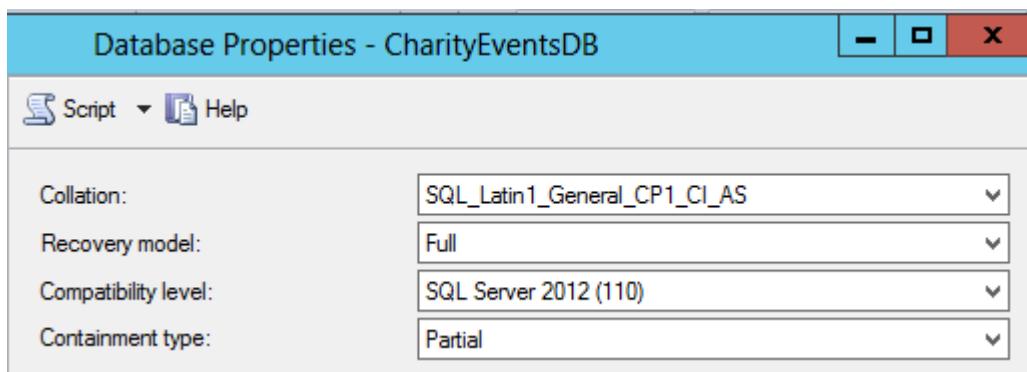
- In the **Object Explorer** pane, right-click the **SQL01** and from the fly-out pane, select **Properties**.
- In the Server Properties window, in the left pane, navigate to the **Advanced** page.
- On the **Advanced** pane, in the right pane, in the **Containment** section, from the **Enabled Contained Databases** drop-down list, select **True**.



- In the Server Properties window, select **OK** to save the settings.

**4. Change the CharityEventsDB database containment type to partial and set up the user login credentials.**

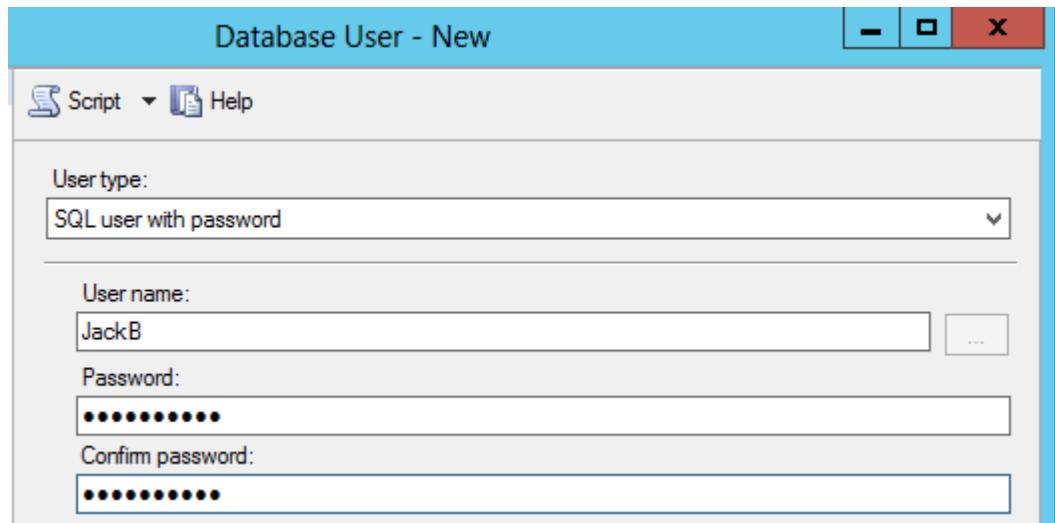
- In the **Object Explorer** pane, right-click **CharityEventsDB** database and from the fly-out pane, select **Properties**.
- In the Database Properties window, in the left pane, navigate to the **Options** page.
- In the right pane, from the **Containment type** drop-down list, select **Partial**.



- In the Database Properties window, select **OK** to save the settings.
- In the **Object Explorer** pane, expand **CharityEventsDB** → **Security**.
- Right-click **Users** and select **New User**.
- In the Database User – New window, from the **User type** drop-down list, ensure that **SQL user with password** is selected.
- In the **User name** text box, type **JackB**.



- i) In the **Password** text box, type *pass@word1*.
- j) In the **Confirm password** text box, type *pass@word1*.



- k) In the Database User – New window, select **OK** to save the contained user settings.
  - l) In the **Object Explorer** pane, right-click **SQL01**→**Disconnect**.
5. Login to the contained database.
- a) In the **Object Explorer** pane, select **Connect**→**Database Engine**.
  - b) In the **Connect to Server** dialog box, from the **Authentication** drop-down list, select **SQL Server Authentication**.
  - c) Enter the **Login** as *JackB* and in the **Password** text box, type *pass@word1*.
  - d) Select the **Options** button.



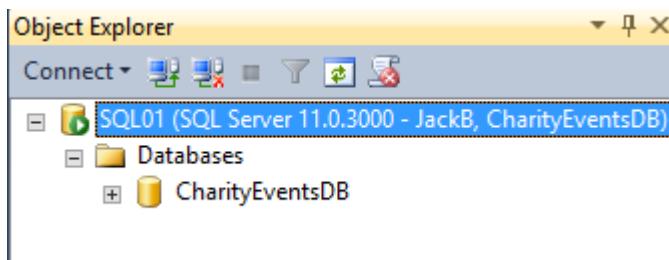
- e) In the **Connect to Server** dialog box, on the **Connection Properties** tab, select the **Connect to database** drop-down list, and type *CharityEventsDB*.



**Note:** Do not attempt to browse or emulate the databases on the server as the JackB user has access only to the CharityEventsDB database.



- f) Select **Connect** to connect the contained user to the contained database.
- g) In the **Object Explorer** pane, expand the **Databases** folder to view the contained database.



# TOPIC B

## Configure Database Options

You have created databases to store your organizational data. You can further optimize the created database by configuring the properties and settings of databases. Familiarizing yourself with database options will enable you to optimize the performance of your database. In this topic, you will configure database options.

### Auto Close

The **Auto Close** option for a database enables SQL Server to release all resources from that database once the last connection has ended. When the last connection is closed, the resources allocated to the database will be completely released back to the system.

When SQL Server starts on a Windows Server, it checks all databases and opens each database for receiving connections by assigning resources to it such as computing memory, network, and storage. When a user tries to connect to the database, these existing resources are used to improve the connection time to enable users to access the database. These resources are shared across multiple connections.

While this might seem to be a good idea, in practice this is not. This is because there is no way of predicting when the next connection might come in, it could be the very next millisecond. In such cases, not only does SQL Server have to complete the resource freeing process, it immediately needs to start the resource allocation process again, both of which take time. The performance of the system during these processes will actually go down.

It is highly recommended that the **Auto Close** option for the database be kept as **False**. Unless the database is one that is used extremely occasionally, most databases should simply let SQL Server allocate the resources and keep them active for any incoming connections.

### Auto Shrink

*Auto Shrink* is an option available for each database in SQL Server that automatically reduces the physical size of the database file(s) if the free space in the file is greater than 25%. That is, if a database file is 100MB and the free space within the file is greater than 25MB, the Auto Shrink option will reduce the file size automatically.

Again, this may be a good option to enable you to save space on the server. However, there are some extremely large overheads for this process.

- By performing the shrink, positions of the data within the file and therefore their references in the indexes will all change requiring a possible rebuild.
- While the shrink might reduce the size, further expansion of the database will require a size increase. A continuous increase-decrease cycle can fragment the physical file(s) on the disk significantly.



Ask the students whether they have encountered situations during which they need to shrink the database.

### Auto Shrink Usage Scenarios

For instance, if a stored procedure creates a few hundred rows in a temporary table for calculation or reporting purposes, SQL Server increases the database size. At the end of the stored procedure execution, it deletes all the temporary rows. The Auto Shrink option will reduce the size of the database. In the meantime, if another user runs the same stored procedure with different parameters, this may result in some records to be created in the temporary table. SQL Server has to again increase the physical size of the database file since Auto Shrink option has reduced the file size earlier.

Increasing and reducing the size of the database file is a time consuming and process intensive job, which can even block other databases using the same physical disk system. The **Auto Shrink** option is recommended to be disabled in the settings for each database. If the database is not going to be modified significantly, database administrators can manually shrink the database.



**Note:** For more information about Auto Close and Auto Shrink options, check out the LearnTO **Configure the Auto Close and Auto Shrink Options for Databases** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.



Ask the students when they should enable the Auto Close and Auto Shrink options.



You may want to show LearnTO **Configure the Auto Close and Auto Shrink Options for Databases** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.

## The Simple Recovery Model

The *Simple recovery model* is the default database recovery model and as the name suggests, is the simplest recovery model. When you use this mode on a database, the transaction log does not store any data past the committing of the data to the disk. In case of database failure, you will need to rely on the last full backup you took of the database. Any data that was added or modified past that backup will be lost.

This is because SQL Server truncates the log as soon as the database has written the changed data to the file on the disk (called a checkpoint). So the log no longer stores any information regarding the transaction. So, if you've taken a backup, say on Jan-01, 2013 at 5:00AM and the database gets corrupted and needs recovery on Jan-03, 2013 at 10:00 AM, you will lose more than two days of work.

## The Full Recovery Model

In the *full recovery model*, the database maintains all transactions in the log till a backup is performed of the log itself. This means that you can recover the database to any point in time before the failure occurred.

The transaction log keeps and maintains all the transactions that occur on the database. When a backup is taken, the database file(s) as well as the log is backed up and the log gets truncated. Any new additions will be stored in the database files as well as the logs.

If a failure occurs, you can recover by first backing up the transaction log and restoring the earlier backups. Once the main restore is complete, simply restore the last transaction log backup and you will get the database back prior to the point at which it had failed.

## The Full Database Recovery Model Process

The full database recovery model process includes the following steps:

1. A full database backup is created at a particular point in time (T0). At this time, the database files and logs are backed up and the log is truncated (emptied) on the server.
2. At a later point in time (T1), a differential backup might be taken. Note that this is optional.
3. At some point (T2), the database files have a failure.
4. The admin immediately (T3) takes a backup of the transaction log which contains the transactions from T1 till just prior to T2.
5. The recovery process is started by using the full and differential backups first.
6. At the end, the recently backed up transaction log is recovered onto the restored database. This redoing all the transaction that occurred from the last backup (at T1) to just before the crash.
7. The database is now fully recovered.



The Full Database Recovery Model Process

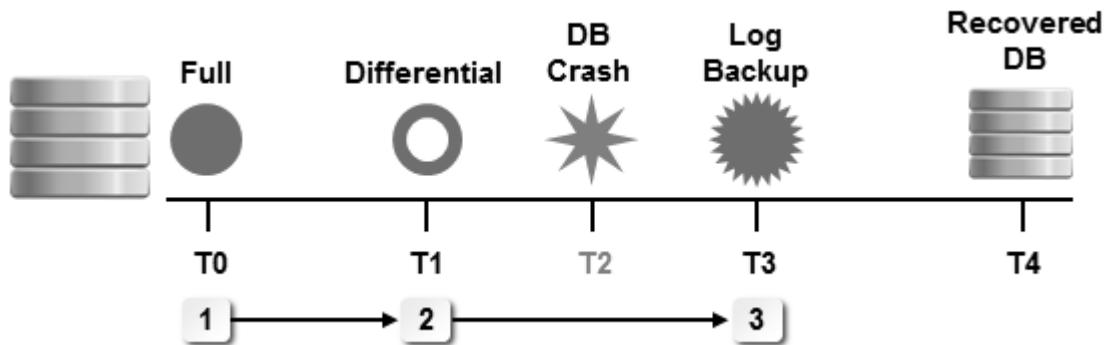


Figure 4-3: The full database recovery model process diagram.

## Bulk Logged Recovery Model

The *bulk logged recovery model* is similar to the full recovery model. In the bulk-logged recovery model, the transaction log uses minimal space for bulk operations such as a bulk insert or index creation or rebuild operations. While you can still perform a point-in-time recovery using a bulk-logged database, if there have been any bulk operations since the time of the last backup, all of those data will be lost.

It is recommended that for reliability and performance reasons, you switch between the full and bulk-logged recovery models. In normal circumstances, run the database in full recovery mode. When a bulk operation is about to happen, switch over to the bulk-logged mode and then once the operation is complete, immediately switch back to the full mode and take a backup. This ensures that all changes not logged due to the bulk logging will be backed up as well.



Ask the students if bulk logging will be sufficient for most database recovery scenarios in their organization.



Methods to Change the Recovery Models

## Methods to Change the Recovery Models

You can change the recovery model of a database using two methods:

- Graphical User Interface (GUI)
- T-SQL statements

In the Microsoft SQL Management Studio at any time, you can change the **Recovery Model** value in the **Database Properties**→**Options** screen in SQL Server Management Studio. Select from the **Full**, **Simple** or **Bulk Logged** options.

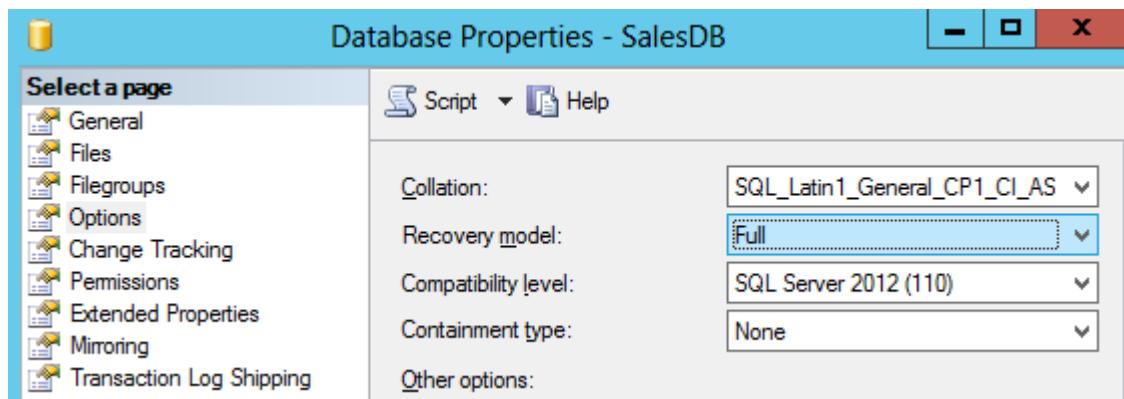


Figure 4-4: The Database Properties window for changing the recovery model.

You can also use T-SQL statements to switch to bulk mode before a bulk operation and then revert back to full recovery mode.

The syntax to switch to bulk mode before a bulk operation and then revert to full recovery mode is as follows.

```
ALTER DATABASE <database name>
SET RECOVERY [SIMPLE | FULL | BULK_LOGGED];
```

## Example

An ALTER DATABASE statement to switch to a bulk logged recovery model for a SalesDB database is as follows.

```
ALTER DATABASE SalesDB
SET RECOVERY BULK_LOGGED;
```

This T-SQL statement modifies the SalesDB database and changes the recovery model to bulk logged.

## File Space

*File space* is the space occupied by a database file on the disk. To decide whether to shrink or grow a database file, you need to identify the file space. To check for the database file space, you need to query the system tables and retrieve information about the size of the files in the database. This information is stored in the dbo.sysfiles table in each database.



## Example

A SELECT statement to display the space occupied by each file in dbo.sysfiles is as follows.

```
SELECT * FROM dbo.sysfiles;
```

This statement will retrieve the details of each file that is part of the database in units in terms of blocks.

**A SELECT statement displaying the space occupied by each file component in the server.**

 A screenshot of the SQL Server Management Studio interface. At the top, there's a title bar for 'SQLQuery1.sql - SQ...Administrator (52)\*'. Below it is a toolbar with icons for New Query, Open, Save, and others. The main area shows a single line of T-SQL code: 'SELECT \* FROM dbo.sysfiles;'. An arrow points from this code down to a results grid. The results grid has two tabs at the top: 'Results' (which is selected) and 'Messages'. The results grid displays the following data:
 

|   | fileid | groupid | size | maxsize   | growth | status  | perf | name        | filename                   |
|---|--------|---------|------|-----------|--------|---------|------|-------------|----------------------------|
| 1 | 1      | 1       | 512  | -1        | 128    | 2       | 0    | SalesDB     | C:\Program Files\Microsoft |
| 2 | 2      | 0       | 128  | 268435456 | 10     | 1048642 | 0    | SalesDB_log | C:\Program Files\Microsoft |
| 3 | 3      | 2       | 512  | -1        | 128    | 2       | 0    | Trans-01    | C:\Drive1\Trans-01.mdf     |
| 4 | 4      | 2       | 512  | -1        | 128    | 2       | 0    | Trans-02    | C:\Drive2\Trans-02.mdf     |
| 5 | 5      | 3       | 512  | -1        | 128    | 2       | 0    | LYTrans     | C:\Drive2\LYTrans.mdf      |

Figure 4-5: dbo.sysfiles statement displaying the available records and their respective file space.

The **Size** column displays the size on the disk for each file in the blocks.

## The FILEPROPERTY Function



The FILEPROPERTY Function (2 Slides)



**Note:** The SpaceUsed property displays the space occupied by the physical database files.

### Example

A SELECT statement with SpaceUsed Property is as follows.

```
SELECT FILEPROPERTY('SalesDB', 'SpaceUsed') AS SpaceUsed;
```

This statement will return the actual space used in the specified file by fetching the details from the name column.

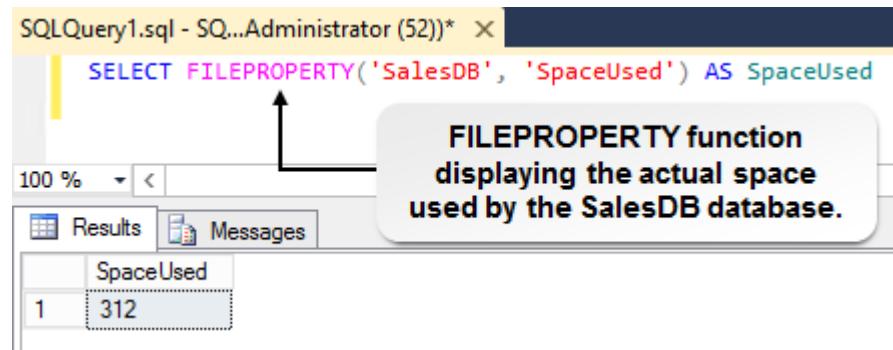


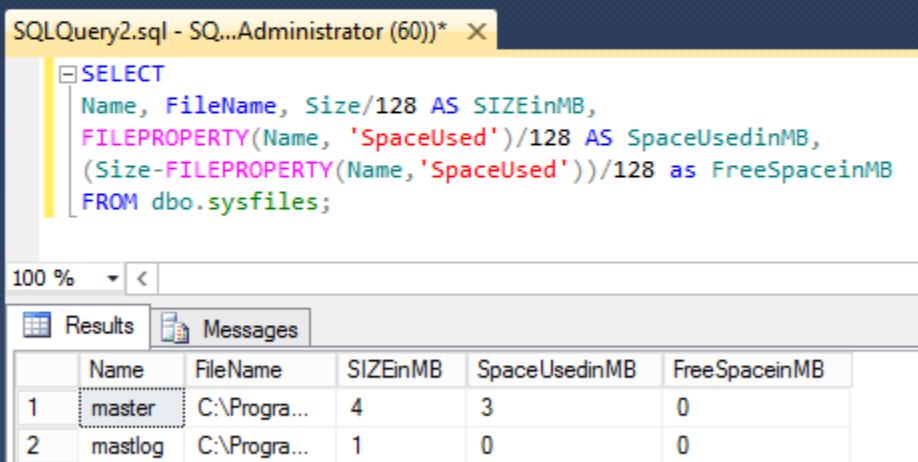
Figure 4-6: The result set of a FILEPROPERTY function.

The SELECT statement to view the file information about space used and free space in megabyte format is as follows.

```
SELECT
 Name,
 FileName,
 Size/128 AS SIZEinMB,
 FILEPROPERTY(Name, 'SpaceUsed')/128 AS SpaceUsedinMB,
 (Size - FILEPROPERTY(Name, 'SpaceUsed'))/128 as FreeSpaceinMB
FROM dbo.sysfiles;
```

This statement will retrieve details about each file in the database, the amount of file size used in the disk, and the free space available.

**FILEPROPERTY function displaying the amount of file size used in the disk, and the free space available in the server.**



```

SQLQuery2.sql - SQ...Administrator (60)*
SELECT
 Name, FileName, Size/128 AS SIZEinMB,
 FILEPROPERTY(Name, 'SpaceUsed')/128 AS SpaceUsedinMB,
 (Size-FILEPROPERTY(Name, 'SpaceUsed'))/128 as FreeSpaceinMB
FROM dbo.sysfiles;

```

|   | Name    | FileName     | SIZEinMB | SpaceUsedinMB | FreeSpaceinMB |
|---|---------|--------------|----------|---------------|---------------|
| 1 | master  | C:\Progra... | 4        | 3             | 0             |
| 2 | mastlog | C:\Progra... | 1        | 0             | 0             |

Figure 4-7: The information about disk space used and free disk space in MB.

Based on this information, you can decide whether you need to shrink the database or log or increase the database size depending on projected future growth.

## Data Movement Between Filegroups

One of the tasks that you might need to perform is to move data in a database on one drive to another drive. This is usually done for performance reasons. Consider an instance where a particular table in a database receives frequent inputs and queries. If the filegroup it belongs to resides on a slow drive, the performance might be impacted. Therefore, you might want to move this database to a different filegroup that has files on a faster file system.



Data Movement  
Between Filegroups

You cannot directly move the data or table to a different filegroup. The method to do this is to change the location of the table's clustered index. A table can have only one clustered index and is referred to a data store.

To change the location of the clustered index, you need to select the table's clustered index in SSMS and navigate to the **Properties**→**Storage** screen and change the filegroup from the **Filegroup** drop-down list. This will move the clustered index and hence the actual data itself to the file group selected from the **Filegroup** drop-down list.

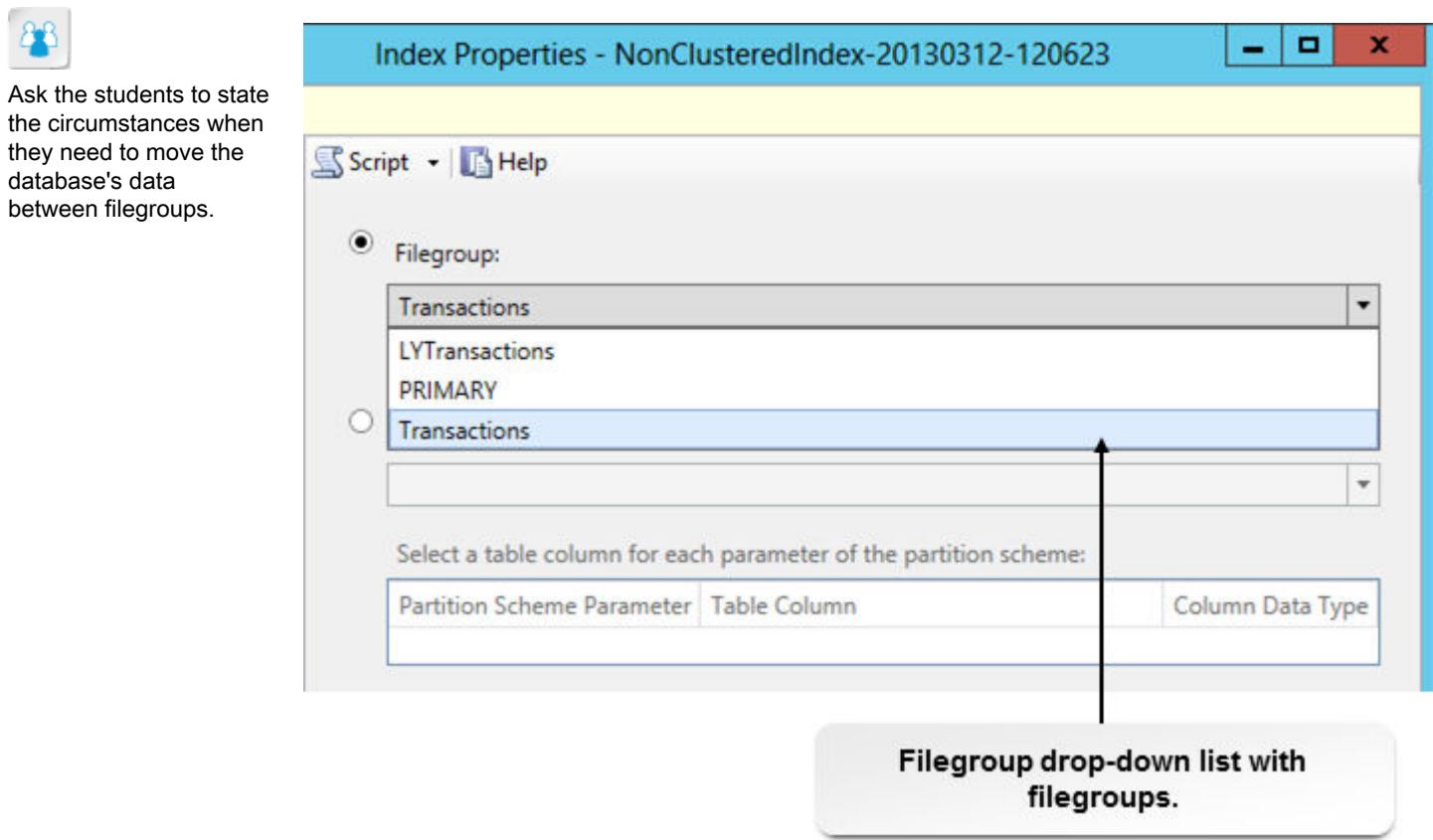


Figure 4-8: The Filegroup drop-down list in the Index Properties window that is used to move filegroups.



# ACTIVITY 4-2

## Enabling Auto Close and Auto Shrink Options

### Before You Begin

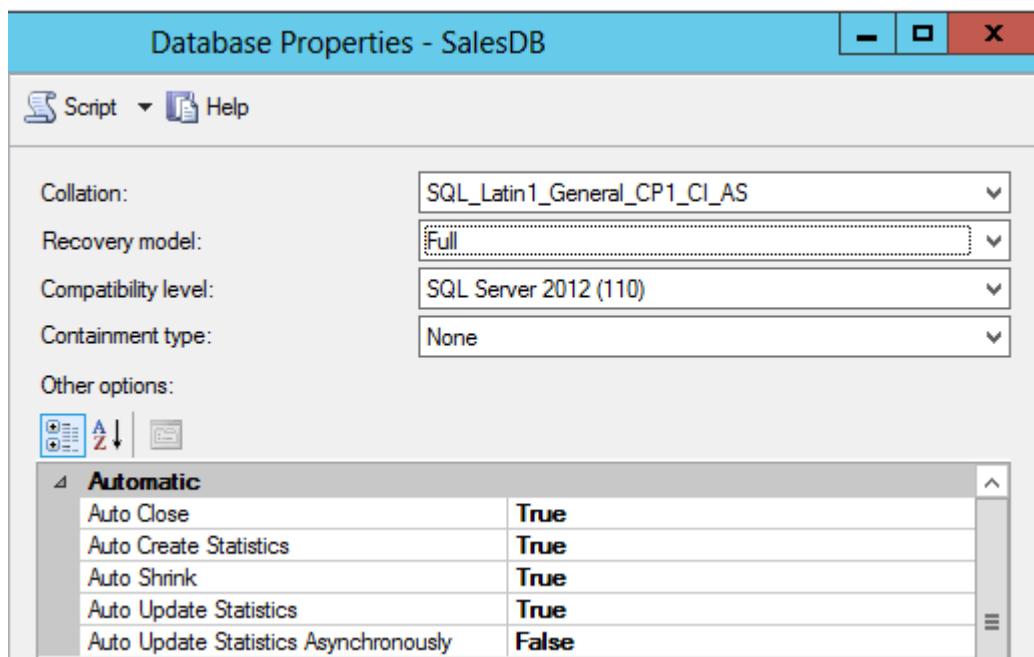
Connect to the SQL Server using Windows authentication.

1. In the **Object Explorer** pane, right-click the **SQL01** and select **Disconnect**.
2. From the **Connect** drop-down list, select **Database Engine**.
3. In the **Connect to Server** dialog box, from the **Authentication** drop-down list, select **Windows Authentication** and select **Connect**.

### Scenario

While auditing the performance of the SalesDB database, you observed that during lean hours at night, there are no transactions on the SalesDB database. You also observed that during peak hours, the SalesDB database has many data transactions leading to a large database size but the database has empty space due to the data deletions. You decide to optimize the performance of SalesDB database using the auto close and auto shrink features.

1. Enable the **Auto Close** and **Auto Shrink** options.
  - a) In the **Object Explorer** pane, expand **SQL01→Databases**.
  - b) Right-click the **SalesDB** database and select **Properties** to navigate to the Database Properties – SalesDB window.
  - c) In the Database Properties – SalesDB window, in the **Select a page** section, navigate to the **Options** page.
  - d) In the right pane, in the **Automatic** section, from the **Auto Close** drop-down list, select **True** to release all resources once the last connection is closed.
  - e) In the **Automatic** section, from the **Auto Shrink** drop-down list, select **True** to shrink the database when there is plenty of empty space in the database file.



- f) In the Database Properties – SalesDB window, select **OK** to save the settings.
2. Which statement is true with regard to auto close?
- The Auto Close option should always be set to True.
  - Auto close releases all the resources from a database when the last connection ends.
  - Auto close predicts the next connection for a database and allocates resources for the database.
  - Auto close should be enabled for all the databases.
3. Which statement is true with regard to auto shrink?
- Auto shrink reduces the size of the database whose free space is greater than 30%.
  - Auto shrink does not alter the position of the data in the shrunken files.
  - Indexes do not require a rebuild when a shrink is performed.
  - Auto shrink may block other databases from using the same physical system.
-

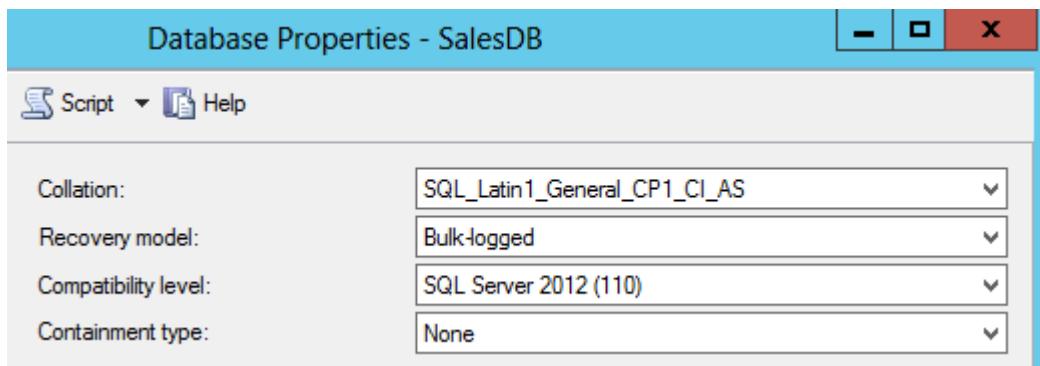
# ACTIVITY 4-3

## Changing the Database Recovery Model

### Scenario

As part of your database maintenance tasks, you observed that the performance of the database has deteriorated during bulk data transfer operations, especially when adding and modifying data. To improve the performance of the database during bulk data transfer operations, you decide to change the recovery model for the database.

1. Change the database recovery model to “Bulk logged.”
  - a) In the **Object Explorer** pane, expand **Databases**→**SalesDB**.
  - b) Right-click the **SalesDB** database and select **Properties**.
  - c) In the Database Properties – SalesDB dialog box, in the left pane, navigate to the **Options** page.
  - d) On the **Options** page, in the right pane, from the **Recovery model** drop-down list, select **Bulk logged** and select **OK** to save the settings.



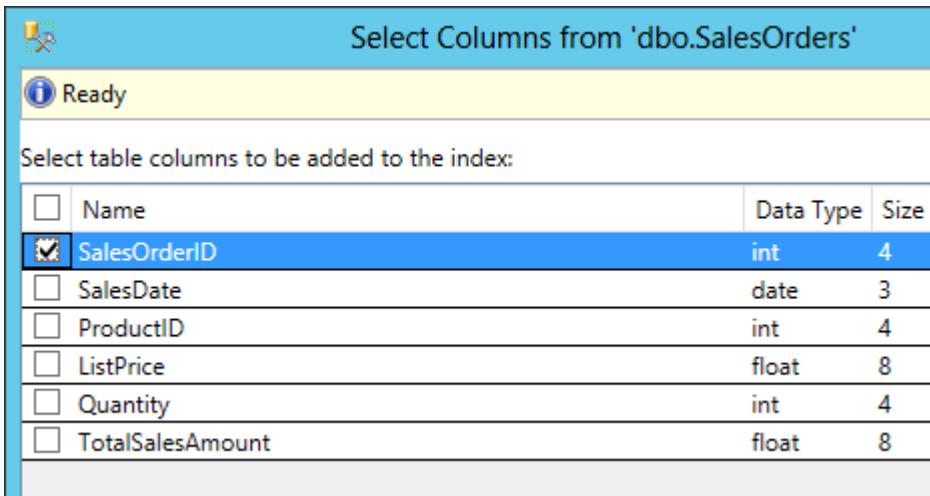
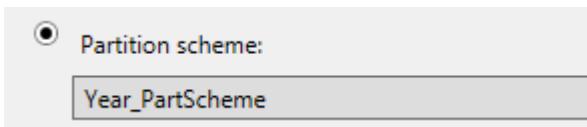
2. How is bulk-logged recovery model different from full recovery model?
  - It supports the transaction log backup.
  - You can recover the database at any point in time using bulk-logged recovery model.
  - The transaction log uses minimal space for index creation and rebuild operations.
  - Bulk-logged recovery model is more reliable than full recovery model.
3. Which statement is true with regard to simple recovery model?
  - Data can be recovered at any point in time before a failure.
  - Data added or modified after the last backup will be protected in case of a failure.
  - It supports differential backups.
  - The transaction log is truncated after a backup is taken.

## ACTIVITY 4–4

### Moving a Table from One Filegroup to Another

#### Scenario

The SalesOrders table is running on a particular partition scheme in a filegroup. Due to hard disk speed constraints, there is some deterioration in database performance when bulk transactions are performed. You would like to move the SalesOrders table to a different filegroup for improving the database performance.

1. In the SalesOrders table, create a clustered index on the SalesOrderID column.
    - a) Expand **SalesDB**→**Tables**→**dbo.SalesOrders** to view the indexes node.
    - b) Right-click the **Indexes** node and select **New Index**→**Non-Clustered Index** to create a new non-clustered index.
    - c) In the New Index window, select **Add** to add the index column.
    - d) In the Select Columns from ‘dbo. SalesOrders’ window, in the **Name** section, check the **SalesOrderID** check box to select the **SalesOrderID** column and select **OK**.
- 
- |                          | Name             | Data Type | Size |
|--------------------------|------------------|-----------|------|
| <input type="checkbox"/> | SalesOrderID     | int       | 4    |
| <input type="checkbox"/> | SalesDate        | date      | 3    |
| <input type="checkbox"/> | ProductID        | int       | 4    |
| <input type="checkbox"/> | ListPrice        | float     | 8    |
| <input type="checkbox"/> | Quantity         | int       | 4    |
| <input type="checkbox"/> | TotalSalesAmount | float     | 8    |
- e) Observe that the SalesOrderID column is displayed in the **Index key columns** section.
  - f) In the New Index window, select **OK** to save the index. This will create the index on the default partition “Year\_PartScheme.”
2. Move the data in the SalesOrder table to the Transactions filegroup.
    - a) To change the filegroup, expand **Indexes** and select the non clustered index.
    - b) Right-click the non clustered index and select **Properties**.
    - c) In the Index Properties - Non-ClusteredIndex window, in the left pane, navigate to the **Storage** page.
    - d) In the right pane, in the **Partition scheme:** section, verify that the current storage is set to the partition scheme **Year\_PartScheme**.
- 

- e) In the right pane, select the **Filegroup** option and from the drop-down list, select **Transactions** to choose the Transactions filegroup.



- f) In the Index Properties - Non-ClusteredIndex window, select **OK** to save the changes. The data in the SalesOrder table will now reside in the Transactions filegroup.  
g) In the message box stating that the index will be created, select **OK**.  
h) Close SSMS.

## Summary

In this lesson, you created a contained database to store your organization data. In addition, you also configured the database options for your database, which will enable you to add performance, reliability, and stability to your database.



Use the review questions provided to generate discussion among the participants.

**When will you enable the Auto Close and Auto Shrink options for optimizing the database performance?**

**A:** Answers may vary, but will include: for shrinking, large databases due to disk space constraints, for transferring the huge databases over the network to other sites, for conserving and optimizing the system resources.

**Which recovery model is optimum for a large transactional database?**

**A:** Answers may vary, but will include: the full and bulk logged recovery models are more suitable for large databases, they have the ability to manage large data transactions. On the other hand the simple recovery model is suitable for a small database with minimal data transactions.



**Note:** Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

# 5

# Managing SQL Server Security

**Lesson Time:** 2 hours, 30 minutes

## Lesson Objectives

In this lesson, you will:

- Configure server security.
- Create a server role.
- Configure database permissions and security roles.

## Lesson Introduction

You have managed databases in SQL Server® to effectively administer the databases in SQL Server 2012. When working with organizational data it is important to manage security within the database engine. You can manage the databases that hold confidential information by setting up database permissions to the server. Familiarizing yourself with SQL Server Security will enable you to allow appropriate access to your organizational data on the server. In this lesson, you will manage SQL Server security.

# TOPIC A

## Configure Server Security

Configuring the required server settings is the first step in managing SQL Server security. It is important to apply security settings to the databases to ensure that the database is protected and can be accessed only by an authorized user. A better understanding on configuring server security will enable to ensure appropriate data access and maintain data integrity on the server. In this topic, you will configure server security.

### Windows Accounts



*Windows accounts* are accounts that help you log in to the SQL Server. These accounts can be created either in the local Windows users or as a part of an organizational Active Directory. The *Windows users* are mainly the actions that users can perform in Windows, whereas the *Active Directory* monitors all changes and updates across all the accounts and passwords within a network. Both these types of Windows users can login to SQL Server if allowed.

Therefore, in order to allow a Windows user to log in to SQL Server, an existing administrator will need to grant them permission to do so. For this, the *administrator* who has full access across all the accounts will need to create a new SQL Server login that will map to the Windows account. This “mapping” between the Windows account and SQL Server, where the users are given full access to Windows or Active Directory, enables the users to log in to SQL Server.

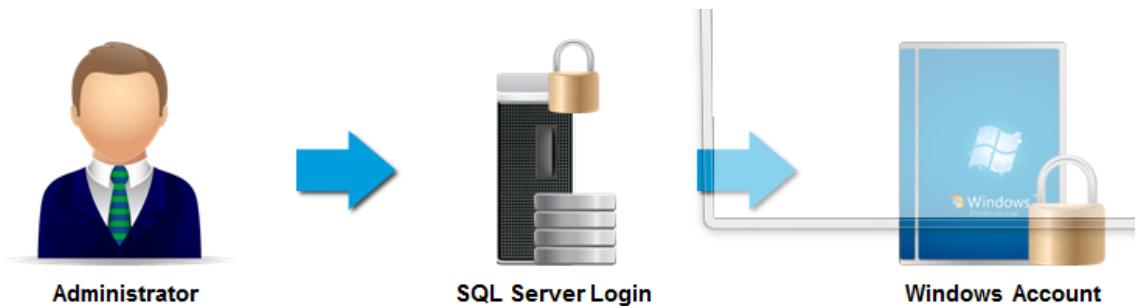


Figure 5-1: The SQL Server login has to map to a windows account.

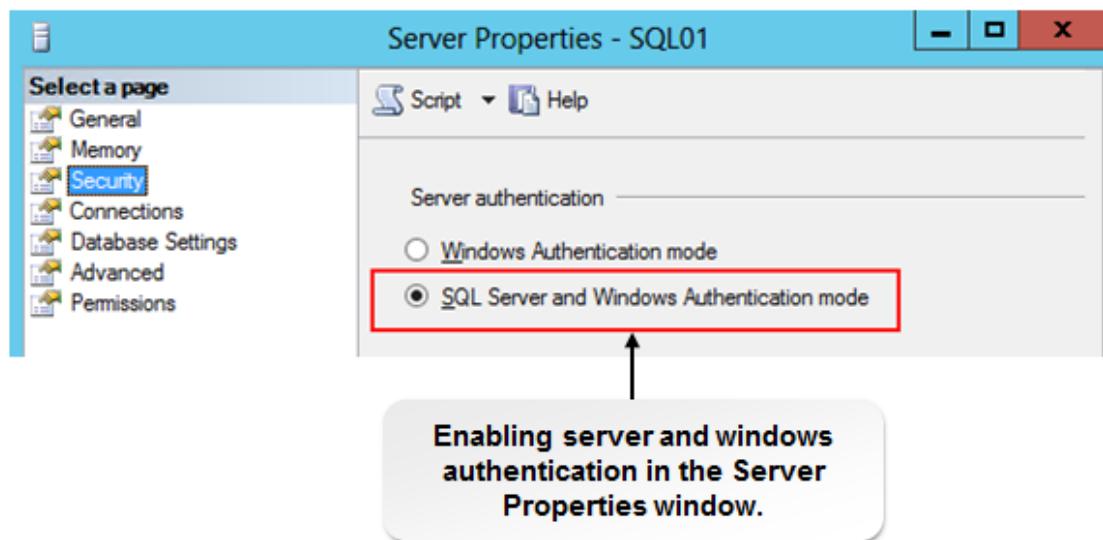
### SQL Server Accounts



*SQL Server accounts* are login accounts that exist only within the server instance and cannot log in to Windows itself. These accounts have access only to the instance in which they are created and are authenticated by the SQL server.

SQL Server accounts are considered to be less secure than Windows accounts due to the way they are implemented. Therefore, SQL Server authentication is turned off by default in each instance and needs to be enabled if SQL Server logins are to be allowed. The option to enable the SQL Server authentication is available in the Security settings and is displayed in the **Server Properties** page of that instance. You can enable either Windows logins or both Windows and SQL Server logins on the security settings.

|  |                                                                               |
|--|-------------------------------------------------------------------------------|
|  | <b>Note:</b> Only SQL Server logins are not allowed in this security setting. |
|--|-------------------------------------------------------------------------------|



## Login Accounts

*Login accounts* are to be created for every user within the network to monitor the activities performed within a database. You need to use these login accounts to access SQL Server. After creating the Windows or SQL based logins at the server level, you will also need to create or add the logins to the databases to which they have access.

Microsoft SQL Server 2012 has an in-built function that allows you to have login accounts associated only with a database and not with a Windows or SQL Server account. The login accounts of the Windows, SQL, or database work at the level of the database to provide access to its logins.

## Security Models

The *security model* of SQL Server is based on granting and denying permissions on logins. The server contains an extensive list of all activities that are possible to do at the server, database, table, and row level. Therefore, logins can be granted extremely granular by either granting or denying the permissions to access these resources.

Therefore a security model is established across the logins accounts that are created for users to enable them to access any data from the database.

|  |                                                                                                                                                                                                         |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Note:</b> Security models control access based on job roles.                                                                                                                                         |
|  | <b>Note:</b> For more information about server security, check out the <b>LearnTO Manage SQL Server Security</b> presentation by selecting the <b>LearnTO</b> tile on your LogicalCHOICE Course screen. |

## The GRANT Statement

The *GRANT statement* allows you to provide certain permissions to a login on the specified resources.



Ask the students which account type would they prefer to create and work with at ease.



You may want to show **LearnTO Manage SQL Server Security** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.

Login accounts that are created need to have specific permissions to alter any changes in the database. You have to use the GRANT statement to grant permissions to the user in order to enable him to work with the database.

There are two different ways that the GRANT statement can be used. The first is to grant permissions for activities on the server.

The syntax of a GRANT statement to grant permissions for activities on the server is as follows.

```
GRANT <permission(s)> TO <login(s)> [WITH GRANT OPTION];
```

Another way of using a GRANT statement is to grant permissions on objects such as tables and stored procedures.

The syntax of a GRANT statement to grant permissions on objects such as tables and stored procedures is as follows.

```
GRANT <permission> ON [OBJECT::]<object name> TO <login(s)> [WITH GRANT OPTION];
```

## Example

A GRANT statement to grant permissions for activities on a server is as follows.

```
GRANT ALTER ANY DATABASE TO JackB WITH GRANT OPTION;
```

This statement will allow the user “JackB” the permission to alter any database on the server. The WITH GRANT OPTION allows you to pass on the same permission to any other user.

A GRANT statement to select a database and grant permission to a single user is as follows.

```
GRANT SELECT ON OBJECT::SalesDB TO JackB;
```

This statement will allow JackB to retrieve data from the SalesDB table, but will not allow him to grant the same permission to someone else.



**Note:** In both cases, you can use ALL to specify all permissions on the server or object.



## The DENY Statement

### The DENY Statement

The *DENY statement* allows you to take away the permissions that were given to the user logins. The DENY statement also has two versions which follow the exact same syntax as the GRANT statement, except that, the permission specified will be taken away from the logins.

The syntax of a DENY statement to deny permissions for activities on the server is as follows.

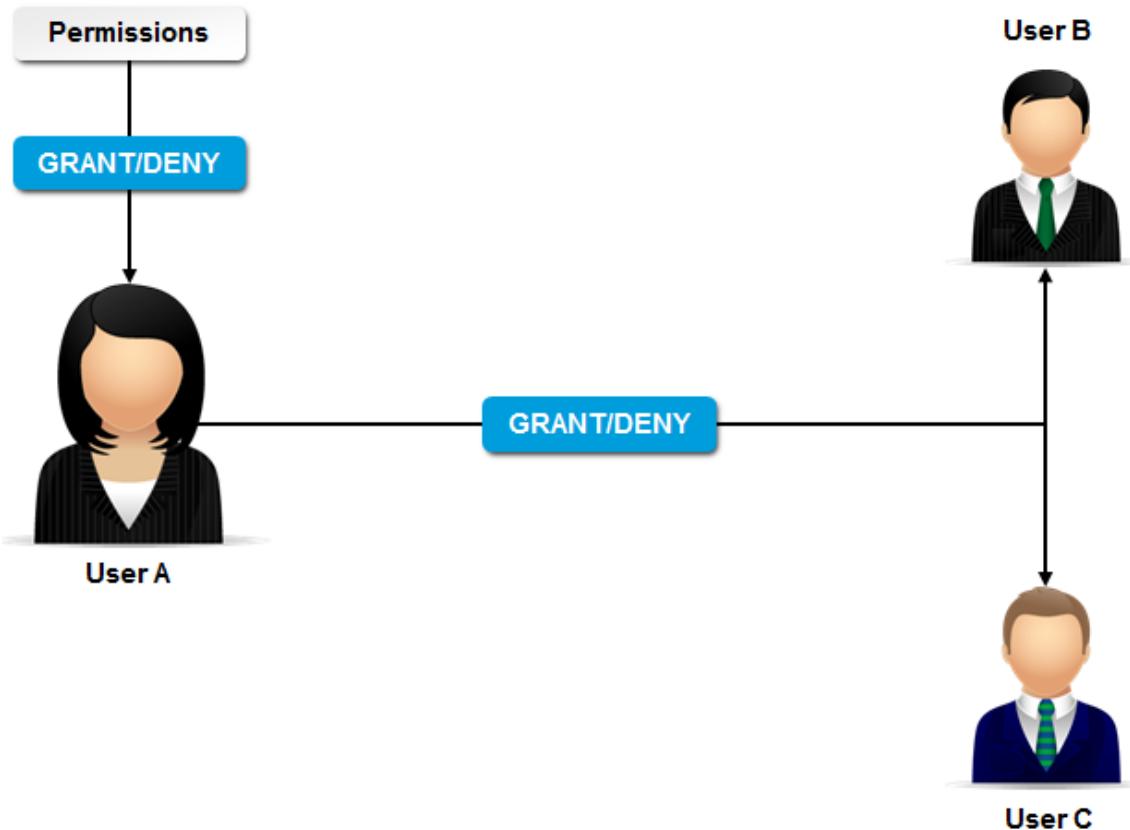
```
DENY <permission(s)> TO <login(s)> [CASCADE];
```

The syntax of a DENY statement to deny permissions on objects such as tables and stored procedures is as follows.

```
DENY <permission> ON [OBJECT::]<object name> TO <login(s)> [CASCADE];
```

The CASCADE option specifies that in case the specified login to which you are denying permission has passed that same permission on to other logins, those logins will also get denied with the same permissions.

Therefore, if User A was granted a permission and he passes the permission to users B and C, if User A is now denied permissions with the CASCADE option set, then users B and C will also lose the permissions.



**Figure 5–3:** The **CASCADE** option enable you to grant and deny permissions between users.



**Note:** A DENY statement will not respond if the CASCADE option is not specified when denying a permission that was approved by the GRANT statement.

## Example

The DENY statement to deny permissions for activities on the server is as follows.

```
DENY <TestSQL> TO <JohnS> [CASCADE];
```

This statement will deny the permissions assigned to JohnS from accessing the TestSQL database. The CASCADE option will revoke all the access for the logins created by JohnS. The DENY statement to deny permissions on objects such as tables and stored procedures is as follows

```
USE AdventureWorks2012 DENY SELECT ON OBJECT : : Sales.Address TO JohnS;
Go
```

This statement will deny the SELECT permission to JohnS to work on the table named Sales.Address in the AdventureWorks2012 database.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Configure Server Security

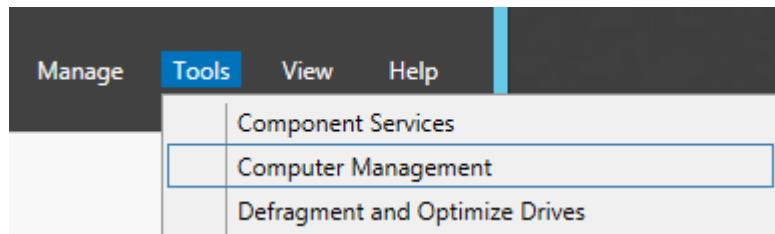
# ACTIVITY 5–1

## Creating a Windows Account

### Scenario

The marketing team of Adventure Works Cycles wants you to create a new windows account to provide a separate login for the employees in the marketing team to access the required data from the server. You decide to create a Windows account using the Computer Management window and grant access to the required databases on the server.

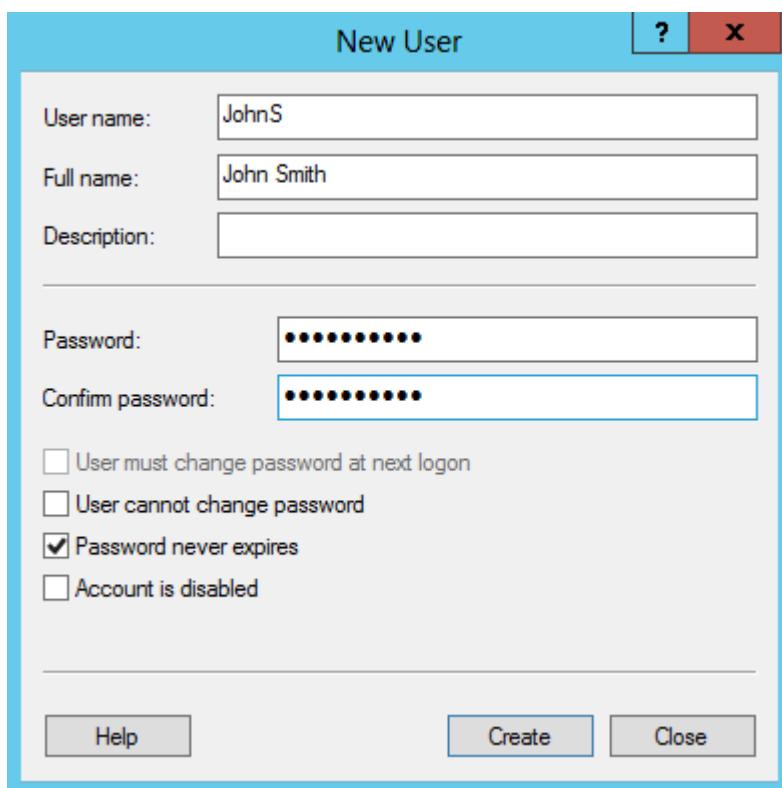
1. Launch Server Manager and add a new user account
  - a) In the desktop interface of Windows Server 2012, move the mouse pointer to the bottom left of the task bar and select **Server Manager** icon to launch it.
  - b) In the Server Manager window, in the top right corner, select **Tools→Computer Management**.



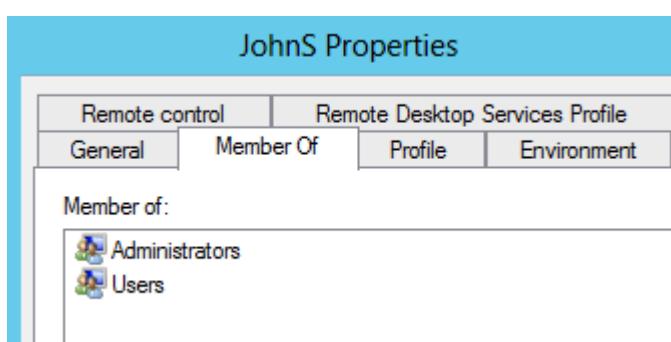
- c) In the Computer Management window, in the **Computer Management (Local)** pane, expand **Local Users and Groups** and select **Users**.
- d) To add a new user to the server, right click the **Users** folder and select **New User**.
- e) In the **New User** dialog box, type the following credentials:
  - Username: **JohnS**
  - Full Name: **John Smith**
  - Password/Confirm Password: **pass@word1**
  - User must change password at next logon: **Off** (uncheck the check box)
  - Password never expires: **On** (Check the check box)



**Note:** By default the “User must change password at next logon” will be turned on. Please ensure to turn it off to set the other properties.



- f) On the **New User** dialog box, select **Create** to create the new Windows account, select **Close** to exit the dialog box.
2. Grant access to the database on the server.
- In the Computer Management window, in the users list, double-click on the newly created account called “JohnS.”
  - In the **JohnS Properties** dialog box, select the **Member Of** tab.
  - Select **Add** and in the **Select Groups** dialog box, type in **Administrators** and select **Check Names** to search for the name in the server.
  - Select **OK** to add the user as a member of the Administrators group.



Adding the new user credentials in the **Member Of** tab is not a recommended practice. It is done here to ensure that the new user can login to the Windows Server directly for this activity.



Check with the participants and ensure that they are able to create the Windows account on the Administrator login.

- e) Log out from Windows Server.

## ACTIVITY 5–2

### Granting Rights on a Database to a Windows Account

#### Before You Begin

Log in to Windows Server as the new user **John Smith** with the password **pass@word1**.

#### Scenario

As database administrator in Adventure Works Cycles, you want to grant rights to the database for the users who have a SQL login mapped with their Windows user account. Therefore, you are asked to grant the required permission to a specific user.

1. Launch the SQL Server Management Studio using Windows Authentication.
  - a) In the desktop interface of Windows Server 2012, move your mouse pointer to the bottom-left corner of the screen and select the **Metro Interface** to view the Start screen.
  - b) In the Start Screen, type SQL and select **SQL Server Management Studio 2012**.
  - c) Select **SQL Server Management Studio** to launch SSMS.



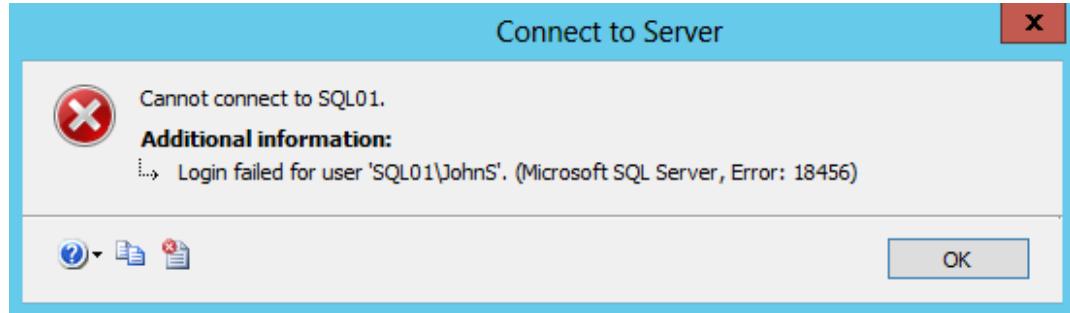
**Note:** Wait for a few minutes until the **Connect to Server** dialog box is displayed.

- d) In the **Connect to Server** dialog box, in the **Server Type** drop-down list, verify that **Database Engine** is selected and in the **Server Name** drop-down list, verify that **SQL01** is automatically selected.



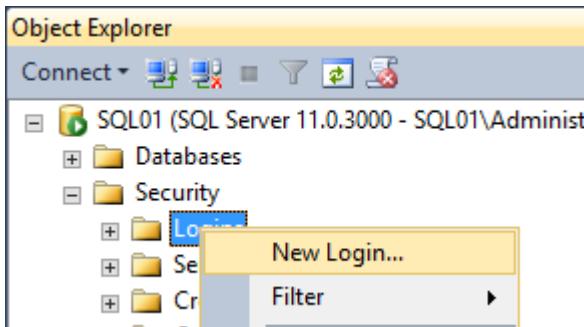
**Note:** The server names are specific to every domain. Please ensure you use the default server name.

- e) In the **Authentication** drop-down list, verify that **Windows Authentication** is selected and select **Connect** to connect to the server.
- f) The server will throw an error message stating that the login failed despite the user being a Windows administrator because the user has not been granted the rights to access any database. Select **OK** and close the **Connect to Server** message box.

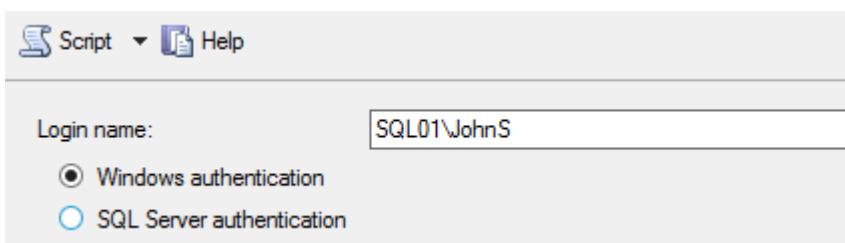


- g) Close the SSMS interface.
  - h) Log out from “JohnS” login on Windows Server.
2. Create a new login account.
    - a) Log in to Windows Server as **Administrator**.

- Open SSMS. In the **Connect to Server** dialog box, in the **Server Type** drop-down list, verify that **Database Engine** is selected and in the **Server Name** drop-down list, verify that the name of your server is selected.
- In the **Authentication** drop-down list, verify that **Windows Authentication** is selected and select **Connect** to connect to the server.
- In the **Object Explorer** pane, expand **Security**.
- Right-click **Logins** and select **New Login**.

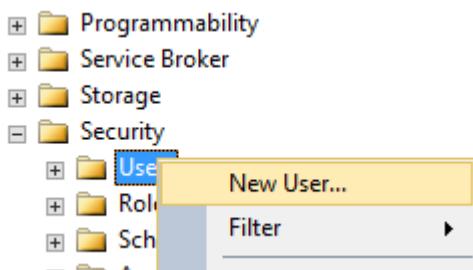


- In the Login-New window, select **Search**.
- In the **Select User or Group** dialog box, type **JohnS** and select the **Check Names** button. Select **OK** to continue.
- Ensure that **Windows Authentication** is selected and change the default database to **SalesDB**. Select **OK** to save the changes made.

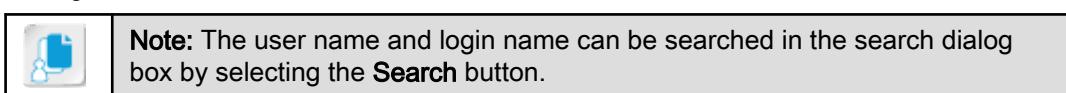


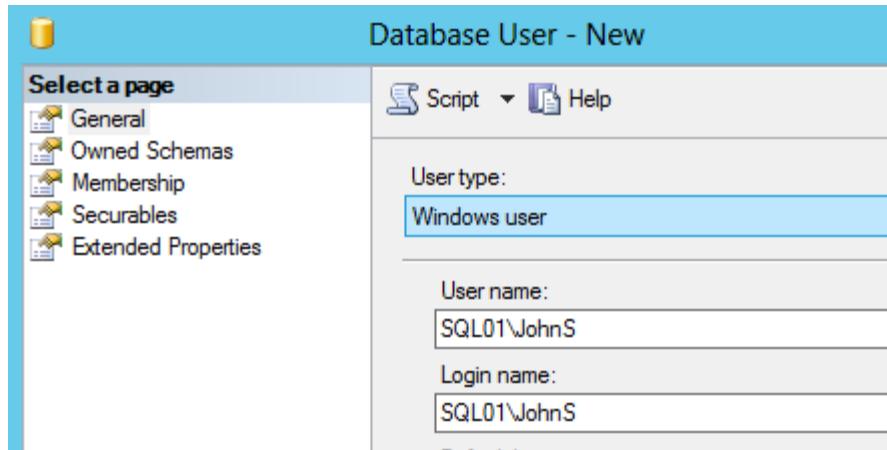
### 3. Create a new user to the database.

- In the **Object Explorer** pane, expand **Databases**→**SalesDB**→**Security**.
- Right-click **Users** and select **New User**.

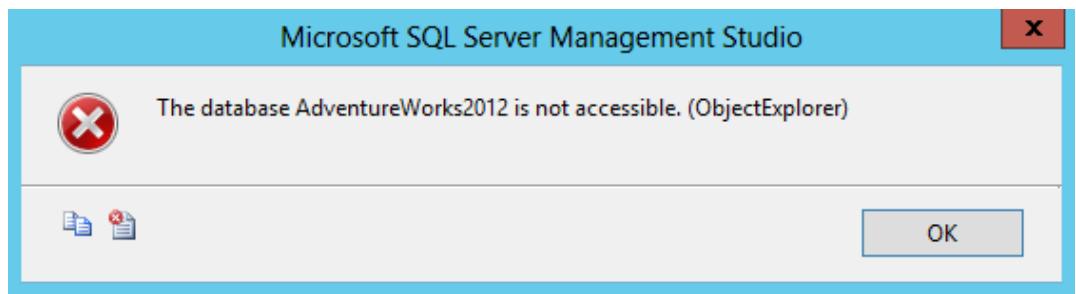


- In the Database User-New window, type the following credentials:
  - User Type: **Windows user**
  - User Name: **SQL01\JohnS**
  - Login Name: **SQL01\JohnS**





- d) In the Database User-New window, select **OK** to save the changes.
- e) Close SSMS and Logoff/Sign Out from the **Administrator** login.
4. Connect to the Windows Server using Windows Authentication.
- Log in to Windows Server as **John Smith**.
  - Open SSMS. In the **Connect to Server** dialog box, in the **Server Type** drop-down list, verify that **Database Engine** is selected.
  - In the **Connect to Server** dialog box, from the **Server Name** drop-down list, select **SQL01**.
  - In the **Authentication** drop-down list, verify that **Windows Authentication** is selected and select **Connect** to connect to the server. This time the server will successfully connect.
5. Verify if the SalesDB database is listed in SSMS.
- In the **Object Explorer** pane, expand **Databases→SalesDB**.
  - On the **Standard** toolbar, select the **New Query** to change the default database.
  - On the **SQL Editor** Toolbar, in the **Available Databases** drop-down list, select **AdventureWorks2012**.
  - An error message will appear stating that the database is not accessible. This happens because the Windows login has not been assigned to that database. In the dialog box that appears, select **OK**. This ensures that the SalesDB database has been successfully configured on Windows Server for the user login id JohnS.



- e) Close SSMS and Logoff/Sign Out from JohnS login.

# ACTIVITY 5–3

## Creating a SQL Server Account

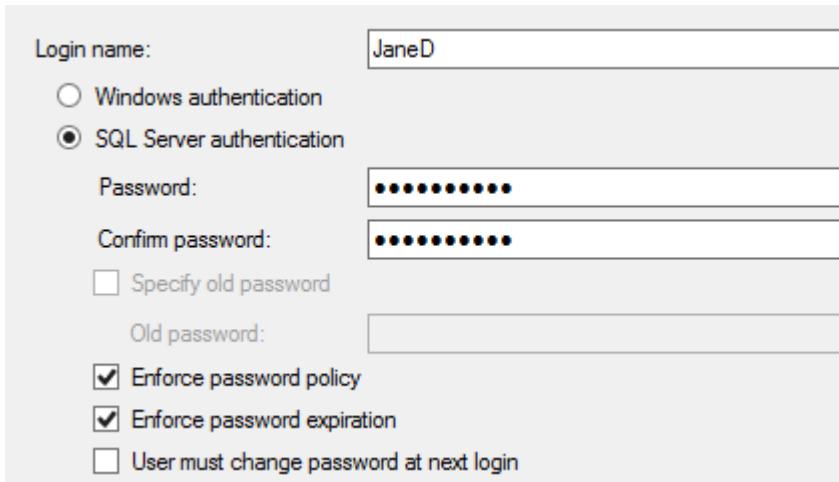
### Before You Begin

1. Log in to Windows Server as **Administrator**.
2. Launch SSMS and in the **Server Type** drop-down list, verify that **Database Engine** is selected and in the **Server Name** drop-down list, verify that the name of your server is selected.
3. In the **Authentication** drop-down list, verify that **Windows Authentication** is selected and then connect.

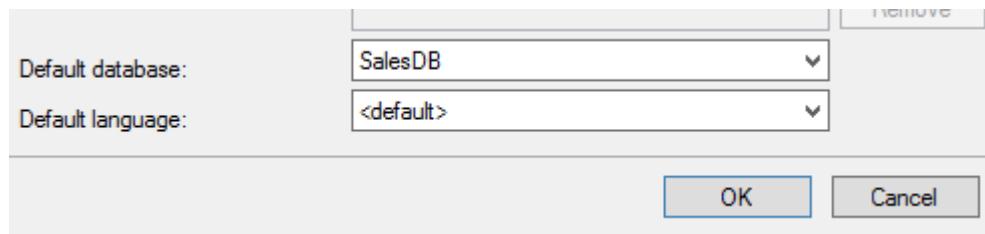
### Scenario

Management of Adventure Works Cycles wants to expand their business and therefore wants a third party approval on the Sales database. You are asked to provide an external partner access to the sales database for some work. You decide to create a SQL Server login for the third-party user to perform this task.

1. Create a new user login in SQL Server.
  - a) In the **Object Explorer** pane, expand **Security** and select **Logins**.
  - b) Right-click the **Logins** node and from the fly-out pane, select **New Login**.
  - c) In the **Login-New** window, type the following credentials:
    - Login Name: **JaneD**
    - Password/Confirm Password: **pass@word1**
  - d) In the **Login-New** window, select **SQL Server authentication** and uncheck the **User must change password at next login** check box.

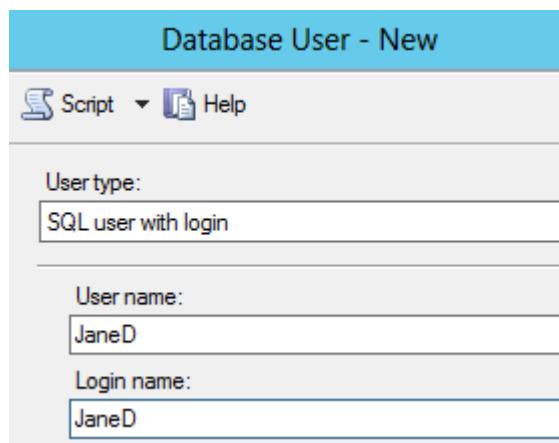


- e) In the default database list, change the <default> database to **SalesDB** in order to be able to login using the **SQL Server Authentication** login. Select **OK** to save the new SQL Server login.



2. Create a new database user account

- a) In the **Object Explorer** pane, expand **Databases**→**SalesDB**→**Security** folder and select **Users**.
- b) Right-click **Users** and select **New User**.
- c) In the Database User-New window, type the following credentials:
  - User type: **SQL user with login**
  - User name: **JaneD**
  - Login name: **JaneD**



- d) In the Database User-New window, select **OK** to save the new database user name. The required account has been created and the third party user can access the account now.
- e) Close SSMS.

## ACTIVITY 5–4

### Granting Rights on a Database to a SQL Server Account

#### Scenario

The global sales head of Adventure Works Cycles wants to approve on certain user permissions on the existing sales database. You are asked to grant the required access to the database using SQL Server credentials to enable her to accomplish this task.

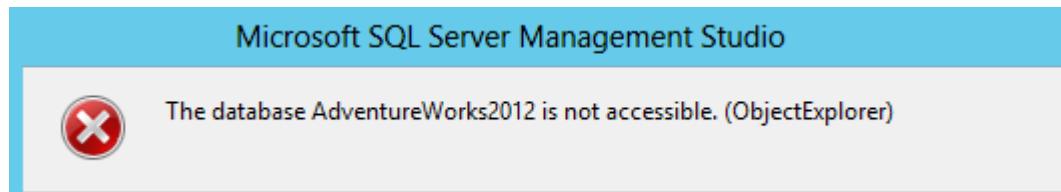
1. Connect to the database using SQL Server Authentication.
  - a) Open SSMS. In the **Connect to Server** dialog box, in the **Server type** drop-down list, verify that **Database Engine** is selected and in the **Server Name** drop-down list, verify that the name of your server is selected.
  - b) In the **Authentication** drop-down list, verify that **SQL Server Authentication** is selected. Type the login as *JaneD* and password as *pass@word1* and select **Connect** to connect to the server. The server will successfully connect to the **SalesDB** database.



2. Verify that the SalesDB database is connected through SQL Server Authentication.
  - a) In the **Object Explorer** pane, expand **Databases** → **SalesDB** → **Security** → **Users** and check if **JaneD** is listed in the user list.



- b) On the **Standard** toolbar, select the **New Query** to change the default database.
- c) On the **SQL Editor** toolbar, from the available database drop-down list, select **AdventureWorks2012** to check if this database has access to SQL Server.
- d) An error message will appear stating that the database is not accessible. This happens because the SQL Server login has not been assigned to that database. The access rights have been given only to the **SalesDB** database. Select **OK** to close the error message.



- e) Close SSMS.

# TOPIC B

## Create Server Roles

You have configured the required server security settings by granting permissions to the required accounts. It is indeed necessary to manage these server permissions in a closed network. Therefore, you will need to create server role groups for easy management. Familiarizing yourself with creating server roles will enable you to work with ease when there are multiple users within the group. In this topic, you will create a server role.

### Server Roles

*Server roles* are roles that group multiple users under a single name so that the settings applied to the particular group applies to all the users within that group as well. These roles are like administrative groups in Windows and Active Directory and can consist of both SQL Server and Windows accounts.

Administrators often find themselves having to work with multiple users at a time. However, assigning each user the appropriate permissions is a time consuming and repetitive job. To reduce this tedious task individually, the administrator has an option of assigning server level permissions using server roles. SQL Server has two types of server roles namely, fixed server roles and custom server roles.



Server Roles

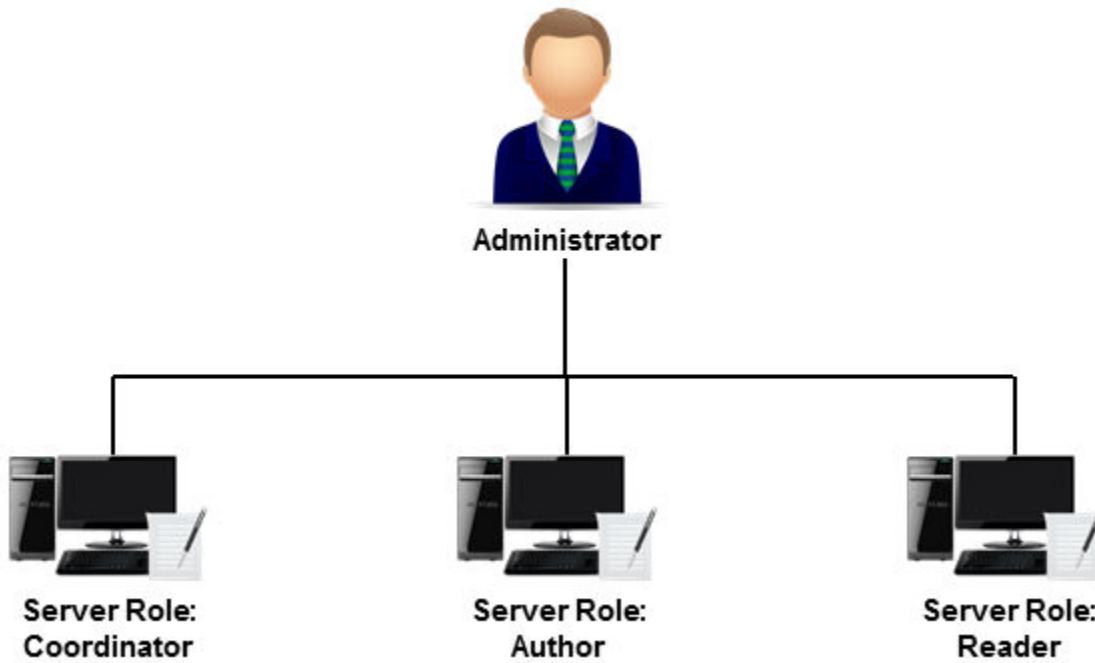


Figure 5–4: An administrator assigning server roles to its users.

### Fixed Server Roles

*Fixed server roles* are roles that are pre-defined within SQL Server and they cannot be deleted or modified in any manner other than adding accounts to them. They assign certain privileges to a user.

There are nine fixed server roles that can be used in SQL Server.



Fixed Server Roles

| <b>Server Role</b> | <b>Used to</b>                                               |
|--------------------|--------------------------------------------------------------|
| SysAdmin           | Perform any operation on the server instance.                |
| ServerAdmin        | Change server level settings and start or stop the instance. |
| SecurityAdmin      | Add, modify, or remove users and permissions.                |
| ProcessAdmin       | Start and end processes running within SQL Server.           |
| SetupAdmin         | Add or delete linked servers.                                |
| BulkAdmin          | Run the BULK INSERT command to add multiple rows.            |
| DiskAdmin          | Manage SQL files on the disk.                                |
| DbCreator          | Add, modify, delete, or restore any database.                |
| Public             | Manage all users as members of this role by default.         |

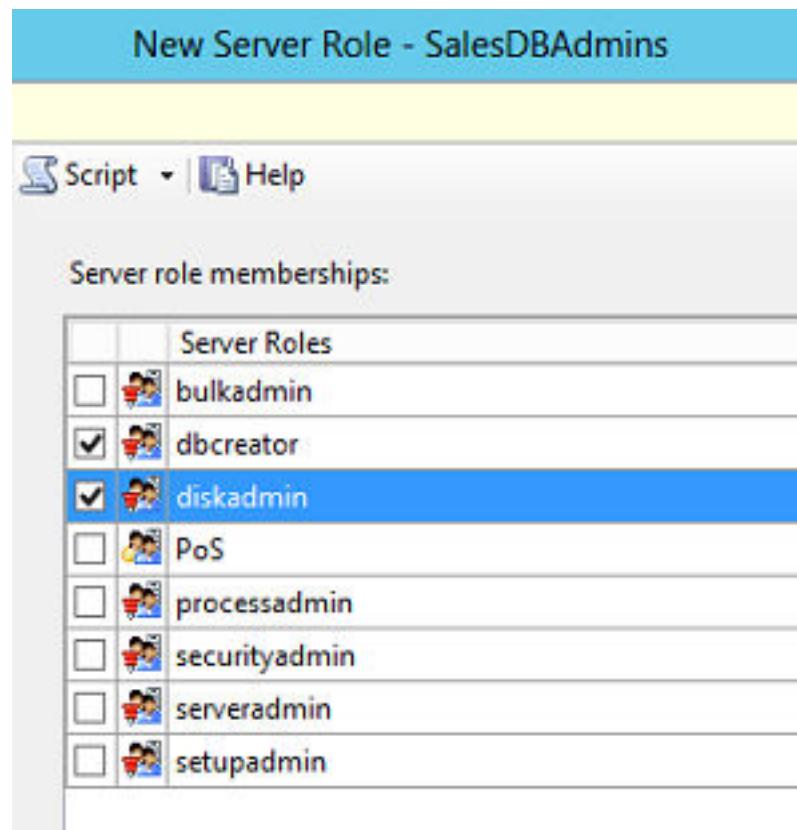


Figure 5-5: Fixed server roles assigned to a new server role named SalesDBAdmins.

## Custom Server Roles

*Custom server roles* are roles that an administrator can create and assign any permission on it. While creating a custom server role, you have to provide the name of the role and the database to which the role has access to or is denied from, by granting or denying access.



**Note:** You can combine multiple fixed server roles to a single role by using a custom server role.

## User-Defined Server Roles

*User-defined server roles* are built-in roles that make it easier to grant or revoke access to a server at any point in time. These roles are always useful while granting permissions, sometimes you would need to create groups of users who have certain specific set of permissions on the server. Therefore, user-defined server role can be created and used in any place where a fixed server role can be used.

While creating the role you can assign any descriptive name to it. Logins need to be assigned to the role; both Windows as well as SQL logins are allowed within a server. When creating the role, it is mandatory to specify the permissions that are granted or denied to the members of that role. You can also use a combination of existing server roles to create a new role with permissions inherited from the others.

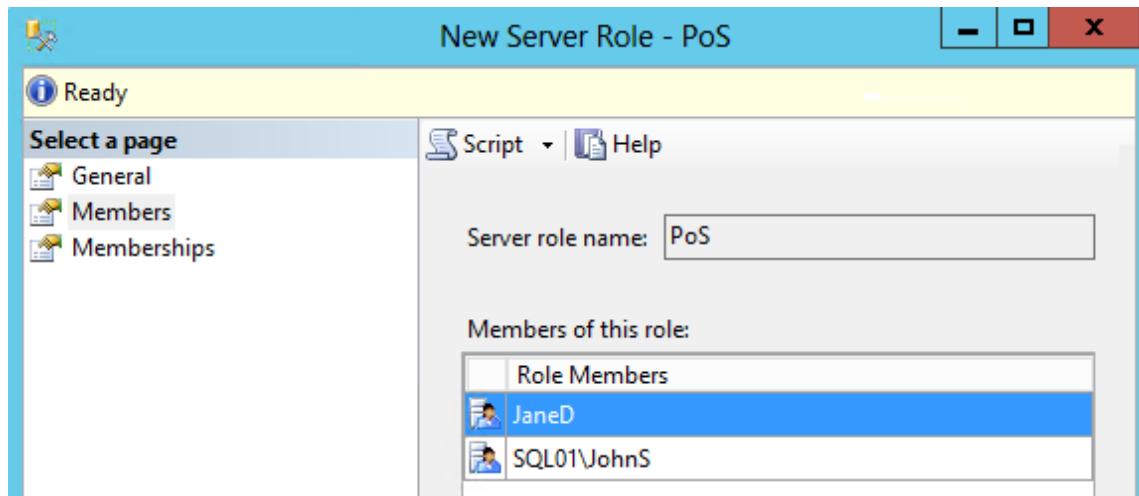


Figure 5–6: The role member *JaneD* is added to the *PoS* server role.



User-Defined Server Roles



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Create Server Roles



Ask the students to give an example of a server role that they have created in an administrative job.

# ACTIVITY 5–5

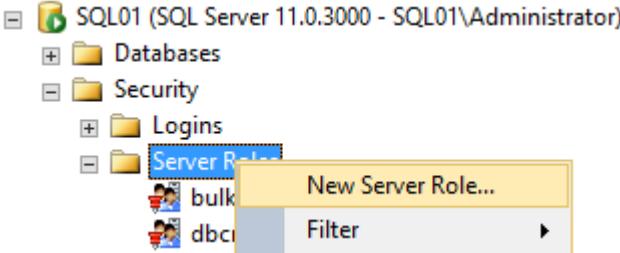
## Creating a Server Role

### Before You Begin

1. Launch SSMS and from the **Server Type** drop-down list, verify that **Database Engine** is selected.
2. In the **Server Name** drop-down list, verify that the **SQL01** is selected.
3. In the **Authentication** drop-down list, select **Windows Authentication** and then select **Connect**.

### Scenario

The sales team of Adventure Works Cycles wants to create a new group of users to manage the database structures of the servers in the company. You are asked to create a new server role to contain these users and use the permissions inherited from some other server roles as well to create this new group of users.

- 
1. Create a new server role and grant access to the server role.
    - a) In the **Object Explorer** pane, expand the **Security** folder and select **Server Roles**.
    - b) Right-click the **Server Roles** node and select **New Server Role...**
    - c) In the **New Server Role** dialog box, in the **Server Role Name** text box, type **SalesDBAdmins**.
    - d) In the **Securables** section, check the **Servers** check box and ensure that the **SQL01** check box is checked.
    - e) In the **Permissions for SQL01** section, check the check boxes in the **Grant** column against the **Alter any database** and **Create any database** rows.

 **Note:** The server names are specific to every domain. Please ensure you use the default server name from your domain.

| Permissions for SQL01: |         |                                     |                          |  |
|------------------------|---------|-------------------------------------|--------------------------|--|
| Explicit               |         |                                     |                          |  |
| Permission             | Grantor | Grant                               | With Grant               |  |
| Alter any credential   |         | <input type="checkbox"/>            | <input type="checkbox"/> |  |
| Alter any database     |         | <input checked="" type="checkbox"/> | <input type="checkbox"/> |  |
| Alter any endpoint     |         | <input type="checkbox"/>            | <input type="checkbox"/> |  |



**Note:** If you want to deny access to the server role in the Permissions for (server name): Check the check box in the **DENY** column against **Alter any Database** and **Create any database**.

- f) In the **Select a page** section, select the **Members** page.
- g) Select the **Add** button to add the new object name.
- h) In the **Select Server Login or Role** dialog box, type *JaneD* and then select **Check Names**. Select **OK** to close the **Select Server Login or Role** dialog box.



**Note:** Observe that in the “Enter the object names to select (examples):” section, the login named [JaneD] will be enclosed in square brackets ([]).

2. Add another object name to the existing server role.
- a) On the **Members** page, select the **Add** button to add a new object name on the **New Server Role** dialog box.
- b) In the **Select Server Login or Role** dialog box type *JohnS* and then select **Check Names**. The login name will appear along with the server name as **SQL01\JohnS**. This is because **JohnS** is a SQL account login.
- c) Check the check box against **JohnS** and select **OK**. In the **Select Server Login or Role** dialog box, select **OK**. Now, there are two members existing in this server role, one a Windows login and the other a SQL login.

| Role Members |             |
|--------------|-------------|
|              | JaneD       |
|              | SQL01\JohnS |

3. Grant permissions to server roles.
- a) In the **Select a page** section, select the **Memberships** page.
- b) On the **Membership** page, in the **Server role memberships** list, select **dbcreator** and **diskadmin** to inherit permissions from the server.

| Server Roles                        |           |
|-------------------------------------|-----------|
| <input type="checkbox"/>            | bulkadmin |
| <input checked="" type="checkbox"/> | dbcreator |
| <input checked="" type="checkbox"/> | diskadmin |

- c) Select **OK** to create the new server role.



Ask the students to create a server role with the name PoS and Grant permissions to the database.

# ACTIVITY 5–6

## Setting a Server Role to Modify Users

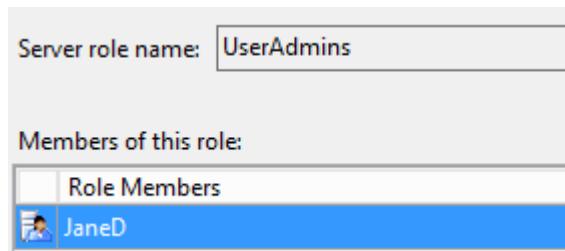
### Scenario

The IT team of Adventure Works Cycles wants to delegate user management rights to a few sub-administrators in the network. For this, you decide to add a new server role that has the ability to create and change other users and roles which help the administrators to monitor the logins for the database within the network.

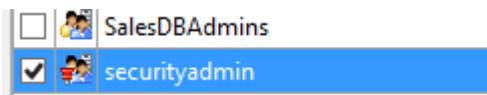
1. Create a new server role to the database.
  - a) In the **Object Explorer** pane, right-click the **Server Roles** node and select **New Server Role**.
  - b) In the **New Server Role** dialog box, in the **Server Role Name** text box, type **UserAdmins**.
  - c) In the **Securables** section, check the **Servers** check box and check the **Logins** and **Server Roles** check boxes.
  - d) In the **Permissions for SalesDBAdmins** section, check all the check boxes in the **Grant** column to grant all the permissions for SalesDBAdmins server role.

| Permissions for SalesDBAdmins: |         |                                     |                          |                          |
|--------------------------------|---------|-------------------------------------|--------------------------|--------------------------|
| Explicit                       |         |                                     |                          |                          |
| Permission                     | Grantor | Grant                               | With Grant               | Deny                     |
| Alter                          |         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Control                        |         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Take owners...                 |         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| View definiti...               |         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

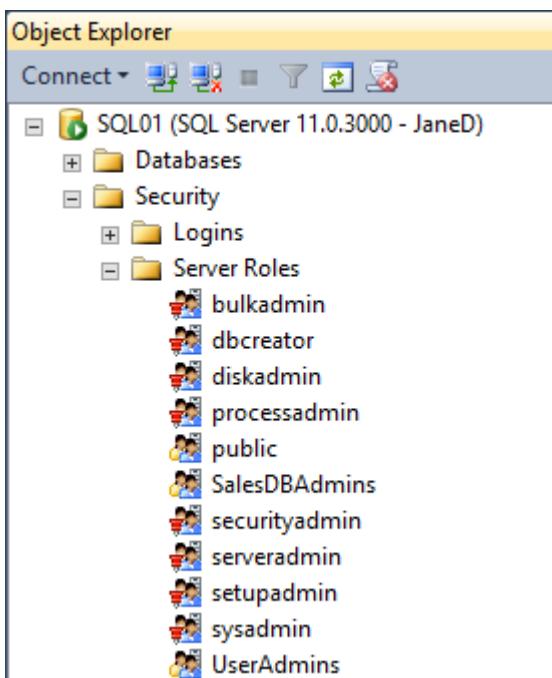
- e) In the **Select a page** section, select the **Members** page.
- f) Select the **Add** button to add a new object name.
- g) In the **Select Server Login or Role** dialog box, in the **Enter the object name to select** text area text box, type **JaneD** and then select **Check Names**. Select **OK** to add **JaneD** to the role.



2. Add members to the server role and grant permissions to them.
  - a) On the **Select a page** section, select the **Memberships** page.
  - b) On the **Server role memberships** list, select **securityadmin** to inherit permissions from the server.



- c) Select **OK** to save the new server role.
- d) In the **Object Explorer** pane, right-click **SQL01** and select **Disconnect**.
3. Connect to the server using SQL Server Authentication.
- In the **Object Explorer** pane, select **Connect** and select **Database Engine**.
  - In the **Server Name** drop-down list, ensure that **SQL01** is selected.
  - From the **Authentication** drop-down list, select **SQL Server Authentication**, in the **Login** text box, type **JaneD** and in the **Password** text box, type **pass@word1** and select **Connect** to connect to the server.
  - In the **Object Explorer** pane, expand **SQL01**→**Security**→**Server Roles**, and observe that the server roles for the JaneD login are displayed.



- e) In the **Object Explorer** pane, right-click the server name with JaneD login and select **Disconnect**.

# TOPIC C

## Configure Database Security

You have created server roles to grant access to a specific group of users. It's time for you to configure the security settings for the databases located in the server. A better understanding on configuring database security will enable you to control the actions performed by users at the database level. In this topic, you will configure database permissions and security roles.

### Database Level Roles



*Database level roles* are roles that are created at the database level. Similar to roles that are defined at the server level, you can create groups or roles at the database level. When such a role is created, you can specify which objects of the database the role has access to and what kind of access it has.

At the database level, you are also permitted to change the compatibility level of the server that you are working with, if the database requires the earlier version of the server.

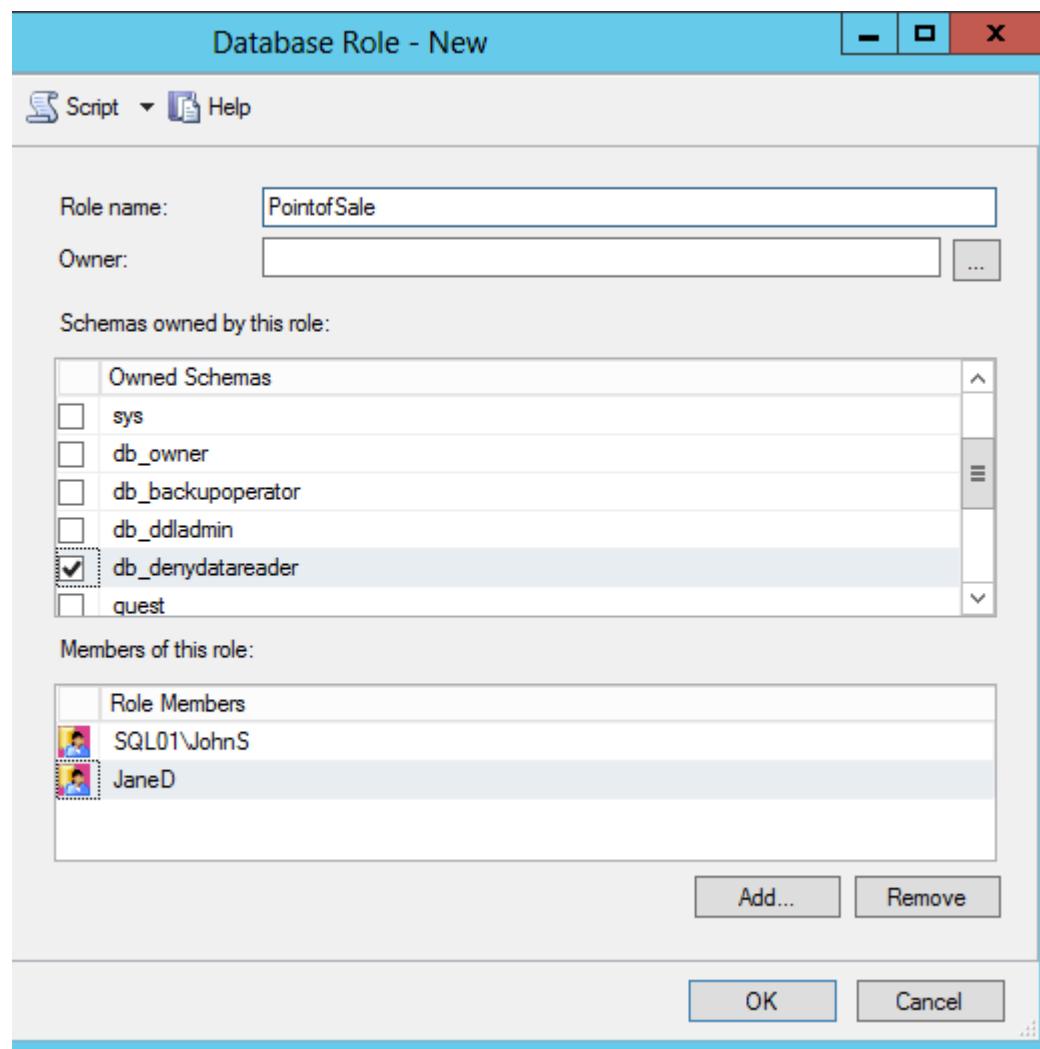


Figure 5-7: The Database Role dialog box is used to grant access to a specific user.

## Fixed Database Roles

*Fixed database roles* define the permissions that users within that role have on objects in the database. Similar to fixed server roles, there are a number of fixed database roles as well. The fixed database roles that are used in SQL are described in the table.

| Fixed Database Role | Users In This Role                                                                                                                             |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| db_owner            | Have complete permissions on all objects and activities within the database. These are similar to sysadmins but for the current database only. |
| db_securityadmin    | Can manage the security of the database by adding or modifying permissions, and logins. Extreme care must be taken to see who has this role.   |
| db_accessadmin      | Can add or remove access for Windows and SQL logins.                                                                                           |
| db_backupoperator   | Have permissions that are required to backup the database.                                                                                     |
| db_ddladmin         | Can run any Data Definition Language (DDL) statements.                                                                                         |
| db_datawriter       | Can add, modify, or delete data in any table of the database.                                                                                  |
| db_datareader       | Can view data in any table of the database.                                                                                                    |
| db_denydatawriter   | Cannot add, modify, or delete data in any table of the database.                                                                               |
| db_denydatareader   | Cannot view data in any table of the database.                                                                                                 |

## Database Permissions

*Database permissions* are the permissions that are given to a database role or a single user on the database. When you want to create a database role or wish to set it for a single user, you can specify the permissions that the role or user has from the available list. These permissions also depend on the type of object being secured.

For instance, all objects will have a CREATE or ALTER permission but only a table might have the INSERT or SELECT permission. Therefore, you can select different types of objects and give different sets of permissions for each in a single role.



Database Permissions

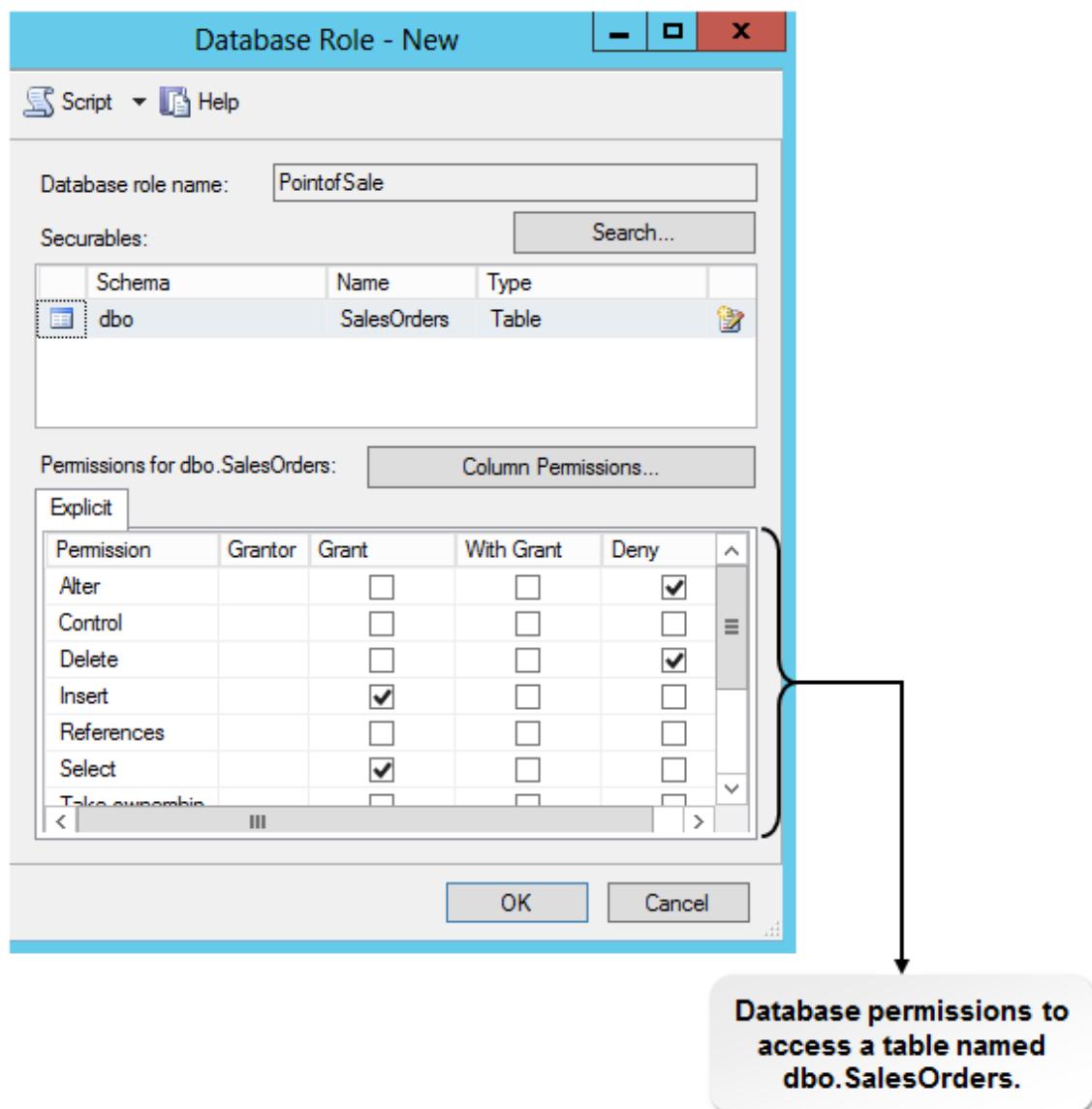


Figure 5–8: Grant and Deny permissions assigned to the PointofSale database role for a database named SalesOrders.

## Protection Against Object Modification



Protection Against  
Object Modification

While protecting data is important, ensuring that data structures are not modified by unauthorized personnel is equally important. Thus, appropriate steps must be taken to protect all objects from accidental or malicious modification.

For this, you could create a specific database role. Users who only need to work with the data are placed in one role and those who have permissions to change the structure of the object are in a different role. Only this role will have the permission to alter any database object.

Another way of protecting the database against object modification is by directly setting the ALTER permission on the objects you wish to protect from modification in its properties window as well.

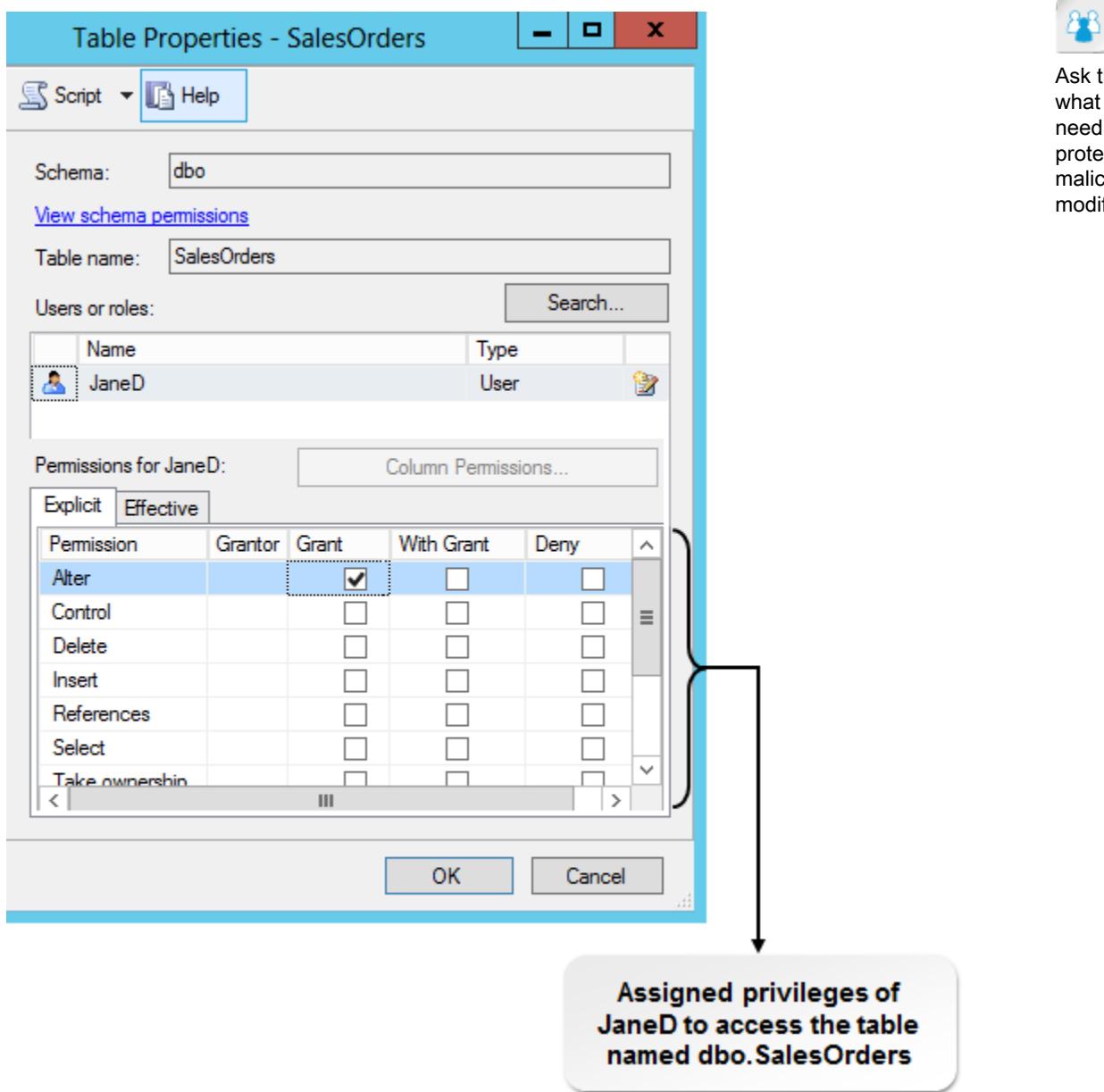


Figure 5-9: The Table Properties dialog box is used to assign permissions for the user login JaneD.

## Least Privilege Access

SQL Server applies the rule of *least privilege access* to determine the access of an individual login for each object.

Least privilege access is applied when a user is a member of multiple roles where each of them is having a number of permissions granted or denied to them.

For example, if User A is a member of roles R1 and R2. R1 has GRANT access to read from a number of tables and DENY access on some other tables. R2 similarly has GRANT on some and DENY on others. Some of these tables are common in R1 and R2 as shown in the following table.



Ask the students as to what are the steps that need to be taken to protect all object from malicious or accidental modifications.



Least Privilege Access  
(2 Slides)

| Table | Role R1 | Role R2 |
|-------|---------|---------|
| T1    | ✓       | ✓       |
| T2    | ✗       | ✓       |
| T3    | -       | ✓       |
| T4    | ✗       | ✗       |
| T5    | ✗       | -       |
| T6    | ✓       | ✗       |

Least privilege access assigned to two server roles to access the given tables.

Figure 5–10: Permissions assigned for Role 1 and Role 2.

In the table, the cross mark stands for DENY, the tick mark stands for a GRANT and “–” means nothing is specified.

SQL Server will apply the following set of permissions for a user who is part of both R1 and R2:

| Table | Resultant Permission of R1 and R2 |
|-------|-----------------------------------|
| T1    | ✓                                 |
| T2    | ✗                                 |
| T3    | ✓                                 |
| T4    | ✗                                 |
| T5    | ✗                                 |
| T6    | ✗                                 |

Combined least privilege access of two server roles.

Figure 5–11: The resultant permissions of Role 1 and Role 2.

The resultant permissions will be set only to table 1 and table 3 because the lowest permission set among the two or more roles is applied on each object.

## Migrating Logins



Migrating Logins

*Migrating logins* is the practice of moving a database from one server to another, along with the logins in the database. However, the accounts associated with those logins, either Windows or SQL, may not exist on the destination server or have a different *Security Identifier (SID)* associated with it. SID is used to identify users distinctively. In which case, these logins will be orphaned and the users will be unable to log in.

If you are moving from one server to another, this issue of orphaned logins will not happen for Active Directory accounts, but in all cases, these issues will happen for SQL Server accounts. To migrate logins properly, Microsoft provides a SQL script that you can run on the source server. The SQL script creates two system stored procedures on the server. By running the procedure, you will get an output of the logins on the system which you can then run on the destination server. The logins will then get created with the same password and SID as in the source server.

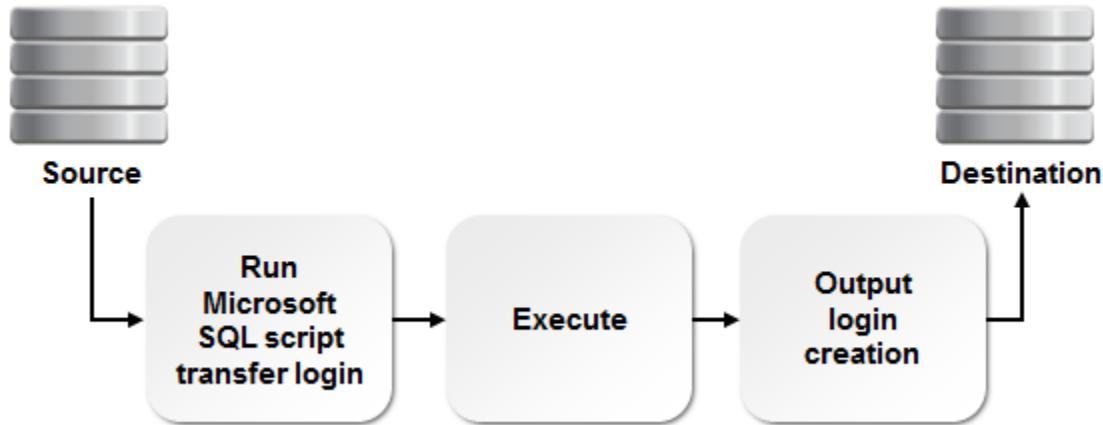


Figure 5–12: Migrating accounts within the source and destination server.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Create Roles and Grant Access to Server

# ACTIVITY 5–7

## Configuring Database Permissions and Security Roles

### Before You Begin

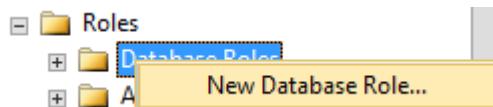
1. In the **Object Explorer** pane, select **Connect→Database Engine**.
2. In the **Connect to Server** dialog box, from the **Authentication** drop-down list, select **Windows Authentication** and select **Connect**.

### Scenario

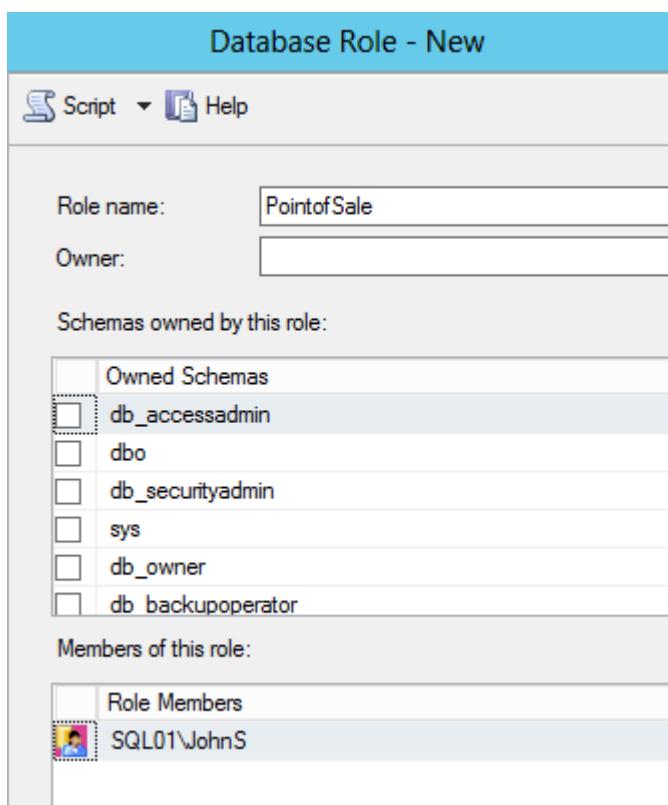
Management of Adventure Works Cycles has decided to provide permissions for users to be able to view and modify data within the sales database. You have to plan to use a new database role so that managing all the users within this group and granting them permissions to alter data is easier. You decide to grant permissions for a few users on the server role.

1. Create a new database role

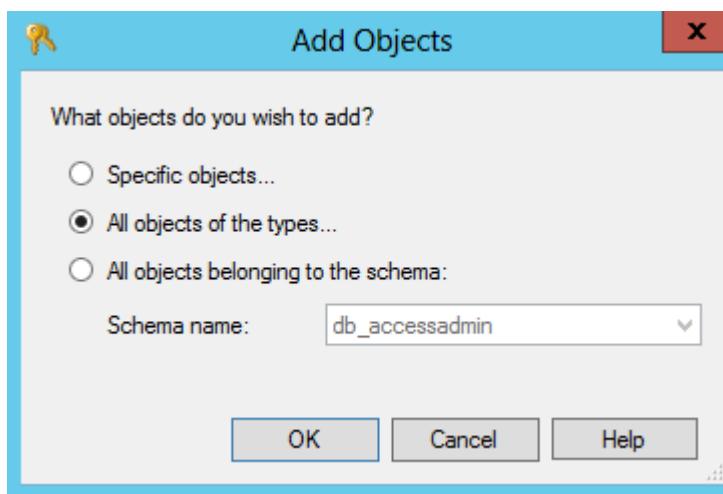
- a) In the **Object Explorer** pane, expand **Databases→SalesDB→Security→Roles**.
- b) Right-click **Roles** and from the fly out pane, select **New→New Database Role**.



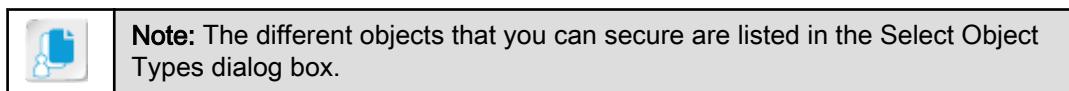
- c) In the **Database Role- New** dialog box, in the **Role name** text box, type the name of the role as **PointOfSale**
- d) In the **Members of this role** section, select the **Add** button to add the object name.
- e) In the **Select Database User or Role** dialog box, in the **Enter the object name to select** text area text box, type **JohnS** and select **Check Names**.
- f) In the **Multiple Object found** dialog box, check **SQL01\John** check box and select **OK**.
- g) In the **Select Database User or Role** dialog box, observe that the user **SQL01\JohnS** is displayed and select **OK**.
- h) In the Database Role-New window, in the **Members of this role** section, observe that the user **SQL01\JohnS** is added.

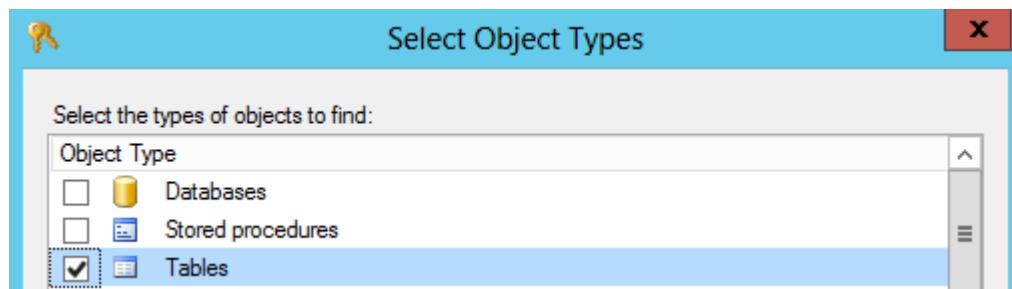


2. Grant permissions to the sales table.
  - a) In the **Select a page** section, select **Securables**.
  - b) In the Database Role-New window, select the **Search** button.
  - c) In the **Add Objects** dialog box, select the **All objects of the types** option and select **OK** to close the **Add Objects** dialog box.



- d) In the **Select Object Types** dialog box, check the **Tables** check box and select **OK** to save the changes.





- e) In the Database Role- New window, in the **Permissions for dbo.SalesOrders** section, set the following permissions and select **OK** to save the permissions given to the new role.
- Alter: Deny
  - Delete: Deny
  - Insert: Grant
  - Select: Grant
  - Update: Deny

| Permissions for dbo.SalesOrders: |  |                |         |                                     | Column Permissions...    |                                     |
|----------------------------------|--|----------------|---------|-------------------------------------|--------------------------|-------------------------------------|
| Explicit                         |  | Permission     | Grantor | Grant                               | With Grant               | Deny                                |
|                                  |  | Alter          |         | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
|                                  |  | Control        |         | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            |
|                                  |  | Delete         |         | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
|                                  |  | Insert         |         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            |
|                                  |  | References     |         | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            |
|                                  |  | Select         |         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            |
|                                  |  | Take ownership |         | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            |

# ACTIVITY 5–8

## Protecting Objects from Modification

### Scenario

The sales manager of Adventure Works Cycles wants to secure the sales transactions table from modifications to the structure that is created, because many “PointOfSale” applications use this table for storing each transaction. You are asked to assign only the appropriate users the permissions on the table to perform modification on the PoS applications.

1. Grant permissions to a particular user.
  - a) In the **Object Explorer** pane, expand **SalesDB->Tables**.
  - b) Right-click the **dbo.SalesOrders** table and select **Properties**.
  - c) In the **Table Properties - SalesOrders** dialog box, in the **Select a page** section, select **Permissions**.
 

 **Note:** The ‘PointOfSale’ role that you created earlier will also be displayed here.
  - d) In the **Table Properties** dialog box, select the **Search** button.
  - e) In the **Select Database Users or Roles** dialog box, in the **Enter the object name to select text area** text box, type **JaneD**, and then select the **Check Names** button. Select **OK** to add the login to the database.
  - f) Select the **JaneD** login and set the following permissions.
    - **Alter:** GRANT
    - **Control:** GRANT
    - **View Definition:** GRANT
  - g) Select **OK** to save the changes made.
2. Which database role is similar to sysadmin?
  - db\_owner
  - db\_accessadmin
  - db\_ddladmin
  - db\_securityadmin
3. Which database permission is set to protect objects from modification?
  - Alter
  - Delete
  - Control
  - Insert

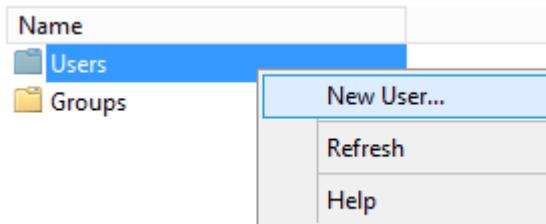
## ACTIVITY 5–9

### Setting Up Least Privilege Access to the Server and Database

#### Scenario

Management of the Adventure Works Cycles wants to check the accounts that are used by the SQL Server services with the least privileges and do not have access to the system. You decide to assign the least privileges on specific users within a server role.

1. Launch Server Manager and add a new user account.
  - a) In the desktop interface of Windows Server 2012, move the mouse pointer to the bottom left of the task bar and select **Server Manager** icon to launch it.
  - b) In the Server Manager window, select **Tools** and select **Computer Management**.
  - c) In the Computer Management window, expand **Local Users and Groups** and select the **Users** folder.
  - d) To add a new user to the server, right-click the **Users** folder and select **New User...**.

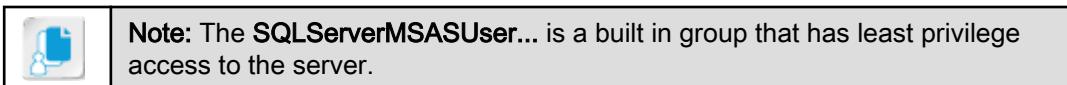


- e) In the **New User** dialog box, type the following credentials:
  - Username: **SteveR**
  - Full Name: **Steve Rickie**
  - Password/Confirm Password: **pass@word1**
- f) Uncheck the **User must change password at next logon** check box and check the **Password never expires** check box.

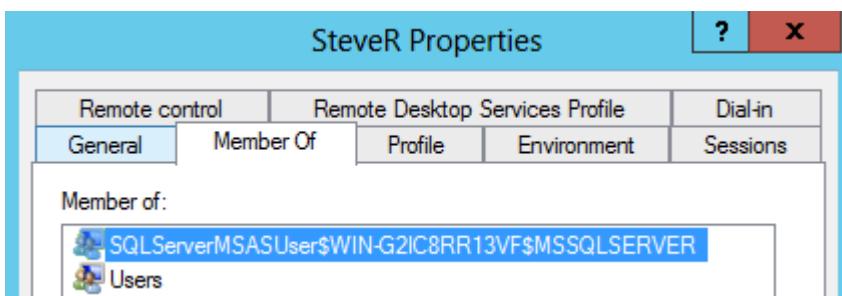


- g) On the **New User** dialog box, select **Create** to create the new Windows account, select **Close** to exit the dialog box.

2. Grant access to the database on the server.
  - a) In the Computer Management window, in the left pane, double-click **Groups**.
  - b) In the Computer Management window, from the right pane, copy the name of the **SQLServerMSASUser** group.
  - c) In the Computer Management window, double-click **SteveR**.
  - d) In the **SteveR Properties** dialog box, select the **Member Of** tab.
  - e) Select **Add** and in the **Select Groups** dialog box, in the **Enter the object name to select** text area text box, paste the name of the group copied and select the **Check Names** button to search for the name in the server.



- f) In the **Select Groups** dialog box, select **OK** to add the new user, **SteveR**, as a member of **SQLServerMSASUser\$WIN-G21C8RR13VF4MSSQLSERVER**.



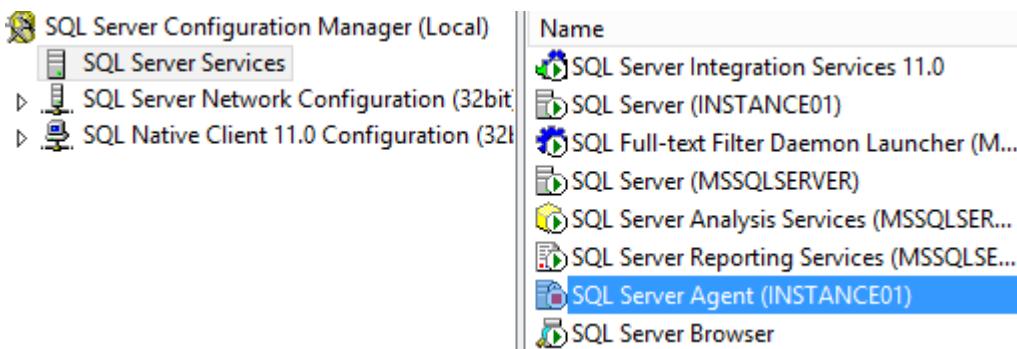
**Note:** The server names are specific to every domain. Please ensure you use the default server name.

- g) Select **OK** to add the new user **SteveR** to the pre-defined group.

| Computer Management |               |                                        |
|---------------------|---------------|----------------------------------------|
| Name                | Full Name     | Description                            |
|                     | Administrator | Built-in account for administering...  |
|                     | Guest         | Built-in account for guest access t... |
|                     | JohnS         | John Smith                             |
|                     | SteveR        | Steve Rickie                           |

- h) Close the Computer Management window.

3. Launch SQL Server Configuration Manager and set up least privilege access to a user on the server.
- In the desktop interface of Windows Server 2012, move your mouse pointer to the bottom-left corner of the screen and select the **Metro Interface** to view the **Start** screen.
  - In the **Start** screen, navigate to **SQL Server Configuration Manager**.
  - Select **SQL Server Configuration Manager** to launch it.
  - In the SQL Server Configuration Manager window, in the left pane, select **SQL Server Services** and in the right pane, double-click **SQL Server Agent (INSTANCE01)**.



- e) In the SQL Server Agent (Instance01) Properties window, change the account name to **SteveR** and password as **pass@word1**

- f) Select **Apply** to update the new changes and select **OK** to save the changes. The least privilege access has been set to the account name Steve R.
  - g) Close the SQL Server Configuration Manager window.
-

# Summary

In this lesson, you have configured security settings to both server and databases. Configuring the required security settings will enable you to assign permissions and revoke them whenever necessary, and give access to specific users. In addition, you have also created server roles to monitor permissions within a closed network.

## What are the best practices to be followed while setting up a SQL Server account on the Windows Server?

**A:** Answers will vary, but may include: SQL login accounts have access only to the instance in which they are created and are authenticated by the server, can enable either Windows login or both on the security settings, Windows user to login to SQL Server, an existing administrator will need to grant them permission to do so.



Use the review questions provided to generate discussion among the participants.

## What are the differences between user-defined and fixed server roles?

**A:** Answers will vary, but may include: user-defined roles are built-in roles that make it easier to grant or revoke access to a server at any point in time; they can be created and used in any place where a fixed server role can be used. Fixed server roles, are pre-defined roles that cannot be deleted or modified in any manner other than adding accounts to them.



**Note:** Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.



# 6

# Implementing Advanced Security Settings

**Lesson Time:** 2 hours, 30 minutes

## Lesson Objectives

In this lesson, you will:

- Configure certificate-based logins.
- Apply auditing and policy-based management strategies.

## Lesson Introduction

You have configured the basic security settings in SQL Server® and the databases within a server. However to secure logins and data in the server, advanced security settings, such as encryption, certificates, auditing, and policy management, should be implemented.

Implementing these advanced security settings will enable you to use digital certificates for logins and to monitor the changes made to your server and databases. Familiarizing yourself with the advanced security settings will enable you to protect and recover your server from security breaches. In this lesson, you will implement advanced security settings.

# TOPIC A

## Create Certificate-Based Logins

In most cases, SQL Server uses the standard Windows based logins or SQL Server based logins for authentication. However, when dealing with highly confidential data, it is important to take extra precautions because both types of logins can be impersonated. This is where certificate-based logins come into the picture. Certificate-based logins ensure that data is accessed only by an authorized user. A better understanding on creating certificate-based logins will enable you to encrypt and protect your data from unauthorized access. In this topic, you will create certificate-based logins.

### Certificates



A *certificate* is a digital identity that is assigned to objects such as a user, database, server, or service. A universally trusted source called the *Certification Authority* (CA) issues these certificates to the objects to prove their identity. A certificate from the CA proves the identity of the object to anyone who requires it.

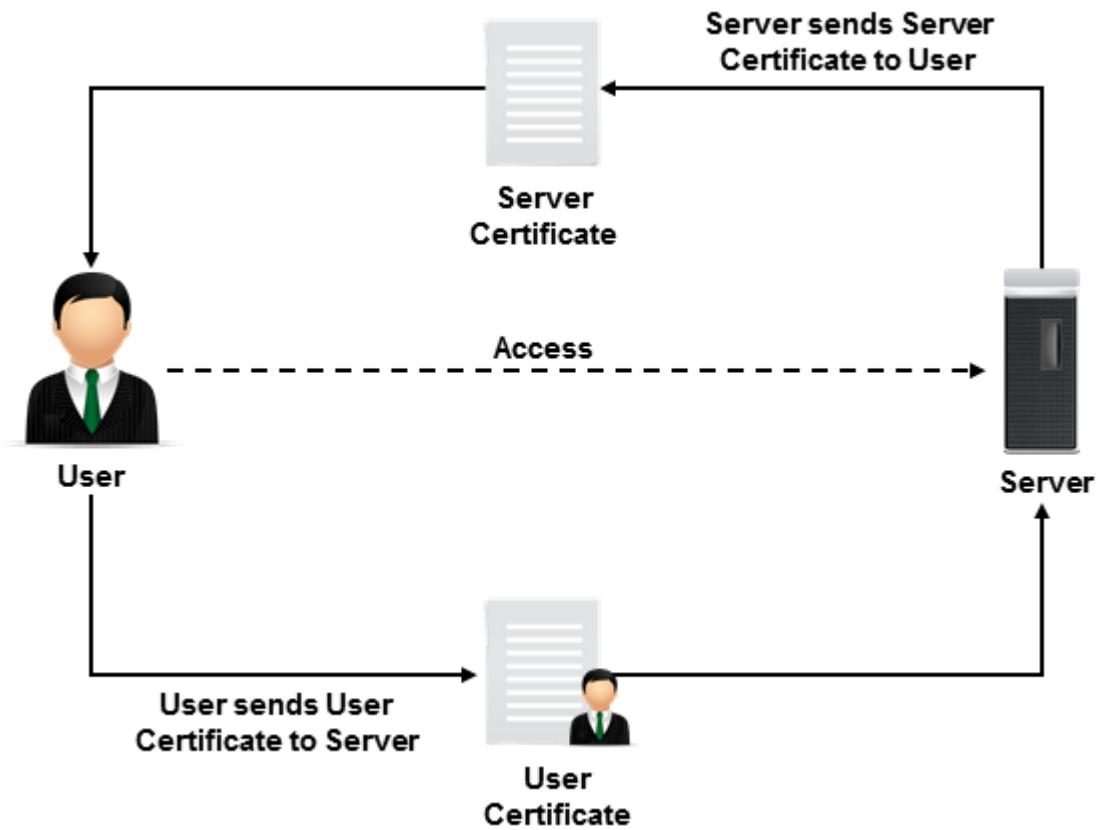


Figure 6-1: The user and server exchange certificates to authorize a user to access the server.

### Asymmetric Keys



*Asymmetric keys* are a pair of keys used in *asymmetric encryption* that uses different keys for encryption and decryption. Each certificate contains two keys: *Private Key* and *Public Key*. The private key is known only to the user of the certificate. The public key is shared with anyone who requests the

certificate. Both these keys are used for encrypting and decrypting the data communicated between the caller and requester.

At each end of the communication, the certificate from the other end is received and verified. The certificate sent to the other end contains only the public key. The private key is retained by the respective certificate owners. Once the verification is successful, any information that needs to be sent across the network to the other side is encrypted using the public key available in the certificate. The encrypted information can only be decrypted by the corresponding private key. So, even if someone gets hold of the public key and the encrypted message, he will not be allowed to decrypt it.

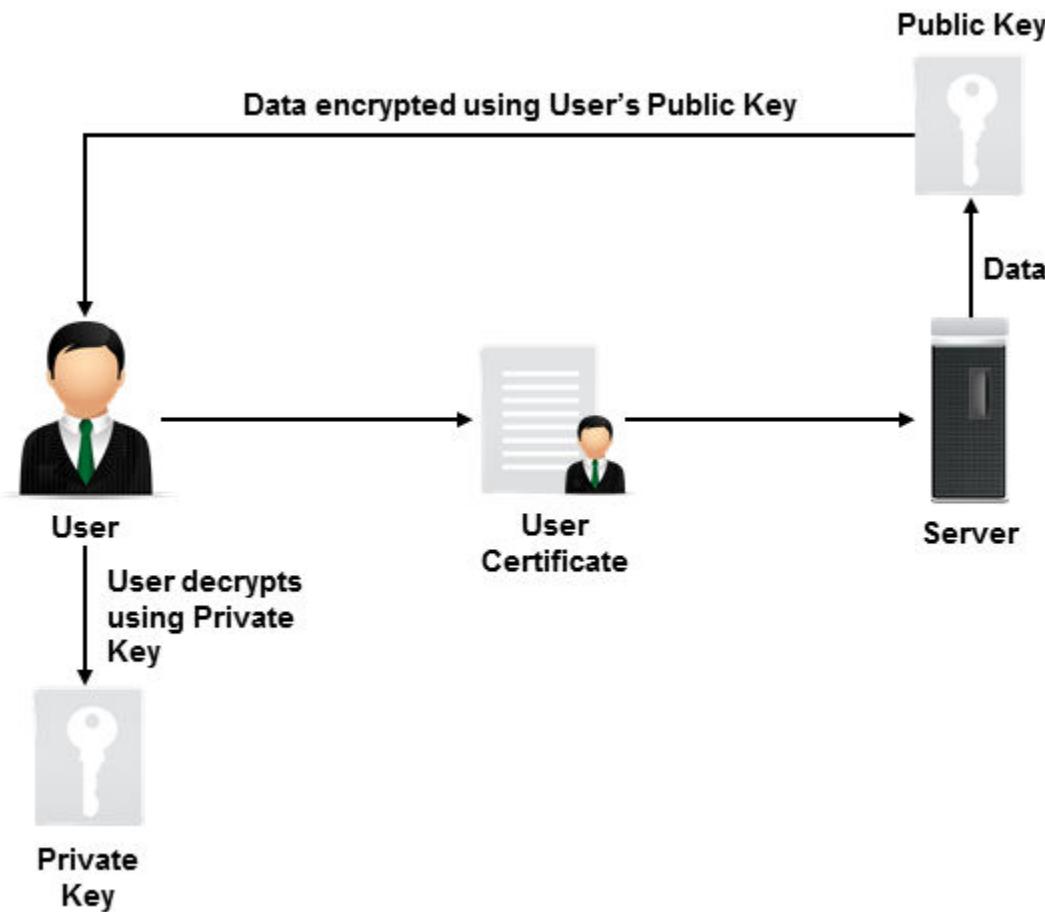


Figure 6-2: The encryption and decryption of data using asymmetric keys.

## Symmetric Keys

Symmetric keys are keys used for both encryption and decryption. SQL Server contains two types of symmetric keys, the *Service Master Key* (SMK) and *Database Master Key* (DMK).

| Symmetric Key | Description                                                                                                                                                                                                                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SMK           | A symmetric key that is auto generated when an SQL Server instance starts up for the first time. This is used to encrypt other keys such as the DMK, linked server passwords, and login credentials.                                                                                                           |
| DMK           | A symmetric key used to protect the asymmetric keys and private keys of certificates stored in the database. The DMK is not used to encrypt data. It is encrypted using the SMK and can only be decrypted by the service account of the server instance or a user principle with credential management rights. |

The syntax for creating a database master key is as follows.

```
USE <database>
 CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<password>'
```

### Example

The SQL statement to create a DMK in the master database is as follows.

```
USE MASTER;
 CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'pass@word1';
```

This statement will create a DMK for the master database and encrypts it using password "pass@word1."



**Note:** Ask the students to list out the advantages and disadvantages of using symmetric keys over asymmetric keys.



### Certificate Logins

*Certificate login* is a login associated with a certificate to ensure that the user is a valid user and any information exchange over the network is encrypted and prevents man-in-the-middle type of attacks where false information is included without the knowledge of users.

To create certificate logins, you need to add a certificate to the SQL Server instance by using one of the following ways.

- Obtaining a digital certificate from a trusted third-party Certification Authority (CA) and importing it to the server.
- Using the "makecert" command line tool supplied as part of SQL Server installation to create a certificate file and importing it to the server.
- Using T-SQL statements to create a self-signed certificate.

The syntax to create a certificate is as follows.

```
CREATE CERTIFICATE <CertificateName>
 WITH SUBJECT= '<Description>',
 EXPIRY_DATE = '<Date>';
 ENCRYPTION BY PASSWORD = '<password>'
```

The certificate created is automatically encrypted by the current DMK. If the DMK has not been created, you can create one or use the ENCRYPTION BY PASSWORD option to specify a password.



**Note:** It is recommended that you create the DMK first and then create the certificate so that the DMK automatically encrypts the certificate.

### Example

A CREATE CERTIFICATE statement to create a certificate encrypted by the database master key is as follows.

```
CREATE CERTIFICATE MyNewCertificate
 WITH SUBJECT='Certificate for user1',
 EXPIRY_DATE='01/01/2025';
```

This statement will create a certificate named MyNewCertificate with description as a certificate for user1 and expiry date 01/01/2025. The certificate created is encrypted by the current DMK.

You can create a certificate-based login by creating a login and mapping it to the certificate. The syntax to create a certificate-based login is as follows.

```
CREATE LOGIN <username> FROM CERTIFICATE <CertificateName>;
```

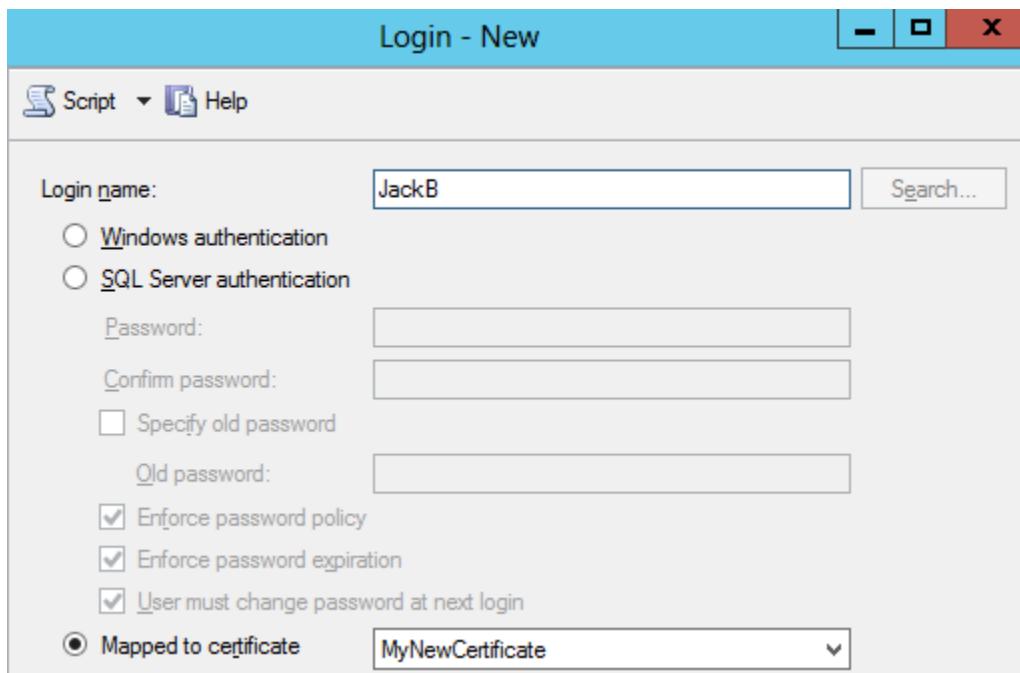
## Example

A CREATE LOGIN statement to create a certificate-based login is as follows.

```
CREATE LOGIN <JackB> FROM CERTIFICATE <MyNewCertificate>;
```

This statement will create a login JackB mapped to a certificate MyNewCertificate.

You can also create a certificate-based login from SSMS. For this, you need to expand **Server→Security→Logins→New Login**.



**Figure 6-3: The Login-New window for creating certificate-based logins.**

Once the login is created, you can log in to SQL Server using the same certificate.

## Exporting Certificates

You can also export the certificate from SQL and import it to another machine. The syntax for exporting a certificate is as follows.

```
BACKUP CERTIFICATE <certificate name>
 TO FILE='<filepath including the name of the certificate file>
 WITH PRIVATE KEY (
 FILE='<filepath including the name of the key>',
 ENCRYPTION BY PASSWORD='<password>';
```

## Example

The BACKUP CERTIFICATE statement to export a certificate with a private key to a file is as follows.

```
BACKUP CERTIFICATE MyNewCertificate
 TO FILE='c:\Temp\JackBCert.cer'
 WITH PRIVATE KEY (
 FILE='c:\Temp\JackBPvtKey.key',
 ENCRYPTION BY PASSWORD='NewSecretPass@word123');
```

This statement will back up the certificate as a file named JackBCert.cer and the private key encrypted with password NewSecretPass@word123 as a file named JackBPvtKey.key in the location c:\Temp.



Ask the students to list scenarios where they would prefer using a certificate-based login rather than a Windows login in SQL Server.



You may want to show LearnTO Create Certificate-Based Logins from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



**Note:** For more information about certificate-based logins, check out the LearnTO **Create Certificate-Based Logins** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.

## Endpoints

*Endpoints* are objects that enable you to connect to SQL Server using protocols such as Simple Object Access Protocol (SOAP), XML Web services, and Windows Communication Foundation (WCF).

In SQL Server 2012, endpoints are a deprecated feature. It is recommended that you do not use endpoints on the server or in applications accessing the server. If you have migrated a database from an older version of SQL Server, which has endpoints configured on it, you need to change the endpoints with the help of a qualified architect.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Manage Certificates and Certificate Logins

# ACTIVITY 6–1

## Managing Certificates and Keys

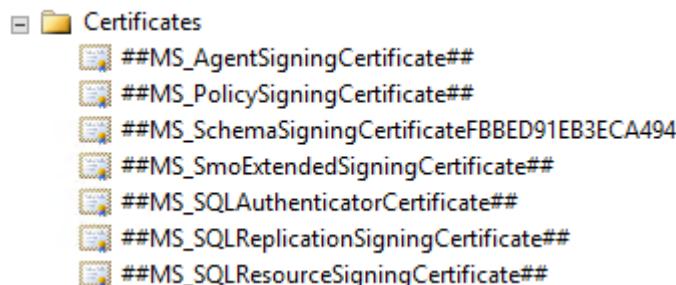
### Scenario

Management of Adventure Works Cycles wants a certificate for the master database so that it can send encrypted data to users who have certificates. You are asked to create a digital certificate encrypted with a DMK of the master database. You decide to first create a DMK for the master database and then create a certificate so that the certificate is automatically encrypted by the DMK.

1. View the default certificates, symmetric keys, and asymmetric keys available in the server.
  - a) In SSMS, in the **Object Explorer** pane, expand **SQL01→Databases**.
  - b) Expand **System Databases→master→Security→Certificates** to view the certificates present in the master data base.
  - c) Observe the list of system generated certificates starting with a double hash (##) that are available in the server.



**Note:** It is recommended that you do not modify or alter the system generated certificates.



- d) Expand the **Asymmetric Keys** and **Symmetric Keys** folder and observe the asymmetric and symmetric keys available in the server.
2. Execute the query to create a DMK to encrypt the private keys of the certificates in the master database.
  - a) Open a **Query Editor** pane, enter the USE keyword followed by the database name MASTER database to create a DMK.

```
USE MASTER;
```

  - b) Enter a CREATE MASTER KEY statement along with the ENCRYPTION BY PASSWORD keyword followed by password “pass@word1” in single quotes to encrypt the DMK.

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'pass@word1';
```

  - c) Execute the CREATE MASTER KEY statement to create a master key.
  - d) On the **Messages** tab, observe that the message “Command(s) completed successfully” is displayed.- 3. Execute the query to create a new certificate.
  - a) Open a **Query Editor** pane, enter a CREATE CERTIFICATE statement followed by the certificate name “MyNewCertificate.”

```
CREATE CERTIFICATE MyNewCertificate
```

- b) Enter the WITH SUBJECT clause followed by the subject of the certificate as "Certificate for JackB."

```
CREATE CERTIFICATE MyNewCertificate
WITH SUBJECT='Certificate for JackB',
```

- c) Enter the EXPIRY\_DATE keyword followed by the date "01/01/2025" for the certificate.

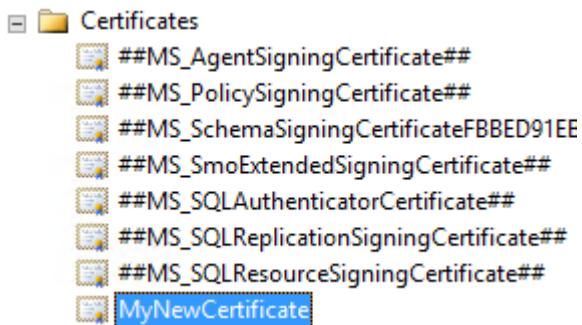
```
CREATE CERTIFICATE MyNewCertificate
WITH SUBJECT='Certificate for JackB',
EXPIRY_DATE='01/01/2025';
```

- d) Execute the CREATE CERTIFICATE statement to create a certificate for JACKB.

- e) On the **Messages** tab, observe that the message "Command(s) completed successfully" is displayed.

4. Verify that the new certificate is created.

- a) In the **Object Explorer** pane, right-click **Certificates** folder and select **Refresh** to verify that the "MyNewCertificate" certificate is added in the list.



- b) Close all the **Query Editor** panes without saving the queries.

# ACTIVITY 6–2

## Creating a Certificate-Based Login

### Scenario

The Management of Adventure Works Cycles wants to ensure that all the data transferred to the users of the server are encrypted. You are asked to create a certificate-based login for the users. You decide to create a certificate-based login by creating a login mapped to the certificate you just created.

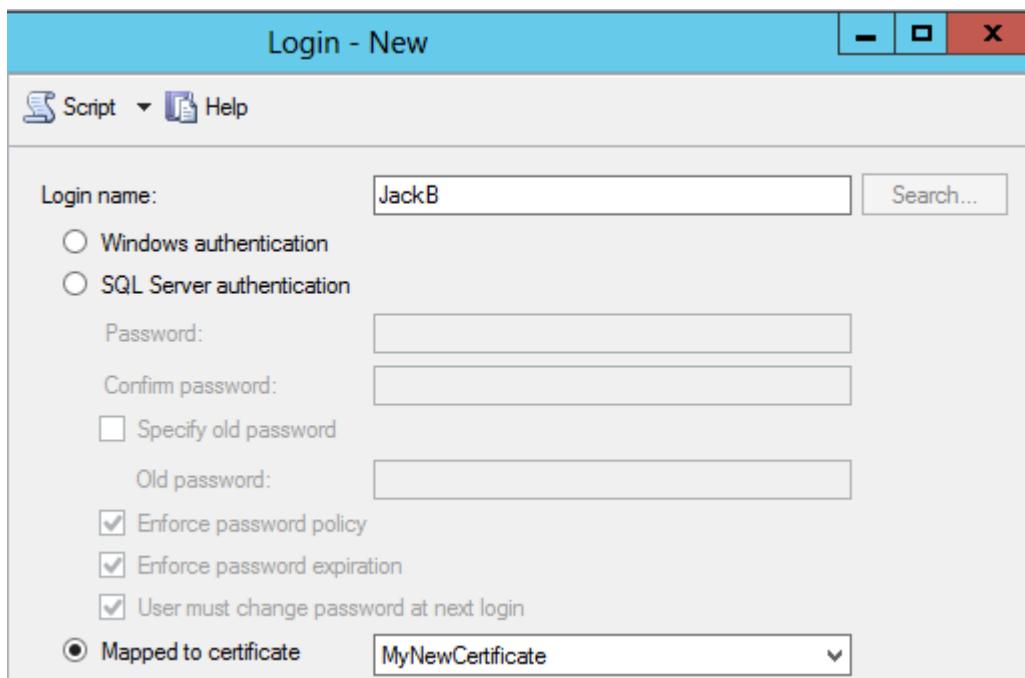
1. Create a new login mapped to the certificate MyNewCertificate.

- a) In **Object Explorer** pane, right-click **SQL01** and select **Refresh**.



**Note:** The name of the server can change from system to system.

- b) Expand **SQL01→Security**.
- c) Right-click **Logins** and select **New Login**.
- d) In the **Login-New** window, in the **Login name** text box, type **JackB**.
- e) Select the **Mapped to certificate** option, from the drop-down list, select **MyNewCertificate**.



Remind participants that the certificate MyNewCertificate is self-generated by the SQL Server and is not obtained from the trusted third-party CA.

- f) In the **Select a page** section, select **Server Roles** page, check the **SalesDBAdmins** check box and select **OK** to create the login.

2. Export the certificate from the master database to ensure that the user can use the certificate.

- a) Open Windows Explorer and check if the C:\Temp folder exists; if not, create a folder named Temp in the C: drive.
- b) In SSMS, open a **Query Editor** pane, enter a BACKUP CERTIFICATE statement followed by the certificate name MyNewCertificate.

```
BACKUP CERTIFICATE MyNewCertificate
```

- c) Enter the TO FILE clause followed by the file path where the certificate is to be exported including the name of the certificate file JackBCert.cer.

```
BACKUP CERTIFICATE MyNewCertificate
TO FILE='c:\Temp\JackBCert.cer'
```

- d) Enter the WITH PRIVATE KEY clause followed by the FILE keyword and the file path where the private key is to be stored including the name of key file JackBPvtKey.key.

```
BACKUP CERTIFICATE MyNewCertificate
TO FILE='c:\Temp\JackBCert.cer'
WITH PRIVATE KEY (
FILE='c:\Temp\JackBPvtKey.key',
```

- e) Enter the ENCRYPTION BY PASSWORD clause followed by the encryption password NewSecretPass@word123.

```
BACKUP CERTIFICATE MyNewCertificate
TO FILE='c:\Temp\JackBCert.cer'
WITH PRIVATE KEY (
FILE='c:\Temp\JackBPvtKey.key',
ENCRYPTION BY PASSWORD='NewSecretPass@word123');
```

- f) Execute the BACKUP CERTIFICATE statement to backup the certificate.

- g) On the **Messages** tab, observe that the message "Command(s) completed successfully" is displayed.

- h) Close the **Query Editor** pane without saving any changes.

**3. Verify that the certificate and key have been created in the folder specified.**

- a) In Windows Explorer navigate to C:\Temp and verify that the certificate and the key have been created.

| Name            | Date modified      | Type                 | Size |
|-----------------|--------------------|----------------------|------|
| JackBCert.cer   | 1/22/2013 12:21 PM | Security Certificate | 1 KB |
| JackBPvtKey.key | 1/22/2013 12:21 PM | KEY File             | 1 KB |

- b) Close Windows Explorer.



**Note:** In production scenarios, you can copy these files in other systems and use the certificate and private key by importing them to other systems.

# TOPIC B

## Apply Auditing and Policy-Based Management Strategies

You have created certificate-based logins to ensure that only a certified user accesses the server. However to track any unauthorized access or violation of permissions, specific events on the server and the databases must be monitored and logged. Auditing and policy management enable administrators to trace the events that caused violations on the server. A better understanding on applying auditing and policy-based management to the server will enable you to maintain audits and configure policies and thus ensure security. In this topic, you will apply auditing and policy-based management strategies.

### Server Audit

*Server audit* is a top level object that enables you to perform auditing at server instances and at database levels. It includes audit specifications that contain a number of events to monitor. The specifications can also monitor and track different server level conditions. The types of audit specifications are as follows.



The Create Audit Window

| Audit Specification          | Description                                                              |
|------------------------------|--------------------------------------------------------------------------|
| Server audit specification   | Handles auditing for events that occur at the SQL Server instance level. |
| Database audit specification | Handles auditing for events that occur at the database level.            |

The audit object also specifies the location for the audit information to be stored. The audit information is stored in a Windows Event Log, a database, or a file. You can create a server audit using T-SQL statements.

The syntax to create a server audit that stores the audit information in a file is as follows.

```
CREATE SERVER AUDIT <AuditName>
TO FILE (FILEPATH= '<Path to store audits>');
```

This code snippet creates a server audit with the name specified in the AuditName parameter. The server audit stores audit information in a file in the file path specified in the path to store audits parameter.

### Example

A CREATE SERVER AUDIT statement to create a server audit that stores audit information in a file is as follows.

```
CREATE SERVER AUDIT <MyServerAuditSPECIFICATION <SAS_Name>
TO FILE (FILEPATH= 'C:\Temp');
```

This statement creates a server audit named MyServerAudit that stores audit information in the Temp folder in C:\ drive.

The syntax to create a server audit that stores audit information in an application log is as follows.

```
CREATE SERVER AUDIT <name> TO APPLICATION_LOG;
```

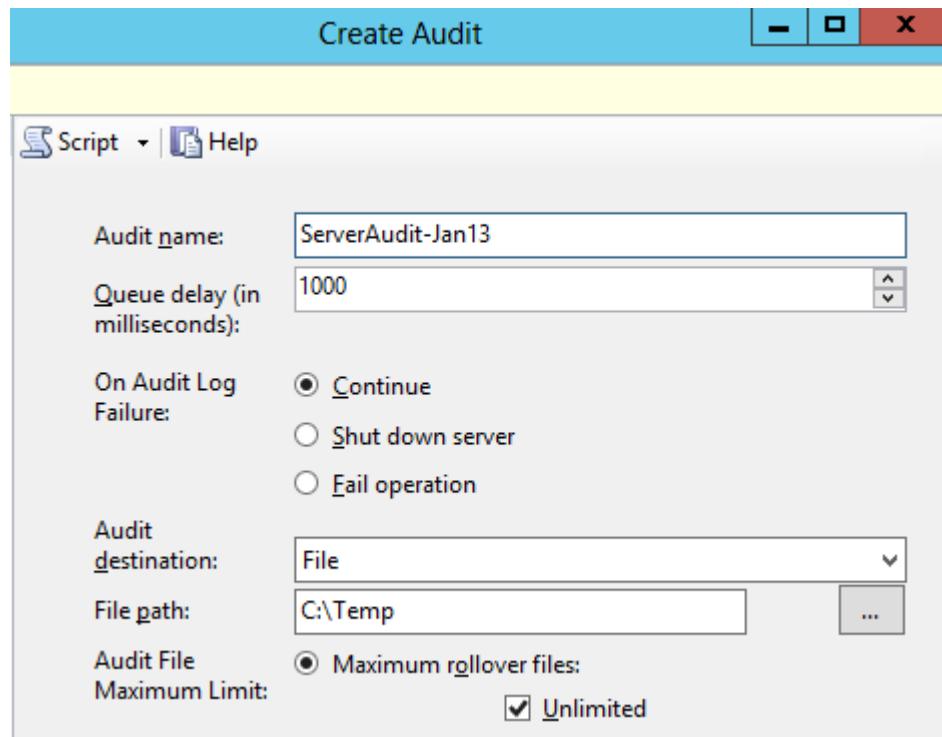
### Example

The CREATE SERVER AUDIT statement to create a server audit and store it in the Windows Application Log is as follows.

```
CREATE SERVER AUDIT Audit_Application_Log TO APPLICATION_LOG;
```

This statement will create a server audit with name Audit\_Application\_Log that stores the audit information in the Windows Application Log.

You can also create server audits from SSMS by expanding **Server→Security→Audits→New Audit**. You can specify the audit name, the destination to store audit information in the Create Audit window. The **On Audit Log Failure** option in the Create Audit window allows you to specify what action needs to be performed if audits cannot be logged.

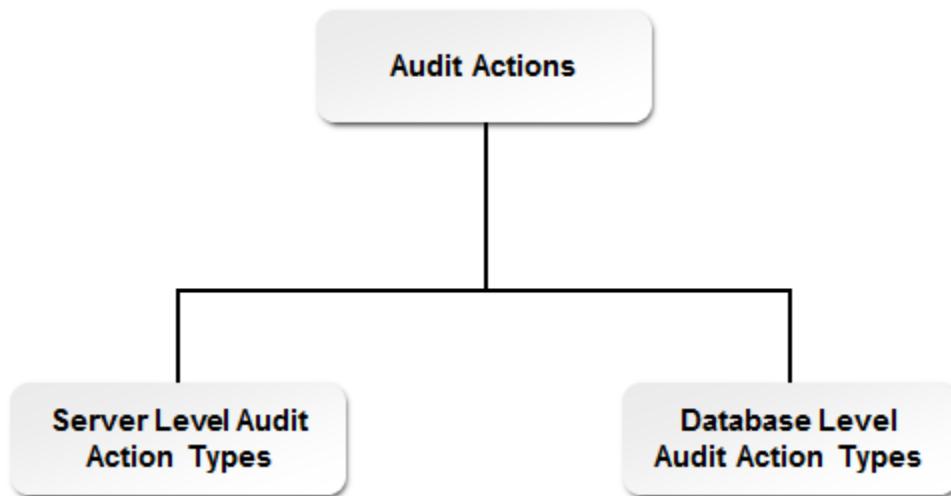


*Figure 6–4: The Create Audit window to create a new server audit.*

## Audit Actions



*Audit actions* are the actions for which the audit information is logged. Audit actions can be classified into server level and database level actions. The audit actions are grouped into pre-determined audit groups. You can audit all events in a group by specifying the group in an audit specification.



**Figure 6-5: Types of audit actions.**

## Server Level Audit Action Types

The server level audit groups include events for auditing actions related to the server.

| <b>Server Level Audit Action Type</b> | <b>Used To</b>                                                       |
|---------------------------------------|----------------------------------------------------------------------|
| SUCCESSFUL_LOGIN_GROUP                | Audit the successful logins in the server instance.                  |
| FAILED_LOGIN_GROUP                    | Audit the failed logins.                                             |
| SERVER_ROLE_MEMBER_CHANGE             | Audit the changes made to server roles.                              |
| SERVER_PRINCIPAL_IMPERSONATION_GROUP  | Audit the changes made to principals using the EXECUTE AS statement. |
| SERVER_PRINCIPAL_CHANGE_GROUP         | Audit the changes made to principals.                                |

## Database Level Audit Action Types

The database level audit groups include events for auditing database actions.

| <b>Database Level Audit Action Type</b> | <b>Used To</b>                                                                                       |
|-----------------------------------------|------------------------------------------------------------------------------------------------------|
| DATABASE_CHANGE_GROUP                   | Audit the use of CREATE, ALTER, and DROP statements in the database.                                 |
| DATABASE_OBJECT_CHANGE_GROUP            | Audit the use of CREATE, ALTER, and DROP statements on the database objects such as tables, columns. |
| DATABASE_ROLE_MEMBER_CHANGE_GROUP       | Audit the addition and removal of login in a database role.                                          |
| DATABASE_PERMISSION_CHANGE_GROUP        | Audit the GRANT, REVOKE, or DENY statements used to change database permissions.                     |
| DATABASE_OBJECT_PERMISSION_CHANGE_GROUP | Audit the GRANT, REVOKE, or DENY statements used on database objects.                                |

| <b>Database Level Audit Action Type</b> | <b>Used To</b>                                                                                |
|-----------------------------------------|-----------------------------------------------------------------------------------------------|
| DATABASE_PRINCIPAL_CHANGE_GROUP         | Audit the principal changes such as creating, altering and dropping users.                    |
| SCHEMA_OBJECT_CHANGE_GROUP              | Audit the CREATE, ALTER, and DROP statements used on the schema objects.                      |
| DATABASE_OPERATION_GROUP                | Audit database operations such as checkpoint or subscription operations.                      |
| DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP  | Audit change of ownership of the database objects.                                            |
| DATABASE_OWNERSHIP_CHANGE_GROUP         | Audit the ALTER AUTHORIZATION statement that is used to change the ownership of the database. |

## Server Audit Specification



Server Audit Specification

The *server audit specification* object audits events at the server instance level. You can have only one server audit specification object per audit object.

Once an audit is created, you can add a server audit specification to it using T-SQL statements. Both the audit and server audit specification objects are in the disabled state and need to be enabled to audit server actions.

The syntax for creating a server audit specification is as follows.

```
CREATE SERVER AUDIT SPECIFICATION <SAS_Name>
FOR SERVER AUDIT <AuditName>
ADD (<List of Audit groups>);
```

The syntax for enabling the server audit and server audit specification is as follows.

```
ALTER SERVER AUDIT <AuditName>
WITH (STATE=ON);
ALTER SERVER AUDIT SPECIFICATION <SAS_Name>
WITH (STATE=ON);
```

### Example

The CREATE SERVER AUDIT SPECIFICATION statement to create a server audit specification for a server audit is as follows.

```
CREATE SERVER AUDIT SPECIFICATION MyServerAuditSpecification
FOR SERVER AUDIT MyServerAudit
ADD (FAILED LOGIN GROUP);
```

This code snippet will create a server audit specification named MyServerAuditSpecification for the server audit MyServerAudit. This server audit specification audits the actions in the FAILED LOGIN GROUP such as failed logins.

The ALTER SERVER AUDIT statements to enable a server audit and server audit specification are as follows.

```
ALTER SERVER AUDIT MyServerAudit
WITH (STATE=ON);
ALTER SERVER AUDIT SPECIFICATION MyServerAuditSpecification
WITH (STATE=ON);
```

This code snippet will enable the server audit MyServerAudit and the server audit specification MyServerAuditSpecification by setting their STATE to ON.

You can also create and add server audit specification to an Audit object from SSMS by expanding **Server→Security→Server Audit Specification→New Server Audit Specification**. In the Create Server Audit Specification window, you can specify the server audit specification name, the server audit to which the server audit specification is configured, and audit action types to be audited.

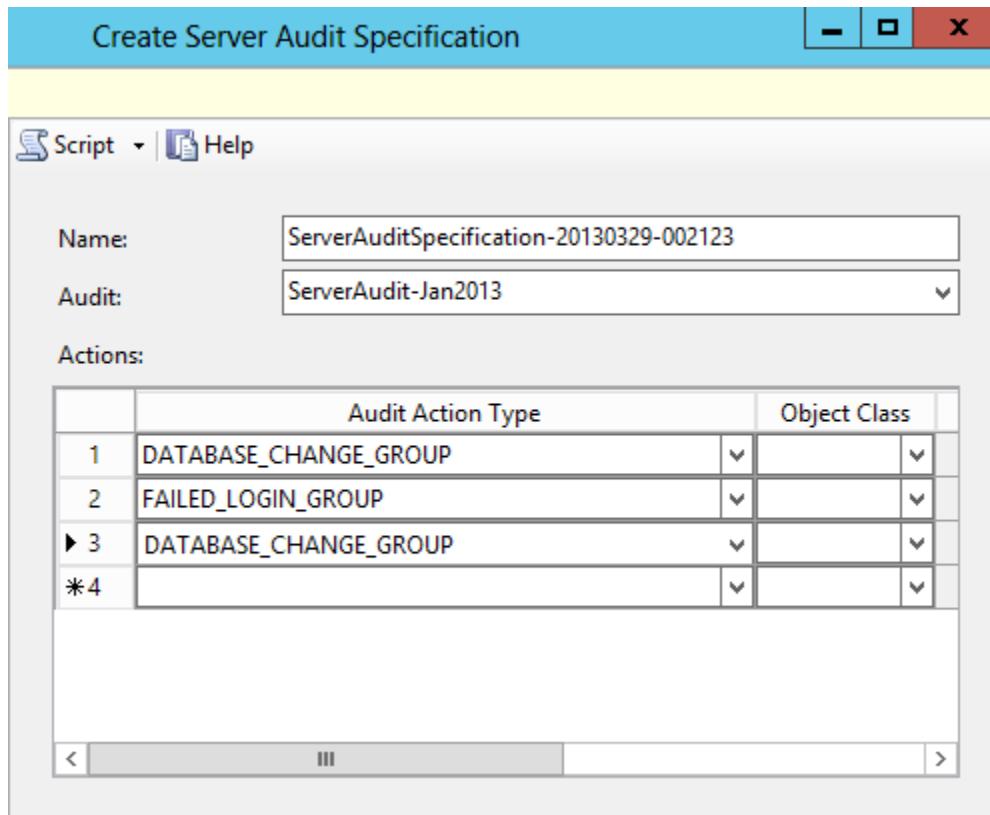


Figure 6–6: A server audit specification is created from the Create Server Audit Specification window in SSMS.

## Database Audit Specification

The *database audit specification* object tracks different parameters that work with a specific database and manages events by logging them based on the server audit to which it is configured.

The database audit specification manages database related events such as changes to logins, structure, and access permissions. It also allows you to audit data retrieval and modification by tracking usage of T-SQL commands such as the SELECT, INSERT, UPDATE, DELETE, and EXECUTE statements. The commands are specified as audit types of the database audit specification.

The syntax for creating a database audit specification is as follows.

```
CREATE DATABASE AUDIT SPECIFICATION <Das_Name>
FOR SERVER AUDIT <AuditName>
ADD (<List of Audit groups>);
```



Database Audit Specification

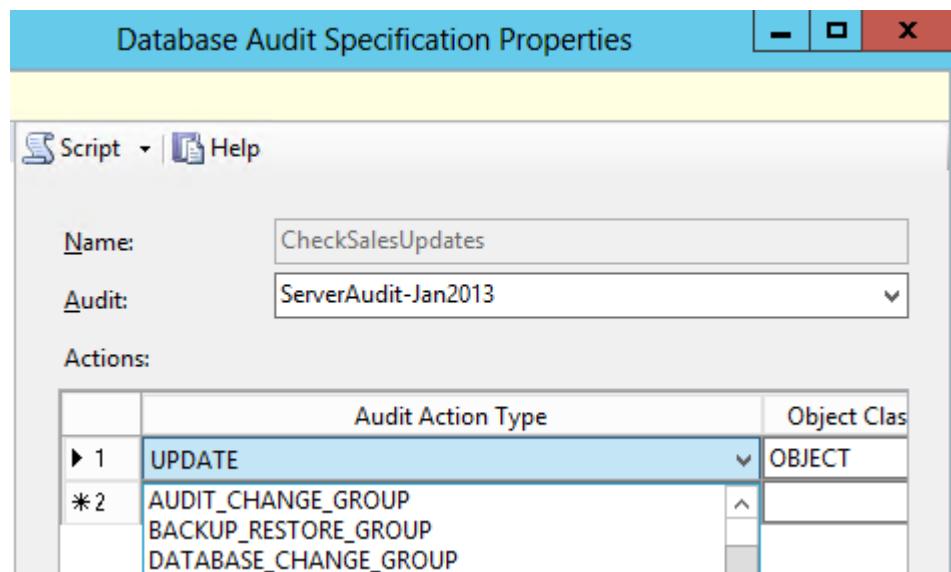
### Example

The CREATE DATABASE AUDIT SPECIFICATION statement to create a database audit specification for a server audit is as follows.

```
CREATE DATABASE AUDIT SPECIFICATION MyDatabaseAuditSpecification
FOR SERVER AUDIT MyServerAudit
ADD DATABASE_CHANGE_GROUP;
```

This statement will create a database audit specification named MyDatabaseAuditSpecification for the server audit MyServerAudit to audit the actions in the DATABASE\_CHANGE\_GROUP such as the CREATE statement performed on a database.

You can also create and add database audit specification to an audit object from SSMS by expanding **Server→Security→Database Audit Specification→New Database Audit Specification**. In the Create Database Audit Specification Properties window, you can specify the database audit specification name, the server audit to which it is configured and audit action types to be audited.



*Figure 6-7: The Audit Action Type specified in a Database Audit Specification Properties window.*

## PBM

*Policy Based Management (PBM)* is a feature of SQL Server that enables administrators to ensure that the different component configurations follow the standards prescribed by their organization. These standards can be implemented as a policy and can span settings for servers, instances, databases, and objects.

Examples for policies are as follows.

- Turning on auditing for failed logins.
- At database level, all user defined stored procedures should start with a prefix usp\_.

By creating a policy once, all SQL Servers in the organization can follow the same set of rules. Once the policies are created, it is easy to export and import them to other SQL Server instances in the organization network.



Ask the students under which circumstances they would prefer an audit over a policy and vice versa.



Facets (2 Slides)

## Facets

*Facets* are objects that contain all the properties that govern a policy. To configure a policy, you need to choose the required facets for the policy. A facet called “Database Options” contains properties to which a policy can be applied. You also need to define a “Condition” that defines the state of the chosen facet and its properties which you want to rule. When creating the policy, the condition for the facet, and optionally the conditions for filtering the facets, have to be specified.

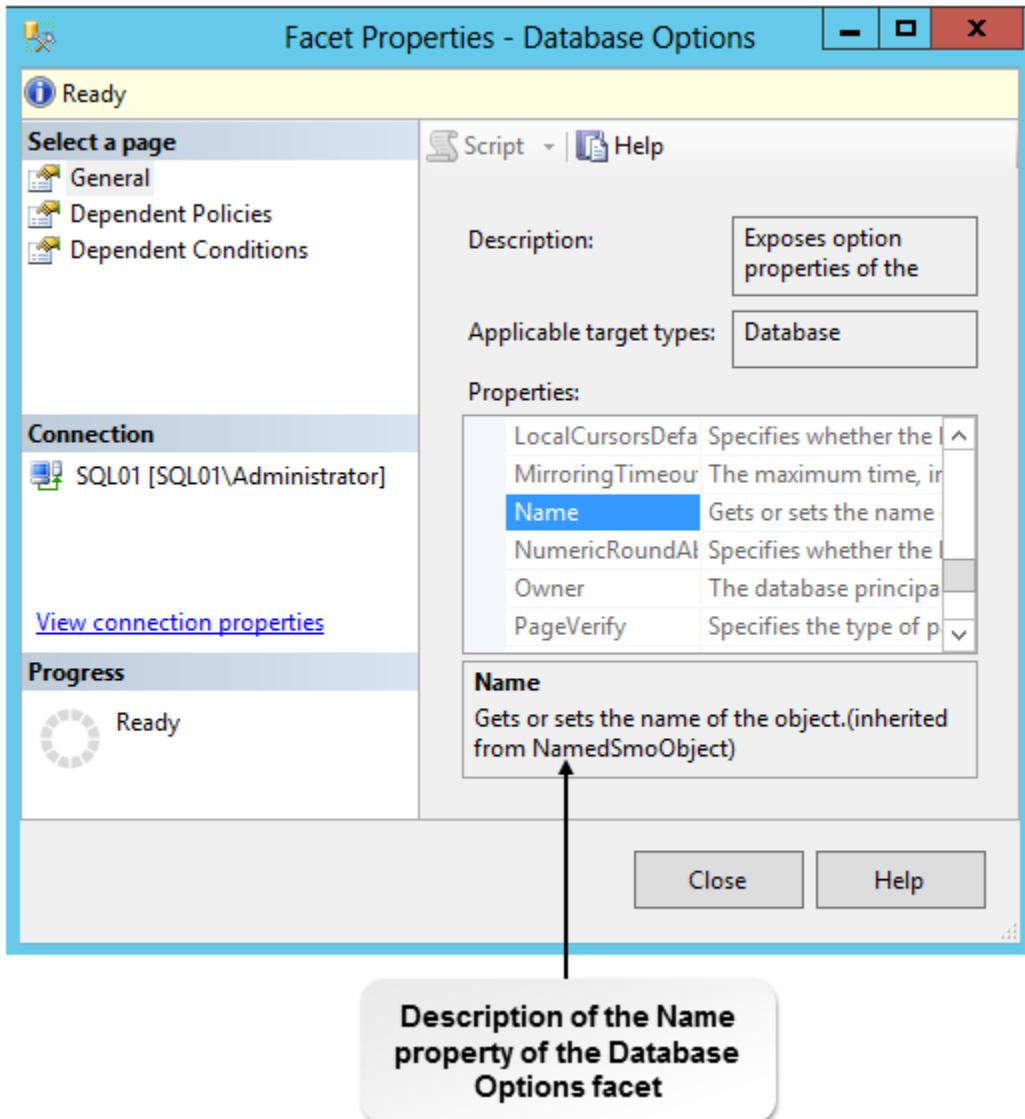
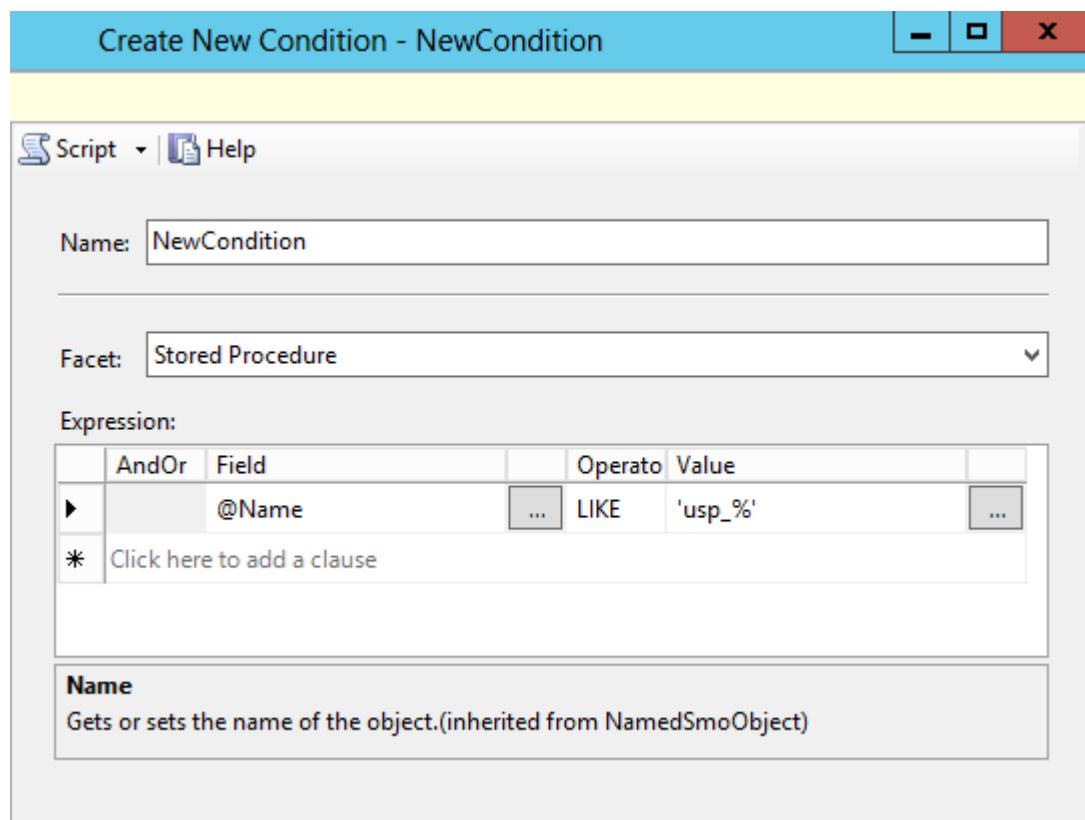


Figure 6-8: The Facet Properties window displaying the properties of the Database Options facet.



**Figure 6-9:** The condition for prefixing the stored procedures is specified in the Create New Condition window.

## Evaluation Modes

Evaluation modes determine how the policies you create should be evaluated. The four evaluation modes are as described.

| Evaluation Mode            | Description                                                                                                 |
|----------------------------|-------------------------------------------------------------------------------------------------------------|
| <b>On Demand</b>           | This is a manual mode where the admin/user triggers evaluating the policy.                                  |
| <b>On Change: Prevent</b>  | This is an automatic mode that prevents changes from being made that conflict with the policy               |
| <b>On Change: Log only</b> | This is an automatic mode in which the change is not prevented, but is logged.                              |
| <b>On Schedule</b>         | This is an evaluation mode in which the SQL Server Agent creates a job to periodically evaluate the policy. |

## Alerts

*Alerts* are messages sent to users for specific events that occur on the server. When using any one of the automated policy management evaluation modes (on change—prevent, on change—log only, or on schedule), the system makes an entry in the system's event log.

The entry made in the event log contains a Message Number field. This number specifies the corresponding event for the entry.

- 34050 denotes that the policy failed in the “On Change: Prevent” evaluation mode

- 34052 denotes that the policy failed in the “On Schedule” evaluation mode
- 34053 denotes that the policy failed in the “On Change: Log only” evaluation mode

This entry can be read by the SQL Server system and an email alert can be sent to an administrator using SQL Server Agent.

You can create an SQL Server Agent alert, based on an entry made by the PBM engine for a policy. The alert is created using the message number of the entry. You can also add optional filter on the alert to display any particular text. The alert is saved and assigned to an operator. The operator is an entity with an e-mail address. You can create an SQL Server Agent alert using the message number of the entry.

## Guidelines to Create a Security Strategy

The guidelines to be followed when creating a security strategy using audits and policies are as follows.

- Audits and policies take up some amount of processing on the server for each activity they apply on. In case the server is a high transaction server, you might want to plan around the performance impact that the audits and policies active on your server might have.
- For security reasons, it is good to audit both unsuccessful as well as successful logins. This lets you know if an attacker was able to compromise your server and the actions he performed on the server. This protects your server from unauthorized access.
- Use Policy Based Management to standardize your server setups in a consistent manner across the organization. PBM can be used to create an appropriate surface area by removing components you do not require. PBM can also be used to ensure criteria, such as strong passwords, expiration of keys and passwords by using low privilege Windows accounts as SQL Server instance accounts and by managing updates using Windows Server Update Services (WSUS).
- In case of auditing, security settings allow the server to shut down if the audits cannot be written. The denial of access to the auditing storage will not allow transactions which might not be audited to continue. You will need to balance the uptime and availability of the database against security requirements and make the appropriate choices as per your organization's requirements.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Apply Auditing and Policy Management Strategies



Discuss the impact on performance of a production sales database when using an audit or policy with the students.

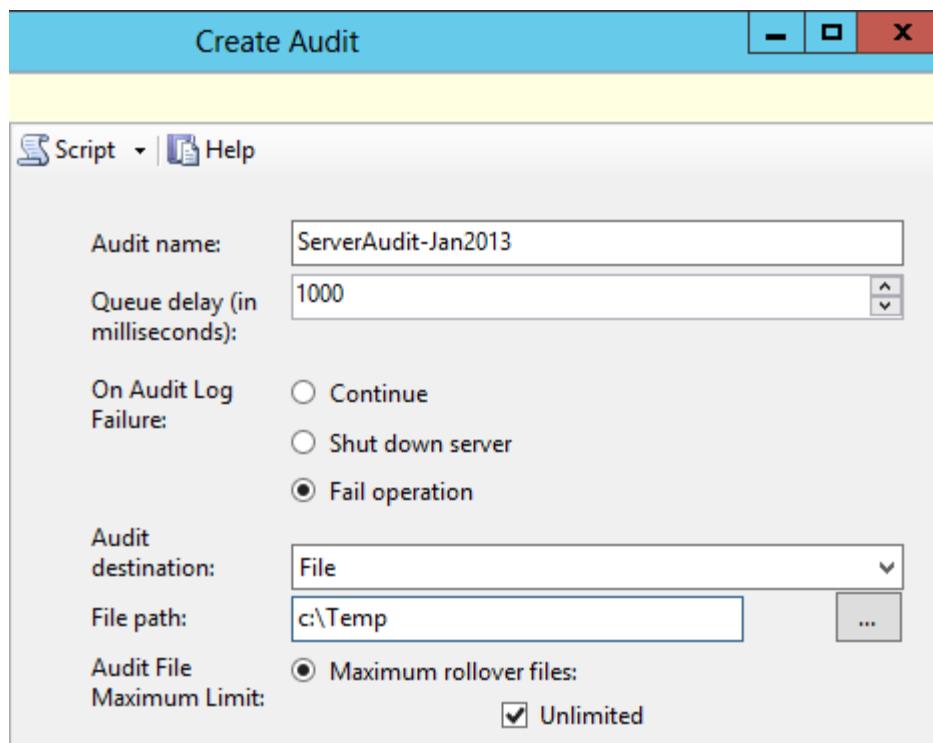
# ACTIVITY 6–3

## Creating a Server Audit

### Scenario

Management of Adventure Works Cycles wants to implement a tracking mechanism for logins so that even the failed attempts to login the server are reported. You are asked to monitor failed logins on the server. You decide to set up a server audit and a server audit specification for auditing failed logins.

1. Create a new Server Audit.
  - a) In the **Object Explorer** pane, in the **Security** folder, right-click **Audits** and select **New Audit**.
  - b) In the Create Audit window, in the **Audit name** text box, type **ServerAudit-Jan2013**.
  - c) In the **On Audit Log Failure** section, select the **Fail operation** option.
  - d) From the **Audit destination** drop-down list, ensure that **File** is selected.
  - e) In the **File path** text box, type **C:\Temp** and select **OK**.

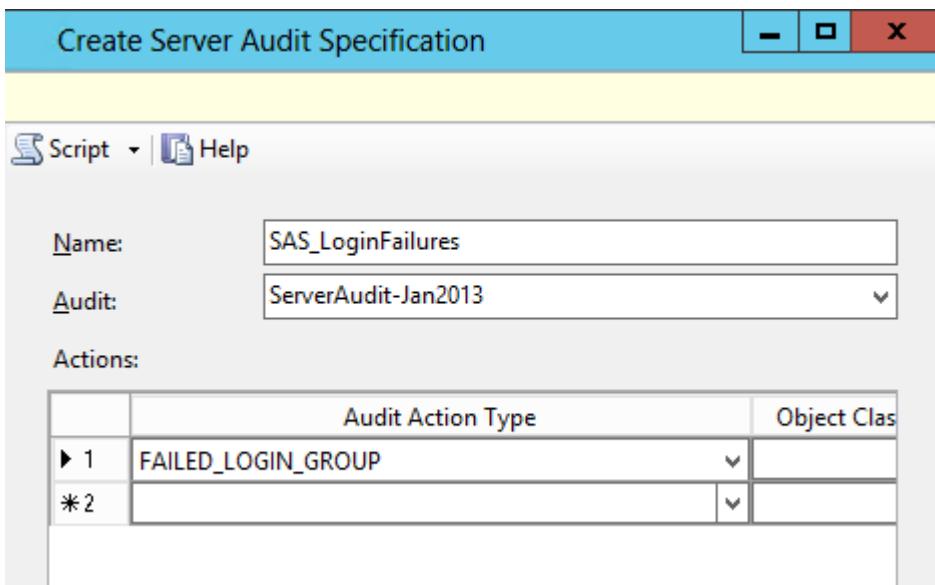


- f) In the **Object Explorer** pane, expand **Audits** and observe that the server audit created is displayed in the **Audits** node.

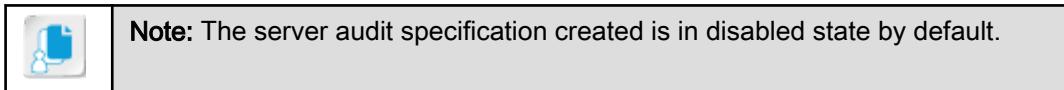
|  |                                                                             |
|--|-----------------------------------------------------------------------------|
|  | <b>Note:</b> The server audit created will be in disabled state by default. |
|--|-----------------------------------------------------------------------------|

2. Create a new server audit specification.
  - a) In the **Object Explorer** pane, right-click **Server Audit Specifications** and select **New Server Audit Specification**.
  - b) In the **Create Server Audit Specification** dialog box, in the **Name** text box, type **SAS\_LoginFailures**

- c) From the **Audit** drop-down list, select **ServerAudit-Jan2013**.
- d) In the **Actions** section, in the **Audit Action Type** column, from the drop-down list, select **FAILED\_LOGIN\_GROUP** and select **OK** to create a new server audit specification.



- e) In the **Object Explorer** pane, expand **Server Audit Specifications** and observe that the **SAS\_LoginFailures** server audit specification is displayed.



3. Enable the server audit and the server audit specification created.
  - a) In **Object Explorer** pane, right-click **SAS\_LoginFailures** and select **Enable Server Audit Specification**.
  - b) In the **Enable Server Audit Specification** dialog box, observe that the **SAS\_LoginFailures** server audit specification is enabled and select **Close**.
  - c) In **Object Explorer** pane, right-click **ServerAudit-Jan2013** and select **Enable Audit**.
  - d) In the **Enable Audit** dialog box, observe that the **ServerAudit-Jan2013** server audit is enabled and select **Close**.
4. Log in with an incorrect login and password.
  - a) In the **Object Explorer** pane, right-click **SQL01** and select **Disconnect**.
  - b) In the **Object Explorer** pane, **Connect→Database Engine**.
  - c) In the **Connect to Server** dialog box, from the **Authentication** drop-down list, select **SQL Server Authentication**.
  - d) In the **Login** and **Password** text boxes, enter an incorrect login and password and select **Connect**.
  - e) In the **Connect to Server** dialog box, an error message stating “cannot connect to server” is displayed. Select **OK**.
5. Verify the audit information for failed login.
  - a) In the **Connect to Server** dialog box, from the **Authentication** drop-down list, select **Windows Authentication** and select **Connect**.
  - b) In the **Object Explorer** pane, expand **SQL01→Security→Audits**.
  - c) Right-click **ServerAudit-Jan2013** and select **View Audit Logs**.
  - d) In the **Log File Viewer** dialog box, in the **Log file summary** section, select a row that has a value **LOGIN FAILED** in the **Action ID** column.
  - e) In the **Log File Viewer** dialog box, resize the **Selected row details** section and observe the details of the failed login.

| Date                   | Event Time       | Server Instance Name | Action ID             |
|------------------------|------------------|----------------------|-----------------------|
| ✓ 1/22/2013 8:07:12 AM | 08:07:12.4378615 | SQL01                | LOGIN FAILED          |
| ✓ 1/22/2013 8:06:47 AM | 08:06:47.8005280 | SQL01                | AUDIT SESSION CHANGED |

Selected row details:

|                      |                                        |
|----------------------|----------------------------------------|
| Date                 | 1/22/2013 8:07:12 AM                   |
| Log                  | Audit Collection (ServerAudit-Jan2013) |
| Event Time           | 08:07:12.4378615                       |
| Server Instance Name | SQL01                                  |
| Action ID            | LOGIN FAILED                           |
| Class Type           | LOGIN                                  |
| Sequence Number      | 1                                      |

- f) Close the Log File Viewer dialog box.

# ACTIVITY 6–4

## Creating a Database Audit Specification

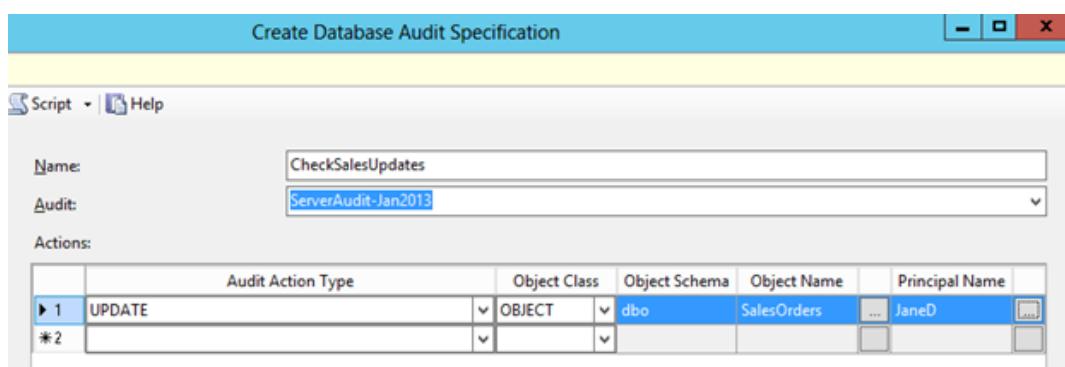
### Scenario

The sales manager of Adventure Works Cycles suspects that someone is manipulating the sales data in the transactions database. He claims that a particular user is changing the data. You are asked to track the changes the user does to the database to prove the sales manager's claim. You decide to create a database audit specification to perform this task.

1. Create a database audit specification to audit the updates done by JaneD on the dbo.SalesOrders table.
  - a) In **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
  - b) In the **Object Explorer** pane, expand **SQL01→Databases→SalesDB→Security**.
  - c) Right-click **Database Audit Specifications** and select **New Database Audit Specification**.
  - d) In the **Create Database Audit Specification** dialog box, in the **Name** text box, type **CheckSalesUpdates**.
  - e) From the **Audit** drop-down list, select **ServerAudit-Jan2013**.
  - f) In the **Actions** pane, in the **Audit Action Type** column, from the drop-down list, select **UPDATE**.
  - g) Scroll right and in the **Object Class** column, from the drop-down list, select **OBJECT**.
  - h) In the **Object Name** column, select the ellipsis (...).
  - i) In the **Select Objects** dialog box, type **SalesOrders** and select **Check Names**.
  - j) In the **Multiple Objects Found** dialog box, check the **[dbo].[SalesOrders]** check box and select **OK**.
  - k) In the **Select Objects** dialog box, select **OK** to add the object for auditing.
  - l) In the **Principal Name** Column, select the ellipsis (...).
  - m) In the **Select Objects** dialog box, type **JaneD** and select **Check Names**.
  - n) In the **Select Objects** dialog box, select **OK** to add **JaneD** to the **Principal Name** column.
  - o) In the **Create Database Audit Specification** dialog box, verify that the settings selected are displayed, and select **OK** to create the database audit specification.



**Note:** The database audit specification created is disabled by default.



2. Enable the database audit specification created.
  - a) In the **Object Explorer** pane, expand **Database Audit Specifications**.
  - b) Right-click **CheckSalesUpdates** and select **Enable Database Audit Specification**.

- c) In the **Enable Database Audit Specification** dialog box, observe that the **Database Audit Specification** is enabled and select **Close**.
  - d) In the **Object Explorer** pane, right-click **SQL01** and select **Disconnect**.
3. Connect to the server as JaneD and perform an update in the dbo.SalesOrders table.
- a) In the **Object Explorer** pane, select **Connect->Database Engine**.
  - b) In the **Connect to Server** dialog box, from the **Authentication** drop-down list, select **SQL Server Authentication**.
  - c) In the **Username** text box, ensure that **JaneD** is selected, in the **Password** text box, type **pass@word1** and select **Connect**.
  - d) Open a **Query Editor** pane and ensure that **SalesDB** database is selected.
  - e) In the **Query Editor** pane, enter an **INSERT INTO** statement with values to insert a new record in the dbo.SalesOrders table.
- ```
INSERT INTO dbo.SalesOrders VALUES ('20130101', 1, 100.20, 5, 501);
```
- f) On the **SQL Editor** toolbar, select **Execute** to execute the query.
 - g) Open a **Query Editor** pane, enter an **UPDATE** statement followed by the **SET** keyword to set a value of 10 to the **Quantity** column of the dbo.SalesOrders table.
- ```
UPDATE dbo.SalesOrders
SET Quantity=10
```
- h) Enter a **WHERE** clause and filter condition to filter records with SalesOrderID 1.
- ```
UPDATE dbo.SalesOrders
SET Quantity=10
WHERE SalesOrderID=1;
```
- i) On the **SQL Editor** toolbar, select **Execute** to execute the **UPDATE** query.
 - j) Close the **Query Editor** panes without saving the queries.
4. Verify the audits for the update done by user JaneD.
- a) In the **Object Explorer** pane, right-click the server connection where you have logged in as **JaneD** and select **Disconnect**.
 - b) In the **Object Explorer** pane, select **Connect->Database Engine**.
 - c) In the **Connect to Server** dialog box, from the **Authentication** drop-down list, select **Windows Authentication** and select **Connect**.
 - d) In the **Object Explorer** pane, expand **Security->Audits**.
 - e) Right-click **ServerAudit-Jan2013** and select **View Audit Logs**.
 - f) In the **Log File Viewer** dialog box, select the row with a value **UPDATE** in the **Action ID** column.
 - g) Expand the details pane to verify the details of the audit. This provides the evidence that the user has manipulated the sales data.



Check the participants to ensure that they were able to observe the username as well as the audit information about the UPDATE statement that is executed by the user.

Target Database	Principal Name
Database Name	SalesDB
Schema Name	dbo
Object Name	SalesOrders
Statement	UPDATE dbo.SalesOrders
SET Quantity=10	
WHERE SalesOrderID=1;	
Additional Information	
File Name	C:\Temp\ServerAudit-Jan2013_9C447816-CBFA-4F86-803D-169B97A3C3A2_0_130033156077840000.sqlaudit
File Offset	12288

- h) Close the **Log File Viewer** dialog box.

ACTIVITY 6–5

Configuring Policy Based Management

Before You Begin

Open a **Query Editor** pane and ensure that the **SalesDB** database is selected.

Scenario

The database manager of Adventure Works Cycles wants all the user-defined stored procedures in the SalesDB database to start with a prefix `usp_`. You decide to create a new policy to enforce the rule.

1. Execute code to create a new stored procedure `getSalesAmounts`.

- a) IN the **Query Editor** pane, enter a `CREATE PROCEDURE` statement followed by the stored procedure name `getSalesAmounts`.

```
CREATE PROCEDURE getSalesAmounts
```

- b) Enter the **AS** keyword followed by a container with the **BEGIN** and **END** keywords.

```
CREATE PROCEDURE getSalesAmounts
AS
BEGIN
END
```

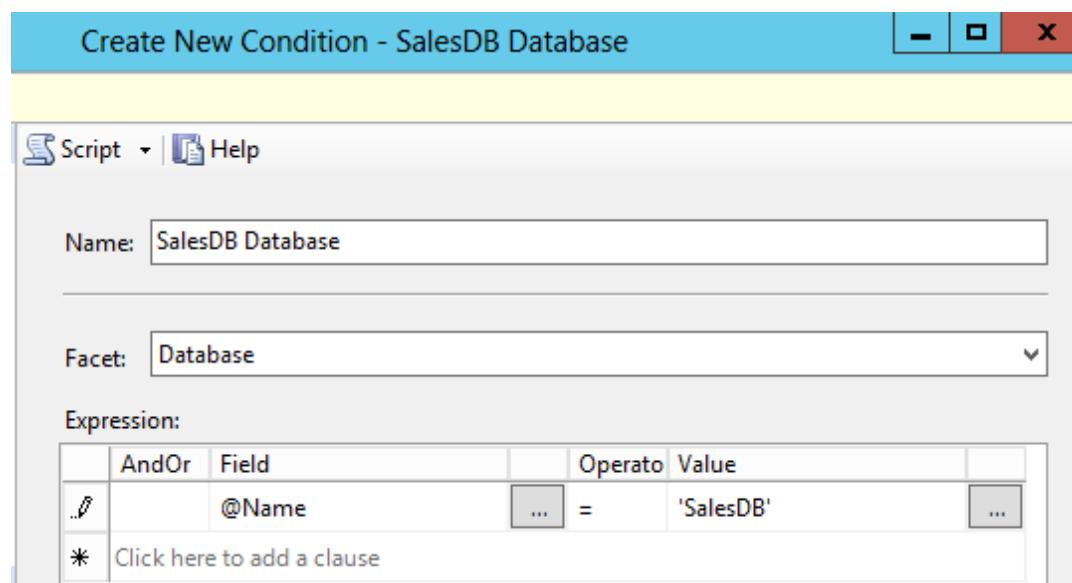
- c) In the container, enter a **SELECT** statement to retrieve values of the `SalesOrderID`, `ProductID`, and `TotalSalesAmount` columns from the `SalesOrders` table.

```
"...
BEGIN
SELECT
SalesOrderID, ListPrice * Quantity AS Sold,
TotalSalesAmount FROM SalesOrders
END
```

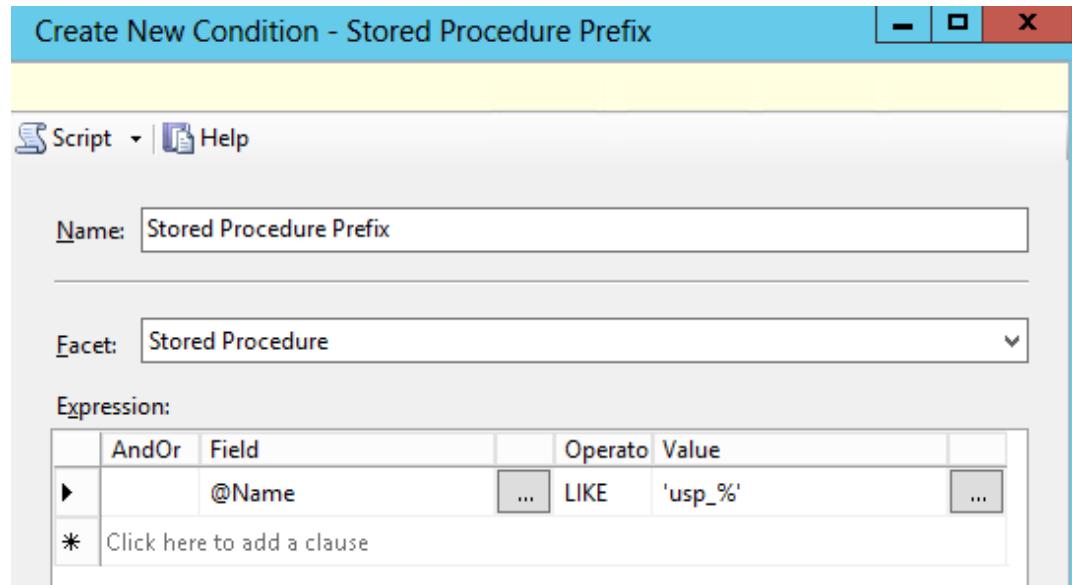
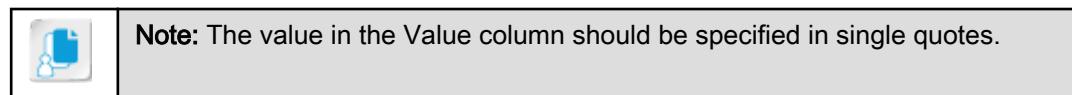
- d) Execute the code to create the `getSalesAmounts` stored procedure.
- e) On the **Messages** tab, observe that the message “Command(s) executed successfully” is displayed.

2. Create a policy to prefix user-defined stored procedures with `usp_`.

- a) In **Object Explorer** pane, expand **Management→Policy Management**.
- b) Right-click **Conditions** and select **New Condition**.
- c) In the **Create New Condition** dialog box, in the **Name** text box, type ***SalesDB Database*** and from the **Facet** drop-down list, select **Database**.
- d) In the **Expression** pane, select **Click here to add a clause**, in the **Field** column from the drop-down list, select **@Name**.
- e) In the **Operator** column, from the drop-down list, select **=** and in the **Value** column select the ellipsis.
- f) In the **Advanced Edit** dialog box, in the **Cell value** section type **'SalesDB'** and select **OK**.
- g) In the **Create New Condition** dialog box, select **OK** to create a new condition.

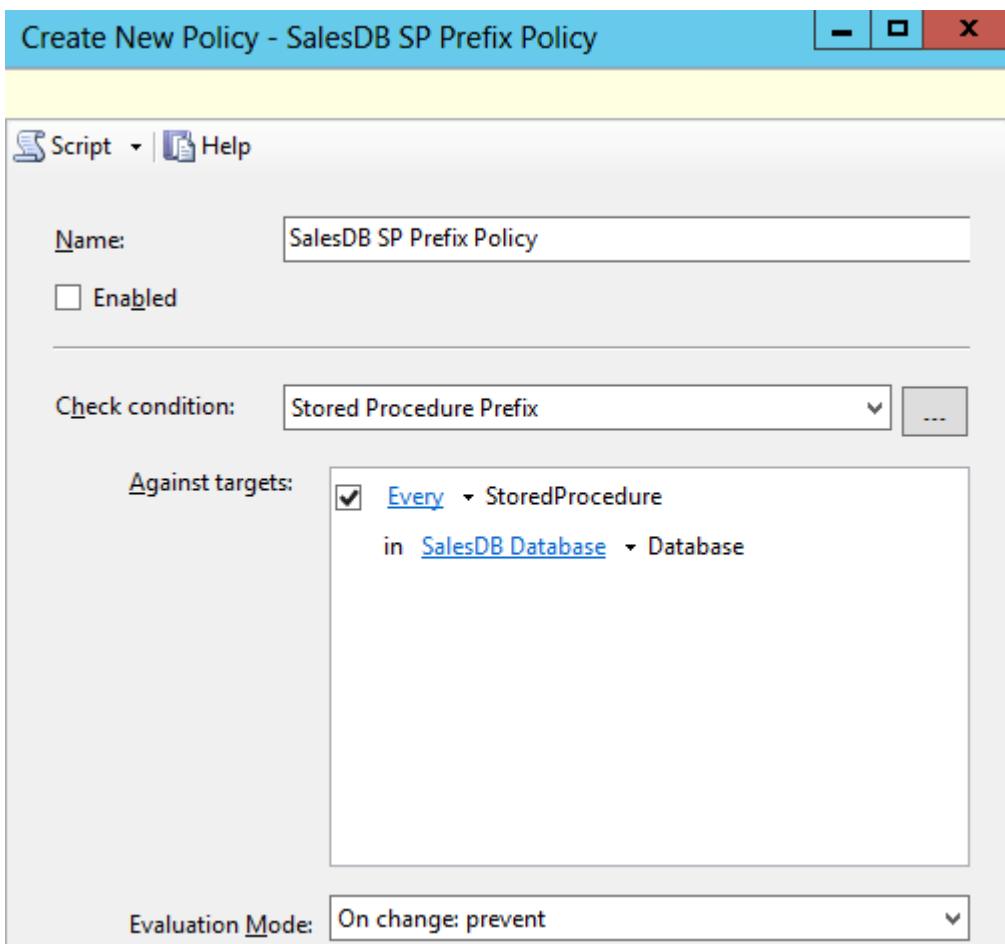


- h) Right-click **Conditions** and select **New Condition**.
- i) In the **Create New Condition** dialog box, in the **Name** text box, type **Stored Procedure Prefix** and from the **Facet** drop-down list, select **Stored Procedure**.
- j) In the **Expression** pane, select **Click here to add a clause**, in the **Field** column from the drop-down list, select **@Name**.
- k) In the **Operator** column, from the drop-down list, select **LIKE** and in the **Value** column select the ellipsis.
- l) In the **Advanced Edit** dialog box, in the **Cell value** section type **'usp_%'** and select **OK**.
- m) In the **Create New Condition** dialog box, select **OK** to create a new condition.



- n) In the **Object Explorer** pane, right-click **Policies** and select **New Policy**.

- o) In the **Create New Policy** dialog box, in the **Name** text box, type **SalesDB SP Prefix Policy**.
- p) From the **Check condition** drop-down list, select **Stored Procedure Prefix**.
- q) In the **Against targets** section, from the **Database** drop-down list, select **SalesDB Database** and ensure that the **Every Stored Procedure in SalesDB Database** check box is checked.
- r) From the **Evaluation Mode** drop-down list, select **On change: prevent** and select **OK**.



Remind students that the new policy will be disabled by default.

3. Enable and evaluate the policy created.
 - a) In the **Object Explorer** pane, expand **Policies** and right-click **SalesDB SP Prefix Policy** and select **Enable**.
 - b) Right-click **SalesDB SP Prefix Policy** and select **Evaluate**.
 - c) In the **Evaluate Policies** dialog box, observe that an error is displayed.
 - d) In the **Target details** pane, select the **View** link in the **Details** column to view the error details.
 - e) In the **Results Detailed View** dialog box, observe that the expected and actual value for **@Name** field is displayed and select **Close**.
 - f) In the **Evaluate Policies** dialog box, select **Close**.

4. Execute the code to create a new stored procedure named `getProducts`.

- a) Open the **Query Editor** pane, enter a **CREATE PROCEDURE** statement followed by the stored procedure name `getProducts`.

```
CREATE PROCEDURE getProducts
```

- b) Enter the **AS** keyword followed by a container with the **BEGIN** and **END** keywords.

```
CREATE PROCEDURE getProducts
AS
```

```
BEGIN  
END
```

- c) In the container, enter a **SELECT** statement to retrieve values of the SalesOrderID, ProductID, and TotalSalesAmount columns from the SalesOrders table.

```
"...  
BEGIN  
SELECT  
SalesOrderID,  
ProductID,  
TotalSalesAmount FROM SalesOrders  
END
```

- d) Execute the code to create the getProducts stored procedure.



Note: The query will take some time to execute since the policy is evaluated.

- e) On the **Messages** tab, observe that the error message stating that "Policy SalesDB Prefix Policy has been violated" is displayed.

Messages

```
Policy 'SalesDB SP Prefix Policy' has been violated by 'SQLSERVER:\  
This transaction will be rolled back.  
Policy condition: '@Name LIKE 'usp_%'''  
Policy description: ''  
Additional help: '' : ''  
Statement: 'CREATE PROCEDURE getProducts  
AS  
BEGIN  
    SELECT  
        SalesOrderID,  
        ProductID,  
        TotalSalesAmount  
    FROM SalesOrders  
END  
'.  
Msg 3609, Level 16, State 1, Procedure sp_syspolicy_dispatch_event,  
The transaction ended in the trigger. The batch has been aborted.
```

- f) In the **Query Editor** pane, modify the name of the stored procedure to usp_getProducts.
g) Execute the code to create the usp_getProducts stored procedure.
h) On the **Messages** tab, observe that the message "Command(s) is completed successfully." is displayed.
i) Close the **Query Editor** panes without saving the query.

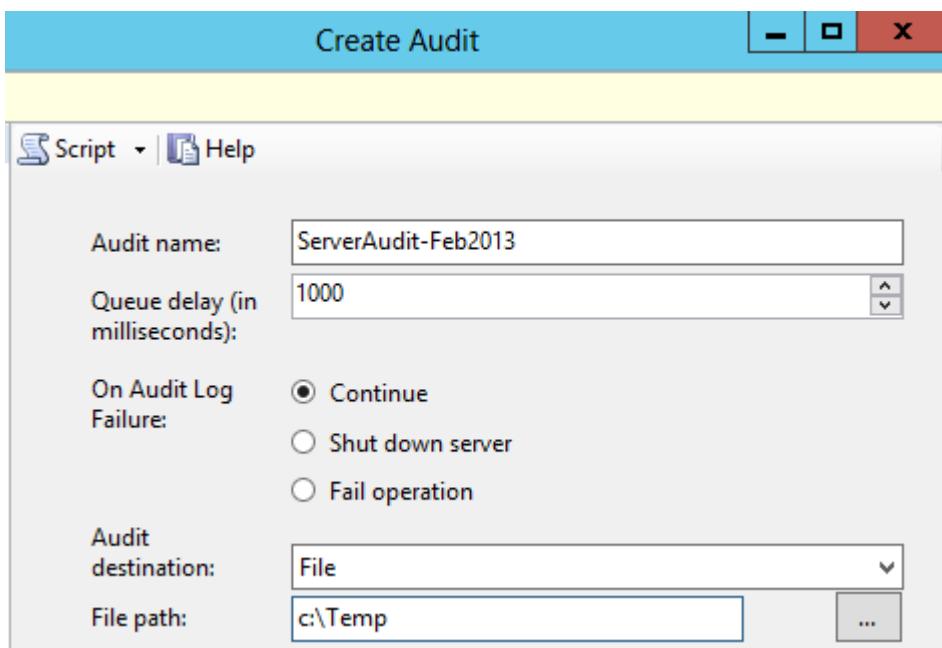
ACTIVITY 6–6

Tracking Privilege Elevation within the Server

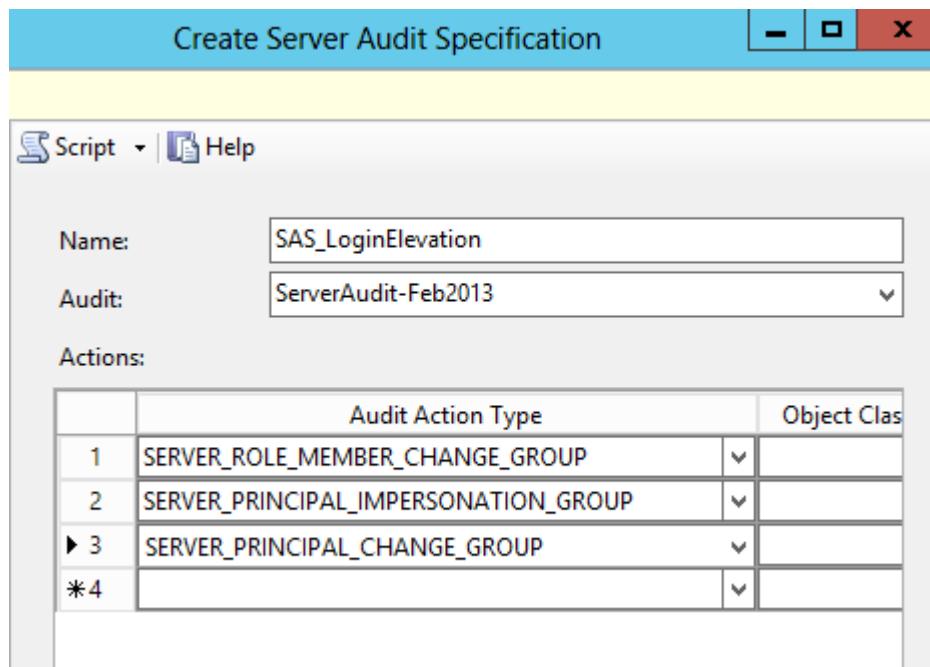
Scenario

The manager of Adventure Works Cycles has been getting reports that someone with access is changing the permissions for users by granting additional privileges and revoking privileges. You decide to audit the changes made to the privileges in the server.

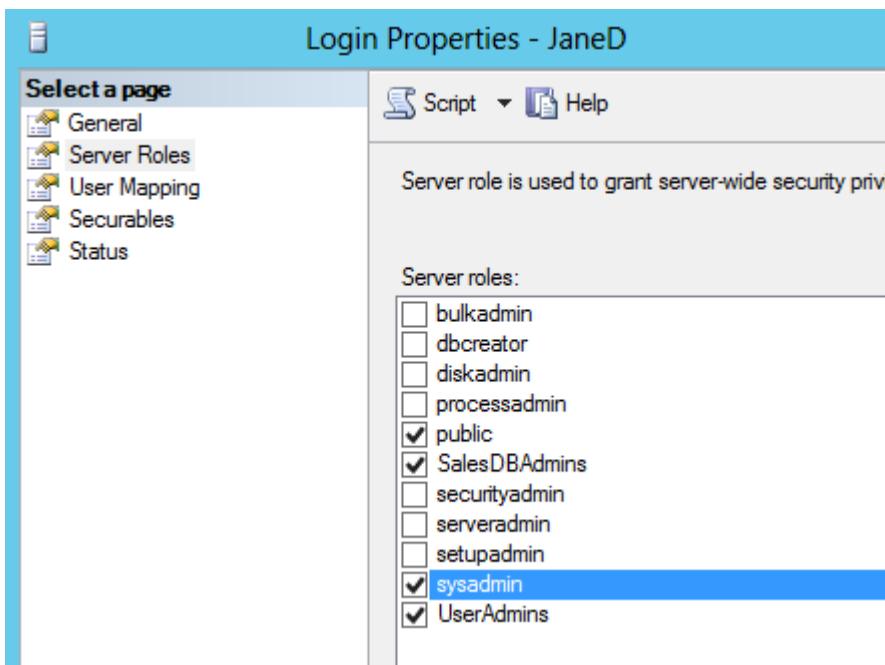
1. Create a server audit and server audit specification to track the changes made to server roles and principals.
 - a) In **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
 - b) In **Object Explorer** pane, expand **SQL01→Security**.
 - c) Right-click **Audits** and select **New Audit**.
 - d) In the **Create Audit** dialog box, in the **Audit Name** text box, type **ServerAudit-Feb2013**
 - e) From the **Audit destination** drop-down list, ensure that **File** is selected.
 - f) In the **File path** text box, type **C:/Temp** and select **OK**.



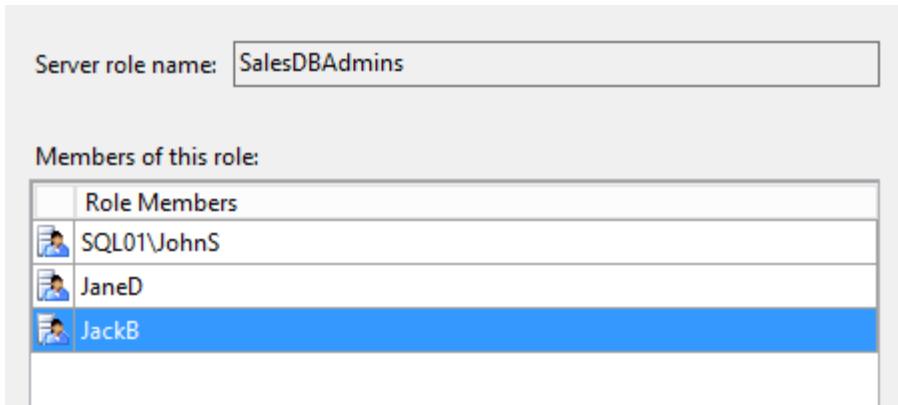
- g) In the **Object Explorer** pane, right-click **Server Audit Specifications** and select **New Server Audit Specification**.
- h) In the **Create Server Audit Specification** dialog box, in the **Name** text box, type **SAS_LoginElevation**.
- i) From the **Audit** drop-down list, select **ServerAudit-Feb2013**.
- j) In the **Actions** section, in the **Audit Action Type** column, from the drop-down list, select the following audit action types to audit the changes made to the server roles, server principals and select **OK**.
 - SERVER_ROLE_MEMBER_CHANGE_GROUP
 - SERVER_PRINCIPAL_IMPERSONATION_GROUP
 - SERVER_PRINCIPAL_CHANGE_GROUP



2. Enable the server audit and the server audit specification created.
 - a) In Object Explorer pane, expand **Server Audit Specifications** node and right-click **SAS_LoginElevation** and select **Enable Server Audit Specification**.
 - b) In the **Enable Server Audit Specification** dialog box, observe that the **SAS_LoginElevation** server audit specification is enabled and select **Close**.
 - c) In Object Explorer pane, expand **Audits** and right-click **ServerAudit-Feb2013** and select **Enable Audit**.
 - d) In the **Enable Audit** dialog box, observe that the **ServerAudit-Feb2013** server audit is enabled and select **Close**.
3. Login as JaneD and attempt to elevate privileges.
 - a) In the **Object Explorer** pane, right-click **SQL01** and select **Disconnect**.
 - b) In the **Object Explorer** pane, select **Connect->Database Engine**.
 - c) In the **Connect to Server** dialog box, ensure that **SQL01** is selected.
 - d) From the **Authentication** drop-down list, select **SQL Server authentication**.
 - e) In the **Login** text box, type **JaneD** and in the **Password** text box, type **pass@word1** and select **Connect**.
 - f) In the **Object Explorer** pane, expand **Server->Security->Logins** and double-click **JaneD**.
 - g) In the **Login Properties** dialog box, in the **Select a Page** section, select the **Server Roles** page.
 - h) In the **Server roles** section, check the **sysadmin** check box and select **OK** to attempt to elevate JaneD as sysadmin.



- i) Observe that an error is displayed and close the message box and select **Cancel**.
4. Revoke the SalesDBAdmins server role permission to JackB.
 - a) In the **Object Explorer** pane, expand **Server Roles** and double-click **SalesDBAdmins**.
 - b) In the **Server Role Properties** dialog box, in the **Members of this role** section, select **JackB** and select **Remove** to remove permissions applied for JackB on the SalesDBAdmins server role.



- c) Select **OK** to save the changes made.
5. Verify the audit logs for changes made to the privileges.
 - a) In the **Object Explorer** pane, right-click the server connection under the **JaneD** login and select **Disconnect**.
 - b) In the **Object Explorer** pane, select **Connect→Database Engine**.
 - c) From the **Authentication** drop-down list, select **Windows authentication** and select **Connect**.
 - d) In the **Object Explorer** pane expand **SQL01→Security→Audits**.
 - e) Right-click **ServerAudit-Feb2013** and select **View Audit Logs**.
 - f) In the **Log File Viewer** dialog box, in the **Log file Summary** section, observe that the audits for the actions **DROP MEMBER** and **ADD MEMBER** are displayed.

Action ID	Class Type	Sequence Number	Succeeded
DROP MEMBER	SERVER ROLE	1	True
ADD MEMBER	SERVER ROLE	1	False
AUDIT SESSION CHANGED	SERVER AUDIT	0	True

- g) Close the Log File Viewer window.
-

ACTIVITY 6–7

Tracking Database Object Modification

Scenario

Management of Adventure Works Cycles wants all the changes made to the database and database objects to be recorded. You are asked to keep track of the changes done to the database and database objects. You decide to use the database audit specification to audit these changes in the log.

1. Modify the existing database audit specification to include database audit types.
 - a) In **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
 - b) In **Object Explorer** pane, expand **SQL01→Databases→SalesDB→Security→Database Audit Specifications**.
 - c) Right-click **CheckSalesUpdates** and select **Disable Database Audit Specification** to modify the database audit specification.
 - d) In the **Disable Database Audit Specification** dialog box, select **Close**.
 - e) Double-click **CheckSalesUpdates** to open the **Properties** dialog box.
 - f) In the **Database Audit Specification Properties** dialog box, in the **Actions** pane, in the **Audit Action Type** column, select the drop-down list and add the following Audit Action Type groups to audit the changes made to the database.
 - DATABASE_CHANGE_GROUP
 - DATABASE_OBJECT_CHANGE_GROUP
 - DATABASE_ROLE_MEMBER_CHANGE_GROUP
 - DATABASE_PERMISSION_CHANGE_GROUP
 - DATABASE_OBJECT_PERMISSION_CHANGE_GROUP
 - DATABASE_PRINCIPAL_CHANGE_GROUP
 - SCHEMA_OBJECT_CHANGE_GROUP
 - DATABASE_OPERATION_GROUP
 - DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP
 - DATABASE_OWNERSHIP_CHANGE_GROUP
 - g) In the **Database Audit Specification Properties** dialog box, select **OK** to create a database audit specification with the selected audit types.



Note: The groups mentioned do not have additional parameters.

	Audit Action Type	Object Class	Object Schema
01	UPDATE	OBJECT	dbo
02	DATABASE_CHANGE_GROUP		
03	DATABASE_OBJECT_CHANGE_GROUP		
04	DATABASE_ROLE_MEMBER_CHANGE_GROUP		
05	DATABASE_PERMISSION_CHANGE_GROUP		
06	DATABASE_OBJECT_PERMISSION_CHANGE_GROUP		
07	DATABASE_PRINCIPAL_CHANGE_GROUP		
08	DATABASE_OBJECT_CHANGE_GROUP		
09	DATABASE_OPERATION_GROUP		
10	DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP		
► 11	DATABASE_OWNERSHIP_CHANGE_GROUP		
*12			

2. Enable the database audit specification.
 - a) In the **Object Explorer** pane, right-click **CheckSalesUpdates** and select **Enable Database Audit Specification**.
 - b) In the **Enable Database Audit Specification** dialog box, select **Close**.
 - c) In the **Object Explorer** pane, expand **Tables** folder in the SalesDB database.
 - d) Right-click **dbo.SalesOrders** and select **Design**.
 - e) In the **SQL01.SalesDB-dbo.SalesOrders** pane, in the **Column Name** column, type **CustomerID**.
 - f) In the **Data Type** column, from the drop-down list, select **int** and in the **Allow Nulls** column, ensure that the check box is checked.

	Column Name	Data Type	Allow Nulls
	SalesOrderID	int	<input type="checkbox"/>
	SalesDate	date	<input type="checkbox"/>
	ProductID	int	<input type="checkbox"/>
	ListPrice	float	<input type="checkbox"/>
	Quantity	int	<input type="checkbox"/>
	TotalSaleAmount	float	<input type="checkbox"/>
►	CustomerID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- g) On the **Standard** toolbar, select the **Save** button to save the table design.
- h) Close the **Table Design** pane.

3. Verify the audit log files.

- In **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
- In **Object Explorer** pane, expand **SQL01→Security→Audits**.
- Right-click **ServerAudit-Jan13** and select **View Audit Logs**.
- In the **Log File Viewer** dialog box, select the row with value **ALTER** in the **Action ID** column.
- Resize the **Selected row details** pane and observe that the command to add the column to the table is tracked along with the time and the user who actually performed the action.

Date	Event Time	Server Instance Name	Action ID
✓ 1/23/2013 5:10:19 AM	05:10:19.1420801	SQL01	ALTER
✓ 1/23/2013 5:10:19 AM	05:10:19.1420801	SQL01	ALTER
✓ 1/23/2013 5:09:01 AM	05:09:01.7340651	SQL01	LOGIN FAILED
✓ 1/23/2013 4:56:35 AM	04:56:35.7397987	SQL01	ALTER
✓ 1/23/2013 4:56:35 AM	04:56:35.7397987	SQL01	ALTER
✓ 1/23/2013 4:56:31 AM	04:56:31.2764656	SQL01	LOGIN FAILED
✓ 1/23/2013 4:53:04 AM	04:53:04.0814917	SQL01	LOGIN FAILED

Selected row details:

Date: 1/23/2013 5:10:19 AM
 Log: Audit Collection (ServerAudit-Jan2013)

Event Time: 05:10:19.1420801
 Server Instance Name: SQL01
 Action ID: ALTER
 Class Type: TABLE

- Close the **Log File Viewer** dialog box.

Summary

In this lesson, you created certificate-based logins and applied auditing- and policy-based management strategies in your server. This enables you to ensure that all the data transactions and changes made to the server are monitored and the server is prevented against any security breaches.



Use the review questions provided to generate discussion among the participants.

When dealing with database level changes, which option of tracking will you use?

A: Answers will vary, but may include: auditing in case of only tracking and policies in order to enforce the rules.

What types of password policies would you implement in your organization to ensure security?

A: Answers will vary, but may include: password length, complexity of the password, password expiry, and repetition of passwords.



Note: Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

7

Applying Encryption and Compression

Lesson Time: 1 hour, 15 minutes

Lesson Objectives

In this lesson, you will:

- Encrypt a database.
- Compress a database table.

Lesson Introduction

You have implemented advanced security settings to audit the server as well as the databases within it. When working with an organizational database, it is important to enhance the security as well as optimize the databases by encrypting and compressing the database. Familiarizing yourself with the database encryption and database compression techniques will enable you to prevent unauthorized access to the database and ensure increased storage space respectively. In this lesson, you will apply encryption and compression.

TOPIC A

Apply Encryption on Databases

When a database has sensitive and highly confidential information, it is likely that it would be the target of malicious hackers to retrieve the confidential information in it. You can protect the database from hackers by encrypting the physical files of the database. A better understanding on applying encryption to the databases will enable you to prevent unauthorized access to your database data. In this topic, you will encrypt a database.



TDE

TDE

Transparent Data Encryption (TDE) is a feature of SQL Server that allows data to be encrypted and decrypted on the fly by SQL Server as it is written or read from the database file on disk. The I/O operation uses the encrypted data only in both directions.

It is the SQL Server engine that performs the translation between the original and encrypted state in memory. Because of this, there is no requirement for any change in the DDL syntax used to store or retrieve the data. This is the reason the feature is also known as “Transparent” as neither the user nor the developers need to know that the data is actually stored in an encrypted form in the database.

TDE works on the entire database including all the data files and log files. TDE uses Advanced Encryption Standard (AES) or Triple Data Encryption Standard (3DES) algorithms to ensure that the data is very difficult to break. The encryption is done using a Database Encryption Key (DEK) that is protected by a certificate stored on the master database of the server.

Even if the database is stolen and added to a different server, it will refuse to open as the DEK key will be missing on the second server. If you are legitimately moving the database to a new server, you will need to move the DEK key as well as the certificate used to protect the DEK key to the target server before being able to add the TDE database to it.

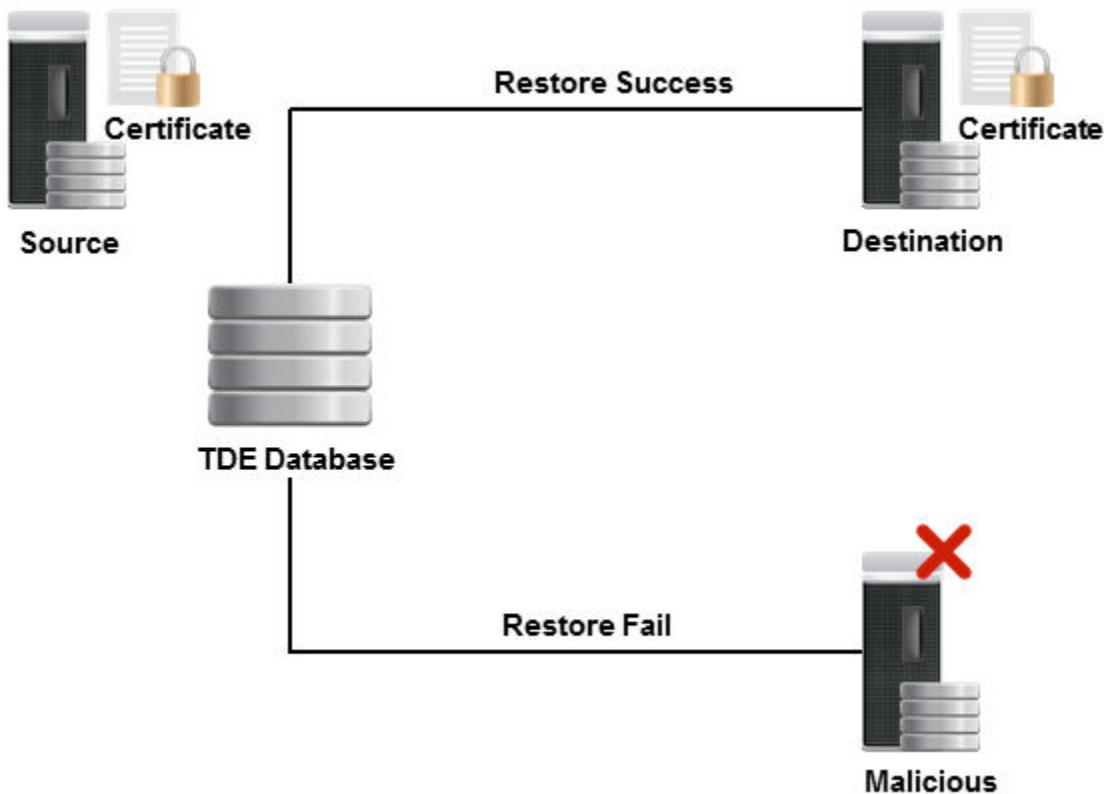


Figure 7-1: Response of a TDE database during successful authorization and malicious attacks.



Note: For more information about TDE, check out the LearnTO **Encrypt Databases Using TDE** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.



Ask the students whether using a certificate is the most secure form of encryption.



You may want to show **LearnTO Encrypt Databases Using TDE** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



Access the **Checklist** tile on your LogicalCHOICE course screen for reference information and job aids on **How to Apply Encryption to Databases**

ACTIVITY 7–1

Applying TDE on a Database

Before You Begin

1. In the **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
2. Open a new **Query Editor** pane and ensure that the **master** database is selected.

Scenario

The sales team of Adventure Works Cycles wants to secure the information present in the Sales database as it contains sensitive financial data. You decide to secure the database using TDE so that there is minimal impact on client applications while also ensuring data safety.

1. Create a master key and a certificate.
 - a) In the **Query Editor** pane, enter the CREATE MASTER KEY statement followed by the ENCRYPTION BY PASSWORD keyword with a defined password 'pass@word1'.


```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'pass@word1'
```
 - b) Execute the CREATE MASTER KEY statement to create the master key.
 - c) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.

 **Note:** If you get an error message stating, 'There is already a master key in the database.' Proceed with step d.

 - d) Enter the CREATE CERTIFICATE statement followed by the certificate name as tdeCert, along with the WITH SUBJECT keyword with the subject of the certificate as TDE Certificate.


```
CREATE CERTIFICATE tdeCert WITH SUBJECT = 'TDE Certificate';
```
 - e) Execute the CREATE CERTIFICATE statement to create a certificate.
 - f) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.
2. Create a backup of the certificate.
 - a) Open a **Query Editor** pane, enter the BACKUP CERTIFICATE keyword followed by the certificate name tdeCert, type the TO FILE keyword followed by the path to save the public certificate in the location 'c:\Temp\public.crt'.


```
BACKUP CERTIFICATE tdeCert TO FILE = 'c:\Temp\public.crt'
```
 - b) Enter the WITH PRIVATE KEY followed by the location to save the private certificate in the location 'c:\temp\private.crt'.


```
BACKUP CERTIFICATE tdeCert TO FILE = 'c:\Temp\public.crt'  
WITH PRIVATE KEY ( FILE = 'c:\Temp\private.crt',
```
 - c) Enter the ENCRYPTION BY PASSWORD keyword followed by the password as 'pass@word1'.


```
BACKUP CERTIFICATE tdeCert TO FILE = 'c:\Temp\public.crt'  
WITH PRIVATE KEY ( FILE = 'c:\Temp\private.crt',  
ENCRYPTION BY PASSWORD = 'pass@word1');
```
 - d) Execute the BACKUP CERTIFICATE statement to backup the public and private keys of the certificate.
 - e) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.
 - f) In the Windows Explorer, navigate to c:\temp folder to view the backup of public and private certificates.

Computer ▶ Local Disk (C:) ▶ Temp			Search Temp
Name	Date modified	Type	
JackBCert	3/6/2013 2:15 PM	Security Certificate	
JackBPvtKey.key	3/6/2013 2:15 PM	KEY File	
private	3/6/2013 5:57 PM	Security Certificate	
public	3/6/2013 5:57 PM	Security Certificate	

3. Setup the DEK within the database.

- a) On the **SQL Editor** toolbar, change the working database from **Master** to **SalesDB**.
- b) Open a **Query Editor** pane, enter the CREATE DATABASE statement followed by the ENCRYPTION KEY WITH ALGORITHM keyword followed by the algorithm name **AES_256**

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256
```
- c) Enter the ENCRYPTION BY SERVER CERTIFICATE keyword followed by the certificate name **tdeCert**.


```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE tdeCert;
```
- d) Execute the statement to setup the Database Encryption Key (DEK) within the database.
- e) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.

4. Enable TDE on the SalesDB database.

- a) Open a **Query Editor** pane, and enter the ALTER DATABASE statement followed by the database name SalesDB.


```
ALTER DATABASE SalesDB
```
- b) Enter the SET ENCRYPTION keyword with the encryption value as ON.


```
ALTER DATABASE SalesDB
SET ENCRYPTION ON;
```
- c) Execute the ALTER DATABASE statement.
- d) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.



Note: You can also enable the encryption feature by right-clicking the database, select **Properties**→**Options**→**Encryption Enabled**; and set it to **True**.

5. Check the state of the database on the server.

- a) Open a **Query Editor** pane, enter the SELECT statement followed by db_name to list all the database names in the server with their database_id with an alias name as db.


```
SELECT db_name(database_id) as db,
```
- b) Type the encryption_state and encryption_type keywords to check the encryption state and encryption type of the databases on the server.


```
SELECT db_name(database_id) as db, encryption_state, encryptor_type
```
- c) Enter the FROM clause followed by the default DMV key sys.dm_database_encryption_keys to display keys available in the system databases.


```
SELECT db_name(database_id) as db, encryption_state, encryptor_type
FROM sys.dm_database_encryption_keys;
```
- d) Execute the SELECT statement to display the keys and certificates available in the server.
- e) On the **Results** tab, observe that the SalesDB database is encrypted with a certificate.

The screenshot shows a SQL Server Management Studio window titled "SQLQuery1.sql - SQ...Administrator (53)*". A query is run against the system catalog view `sys.dm_database_encryption_keys`:

```
SELECT db_name(database_id) as db,
       encryption_state, encryptor_type
  FROM sys.dm_database_encryption_keys;
```

The results grid displays two rows of data:

	db	encryption_state	encryptor_type
1	tempdb	3	ASYMMETRIC KEY
2	SalesDB	3	CERTIFICATE

6. Read and write data into SalesOrders table from the SalesDB database.

- Open a **Query Editor** pane, enter the `INSERT INTO` statement followed by the table name `SalesOrders` with the following values.
 - `Salesdate: 2013-01-01`
 - `ProductID: 1`
 - `ListPrice: 29.95`
 - `Quantity: 3`
 - `TotalSaleAmount: 89.85`

```
INSERT INTO SalesOrders
(Salesdate, ProductID, ListPrice, Quantity, TotalSalesAmount)
VALUES ('2013-01-01', 1, 29.95, 3, 89.85);
```
 - Execute the statement to insert the specified values into the `SalesOrders` table.
 - On the **Messages** tab, observe that a message '(1 row(s) affected)' is displayed.
 - Open a **Query Editor** pane and enter a `SELECT` statement to display all records from `SalesOrders` table.
- ```
SELECT * FROM SalesOrders;
```
- Execute the statement and observe that the inserted new row along with the records in the `SalesOrders` table. You do not have to perform any additional configuration for database encryption. The encryption settings will be updated to the new record.

The screenshot shows a SQL Server Management Studio window titled "SQLQuery1.sql - SQ...Administrator (53)\*". A query is run against the `SalesOrders` table:

```
SELECT * FROM SalesOrders;
```

The results grid displays two rows of data:

|   | SalesOrderID | SalesDate  | ProductID | ListPrice | Quantity | TotalSalesAmount | CustomerID |
|---|--------------|------------|-----------|-----------|----------|------------------|------------|
| 1 | 1            | 2013-01-01 | 1         | 100.2     | 10       | 501              | NULL       |
| 2 | 2            | 2013-01-01 | 1         | 29.95     | 3        | 89.85            | NULL       |

- Close the **Query Editor** panes without saving the queries.

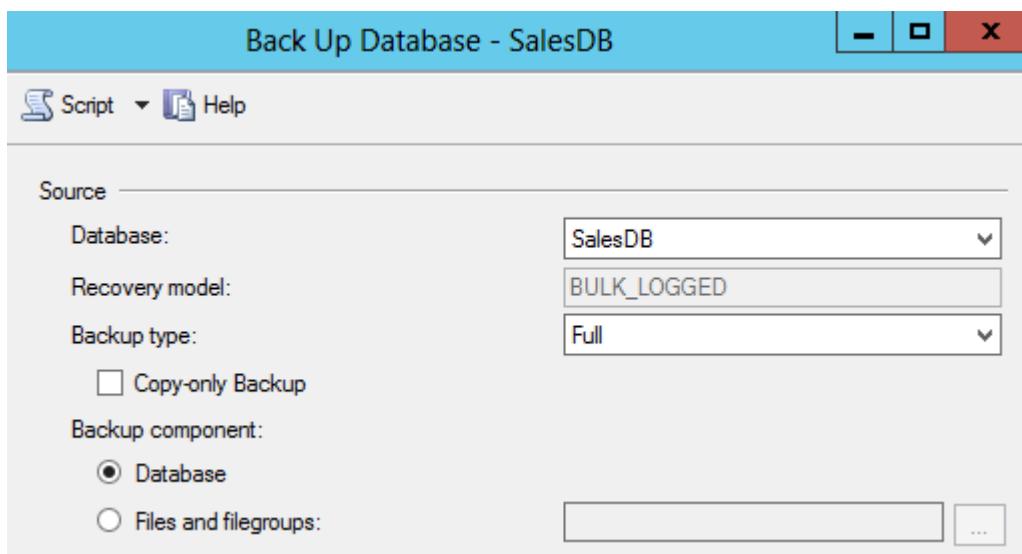
# ACTIVITY 7–2

## Preventing Data Loss During Backup

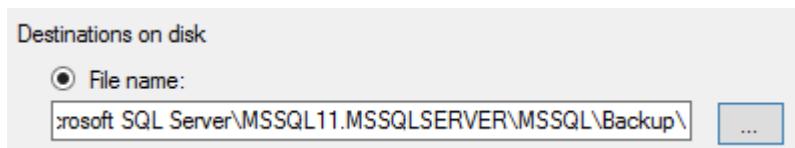
### Scenario

The sales team is concerned that sensitive data might fall into the wrong hands when your database backup is being taken to an offsite location. Because you know that the database is protected with TDE, you wish to check whether someone can use the backup to restore the database and gain access to the sensitive data.

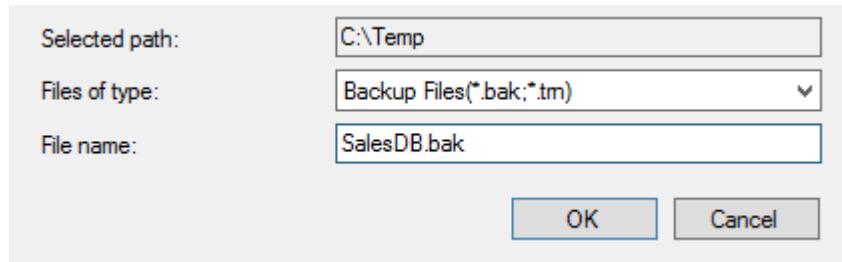
1. Backup the SalesDB database.
  - a) In the **Object Explorer** pane, expand **Databases**→**SalesDB**.
  - b) Right-click **SalesDB** and select **Tasks**→**Back Up**.
  - c) In the **Backup Database – SalesDB** window, in the right pane, ensure that the following details are selected.
    - Database: **SalesDB**
    - Backup type: **Full**
    - Backup component: **Database**



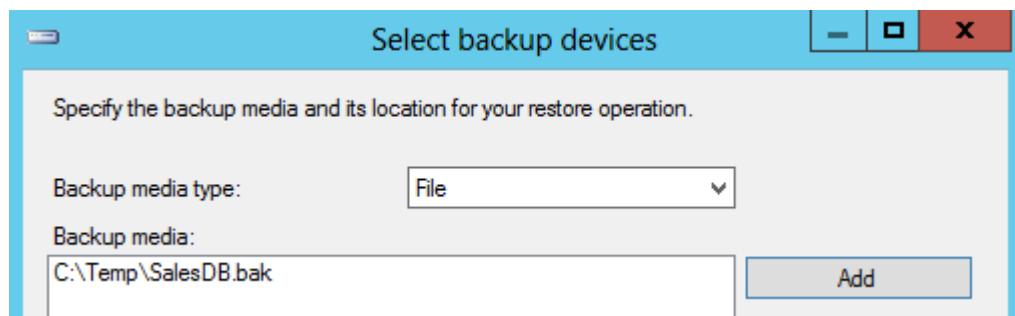
- d) In the **Backup Database – SalesDB** window, in the **Destination** section, select the **Remove** button to remove the default backup destination.
- e) Select **Add** to add a new destination.
- f) In the **Select Backup Destination** dialog box, select the **ellipsis (...)** button.



- g) In the **Locate Database Files** dialog box, navigate to the C:\ drive and select the **Temp** folder.
- h) In the **File name** text box, type **SalesDB.bak** and select **OK** to add the new destination.



- i) In the **Select Backup Destination** dialog box, select **OK**.
  - j) In the **Back Up Database – SalesDB** dialog box, select **OK** to initiate the backup.
  - k) In the **Microsoft SQL Server Management Studio** message box, observe that the backup completed message is displayed and select **OK** to close the message box.
  - l) In the **Object Explorer** pane, right-click **SQL01→Disconnect**.
2. Restore the backup to a different server.
- a) In the **Object Explorer** pane, from the **Connect** drop-down list, select **Database Engine**.
  - b) In the **Connect to Server** dialog box, type **SQL01\INSTANCE01** and select **Connect**.
  - c) In the **Object Explorer** pane, expand **SQL01\INSTANCE01→Databases**.
  - d) Right-click **Databases** and select **Restore Database**.
  - e) In the **Restore Database** dialog box, from the **Source** section, select **Device** and select the **ellipsis (...)** button.
  - f) In the **Select backup devices** dialog box, select the **Add** button.
  - g) In the **Locate Backup File** dialog box, in the **File name** text box, type **C:\Temp\SalesDB.Bak**, and then select **OK**.
  - h) In the **Select backup devices** dialog box, in the **Backup media** section, observe that the backup file is displayed.

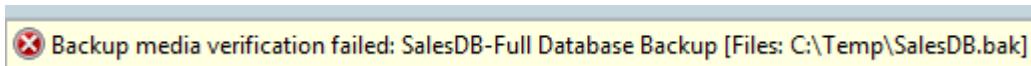


- i) In the **Select backup devices** dialog box, select **OK** to return to the **Restore Database** dialog box.
- j) In the **Restore Database** dialog box, in the **Backup sets to restore** section, observe that the backup set of the SalesDB database is displayed.

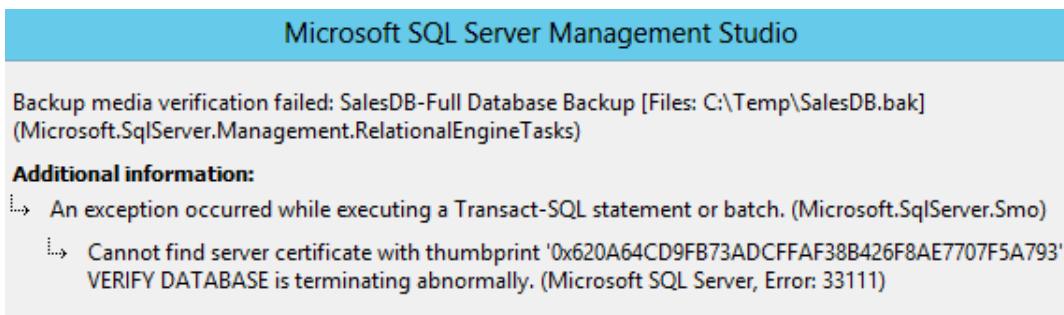
| Restore plan                        |                              |           |      |        |          |          |  |
|-------------------------------------|------------------------------|-----------|------|--------|----------|----------|--|
| Backup sets to restore:             |                              |           |      |        |          |          |  |
| Restore                             | Name                         | Component | Type | Server | Database | Position |  |
| <input checked="" type="checkbox"/> | SalesDB-Full Database Backup | Database  | Full | SQL01  | SalesDB  | 1        |  |

- k) In the **Restore Database** dialog box, select the **Verify Backup Media** button.

- i) On the top of the dialog box, an error message stating 'Backing media verification failed' is displayed.



- m) Select the error message to view the error details. The error indicates that the encryption certificate is missing and hence the system is unable to proceed with the recovery. You will not be able to complete the restore of the backup. Select **OK** to close the error message.



- n) In the **Restore Database** dialog box, select **Cancel** to exit the database restore process. This ensures that the database is safe and only a user with a valid permission will be have access to the sensitive data in the database.

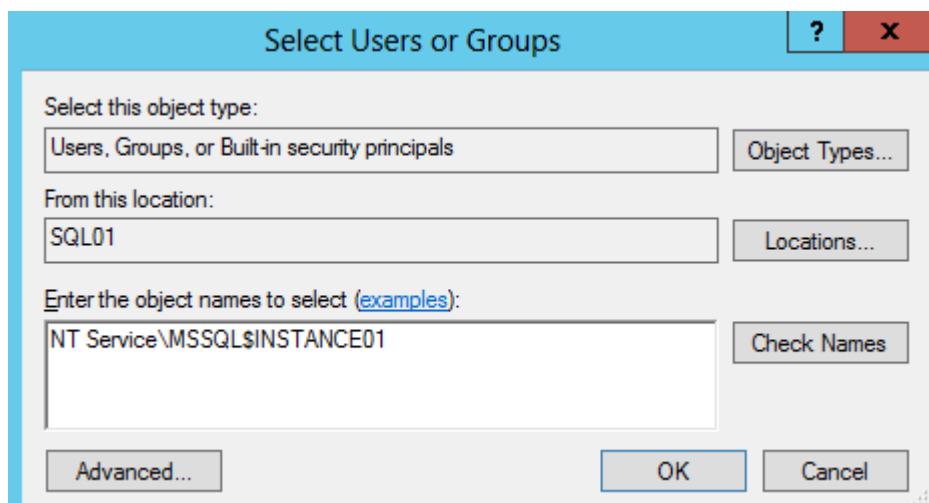
# ACTIVITY 7–3

## Restoring a Database Using the TDE Certificates

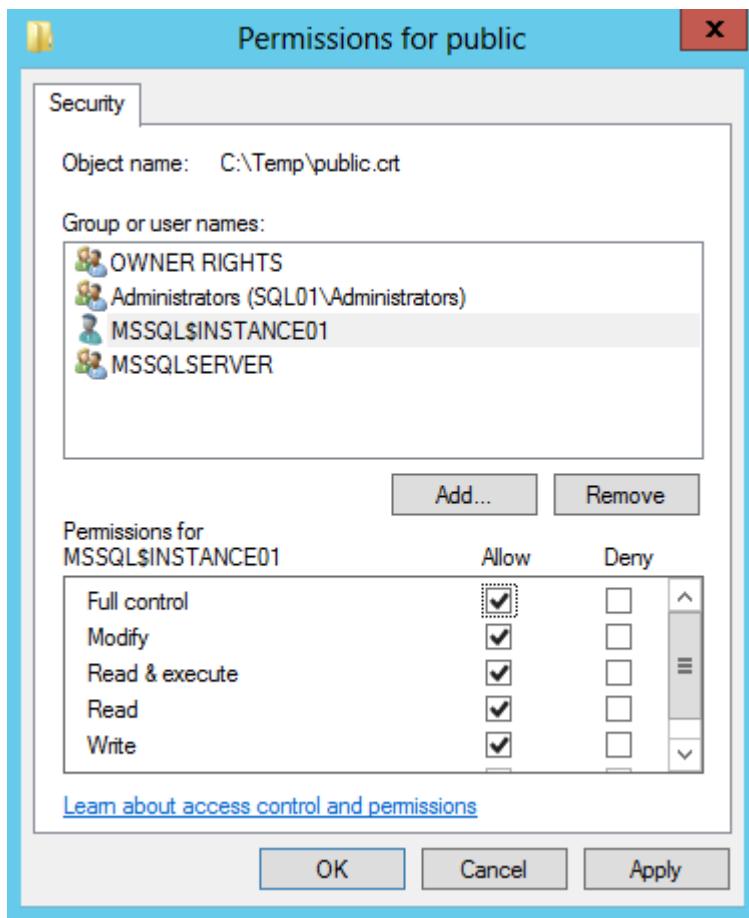
### Scenario

Your manager wants you to create a backup of the TDE protected database legitimately onto a different system. You decide to access the certificates and add them into the server instance before performing the database restore.

1. Grant permissions to the public certificate.
  - a) In **Windows Explorer**, navigate to the **C:\Temp** folder.
  - b) Right-click the **public.crt** file and select **Properties**.
  - c) In the **public Properties** dialog box, select the **Security** tab and select **Edit**.
  - d) In the **Permissions for public** dialog box, select **Add**.
  - e) In the **Select Users or Groups** dialog box, in the **Enter the object names to select** text box, type **NT Service\MSSQL\$INSTANCE01** and select **Check Names**.



- f) Select **OK** to return to the **Permissions for public.crt** dialog box.
- g) In the **Permissions for public.crt** dialog box, in the **Group or user names** section, select **MSSQL\$INSTANCE01**.
- h) In the **Permissions for MSSQL\$INSTANCE01** section, in the **Allow** column, check the check box against the **Full control** row to provide complete access to the **MSSQL\$INSTANCE01** user.



- i) Select **OK** twice to save the changes.
2. Grant access to the private certificate.
- a) In the **Temp** folder, right-click the **private.crt** file and from the fly-out pane, select **Properties**.
  - b) In the **private Properties** dialog box, select the **Security** tab and select **Edit**.
  - c) In the **Permissions for private** dialog box, select **Add**.
  - d) In the **Select Users or Groups** dialog box, in the **Enter the object names to select** text box, type **NT Service\MSSQL\$INSTANCE01** and then select the **Check Names** button.
  - e) Select **OK** to return back to the **Permissions for private** dialog box.
  - f) In the **Permissions for private** dialog box, in the **Group or user names** section, select **MSSQL \$INSTANCE01** and then select **OK**.
  - g) In the **Permissions for MSSQL\$INSTANCE01** section, in the **Allow** column, check the check box against the **Full control** row to provide complete access to the **MSSQL\$INSTANCE01** user.
  - h) Select **OK** twice to save the changes.
3. Import the public and private keys of the certificate.
- a) Return to the SSMS window.
  - b) Select the **New Query** button, in the **Query Editor** pane, type the CREATE MASTER KEY statement to create a key with password as 'pass@word1'.
- ```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'pass@word1';
```
- c) Execute the CREATE MASTER KEY statement to create the master key.
 - d) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.
 - e) Open a **Query Editor** pane, enter the CREATE CERTIFICATE keyword followed by the certificate name tdeCert, type the FROM FILE keyword followed by the path as 'c:\temp\public.crt' specify the location from where the public certificate needs to be imported.

```
CREATE CERTIFICATE tdeCert FROM FILE = 'c:\Temp\public.crt'
```

- f) Enter the **WITH PRIVATE KEY** keyword followed by the location to save the private certificate in the location 'c:\temp\private.crt'.

```
CREATE CERTIFICATE tdeCert FROM FILE = 'c:\Temp\public.crt'
WITH PRIVATE KEY (FILE = 'c:\temp\private.crt',
```

- g) Enter the **DECRYPTION BY PASSWORD** keyword followed by the password as 'pass@word1'.

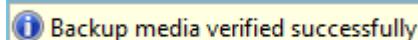
```
CREATE CERTIFICATE tdeCert FROM FILE = 'c:\Temp\public.crt'
WITH PRIVATE KEY (FILE = 'c:\temp\private.crt',
DECRYPTION BY PASSWORD = 'pass@word1');
```

- h) Execute the CREATE CERTIFICATE statement to import the public and private keys of the certificate.

- i) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.

4. Perform the restore operation.

- In the **Object Explorer** pane, right-click **Databases** folder and from the fly-out pane, select **Restore Database**.
- In the Restore Database window, in the **Source** section, select **Device** and select the **ellipsis (...)** button.
- In the **Select backup devices** dialog box, select the **Add** button.
- In the **Locate Backup File – SQL01\INSTANCE01** dialog box, in the **File name** text box, type **C:\Temp\SalesDB.bak** and select **OK** twice to return to the Restore Database window.
- In the Restore Database window, select the **Verify Media Backup** button, on the top of the window, observe that a message stating 'Backup media verified successfully' is displayed.



- In the Restore Database window, in the **Select a page** section, select the **Files** page.
- In the **Restore database files as** section, check the **Relocate all files to folder** check box.
- In the **Data file folder** text box, type **C:\Temp** to change the folder of the data file.
- In the **Log file folder** text box, type **C:\Temp** to change the folder of the log files.

Restore database files as			
<input checked="" type="checkbox"/> Relocate all files to folder			
Data file folder :		C:\Temp	
Log file folder :		C:\Temp	
Logical File Name	File Type	Original File Name	Restore As
SalesDB	Rows Data	C:\Program Files (x86)\Micros...	C:\Temp\SalesDB.mdf
Trans-01	Rows Data	C:\Drive1\Trans-01.ndf	C:\Temp\Trans-01.ndf
Trans-02	Rows Data	C:\Drive2\Trans-02.ndf	C:\Temp\Trans-02.ndf
LYTrans	Rows Data	C:\Drive2\LYTrans.ndf	C:\Temp\LYTrans.ndf
SalesDB_log	Log	C:\Program Files (x86)\Micros...	C:\Temp\SalesDB_log.ldf

- Select **OK** to start the restore process. Observe that a message box stating 'Database 'SalesDB' restored successfully' is displayed. Select **OK** to close the message box.
- In the SQL Editor toolbar, from the database drop-down list, select **SalesDB**.

- i) Open a **Query Editor** pane and type the SELECT statement to display all records from the SalesOrders table.

```
SELECT * FROM SalesOrders;
```

- m) Execute the SELECT statement.

- n) On the **Results** tab, observe that the records available in the SalesOrders table are displayed, ensuring that the SalesDB database is restored from its backup file. You can now view the results that are now being decrypted by the same certificate used earlier to encrypt them.

The screenshot shows the SQL Server Management Studio interface. A query window titled "SQLQuery1.sql - SQ...Administrator (53)*" contains the SQL command: "SELECT * FROM SalesOrders;". Below the query window is the "Results" tab of the execution pane, which displays the following data:

	SalesOrderID	SalesDate	ProductID	ListPrice	Quantity	TotalSalesAmount	CustomerID
1	1	2013-01-01	1	100.2	10	501	NULL
2	2	2013-01-01	1	29.95	3	89.85	NULL

- o) Close the **Query Editor** panes without saving the query.

TOPIC B

Compress a Database Table

You have applied TDE to a database to encrypt the database. When working with huge databases, there may be circumstances where your databases get overloaded with bulk data transactions, resulting in increase in database size. You can handle this by using data compression feature of SQL Server 2012. Familiarizing yourself with applying data compression on your databases will enable you to reduce the disk space required to store the data in your database objects. In this topic, you will compress a database table.

Data Compression

Data compression is the process of reducing the size of your database data. By enabling compression, you can save on storage space as well as increase I/O performance since the amount of data that needs to be read or written is lesser. However, this does affect performance at the CPU level, and therefore, care must be taken to balance the effects of using data compression. SQL Server 2012 enterprise edition is the only version that supports data compression within a database.

The data in a database can be compressed at two different levels; they are row-level compression and page-level compression. The *row-level compression* method is used to compress the storage of data within rows. The *page-level compression* is used to compress the storage of data on a page.

Data compression can be configured on the following:

- Heap tables
- Clustered index
- Non-clustered index
- Indexed views

You can even compress a table stored on different partitions with different compression schemes (row or page). Once the data is compressed, changing the compression scheme requires you to perform re-compression using the new method again. Depending on the size of the data in the table, this could take a considerable amount of time.



You may want to show **LearnTO Compress Database Tables** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOS for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



The System Stored Procedure



Note: For more information about data compression, check out the **LearnTO Compress Database Tables** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.

The `sp_estimate_data_compression_savings` System Stored Procedure

You can check the space savings using the `sp_estimate_data_compression_savings` system stored procedure. This stored procedure will display the following details of the database.

- Object name
- Index id
- Number of partitions
- Size with current compression setting
- Size with requested compression setting
- Sample size with current compression setting
- Sample size with requested compression setting

The syntax to check the space savings on the database is as follows.

```
sp_estimate_data_compression_savings
<schema>,
<object>,
<index>,
<partition>,
<compression type>
```

In this syntax, the index and partition parameters are optional and can be specified as NULL. The compression type parameter can be set to ROW, PAGE, or NONE.

The sp_estimate_data_compression_savings system stored procedure to compress the size of the SalesOrderDetail table.

	object_name	schema_name	index_id	partition_number	size_with
1	SalesOrderDetail	Sales	1	1	10040
2	SalesOrderDetail	Sales	2	1	3776
3	SalesOrderDetail	Sales	3	1	2336

Figure 7-2: Result set of the sp_estimate_data_compression_savings stored procedure.

Data Compression Options

You can compress a table during its creation using the DATA_COMPRESSION option. The syntax to create a compressed table is as follows.

```
CREATE TABLE <name>
( <columns> )
WITH (DATA_COMPRESSION = <ROW | PAGE | NONE>);
```

Alternatively, you can optimize an oversized table, by applying data compression to it. The syntax to apply data compression to a table is as follows.

```
ALTER TABLE <name>
REBUILD PARTITION = ALL
WITH (DATA_COMPRESSION = <ROW | PAGE | NONE>);
```

Example

The CREATE TABLE statement to create a compressed table is as follows.

```
CREATE TABLE employeeaddress
(Employee Name nvarchar(25), Employee Address nvarchar(25))
WITH (DATA_COMPRESSION = ROW);
```



Ask the students to list the circumstances in which row-level compression would be better than page-level compression.

This code snippet will create an employeeaddress table with two columns Employee Name and Employee Address. In addition, this will also apply row-level compression to the created table.

The ALTER TABLE statement to apply data compression to SalesLT.Product table is as follows.

```
ALTER TABLE SalesLT.Product  
REBUILD PARTITION = ALL  
WITH (DATA_COMPRESSION=PAGE);
```

This code snippet will alter the table named SalesLT.Product and applies page-level compression to compress the size of the data in this table.

You can perform a partition-by-partition compression with different compression settings. For instance, you might want to save space on previous years' transactions stored on a different partition by compressing it, but keep the current years partition uncompressed.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Apply Data Compression to a Table

ACTIVITY 7–4

Applying Data Compression to a Table

Before You Begin

1. In the **Object Explorer** pane, right-click **SQL01\INSTANCE01**→**Disconnect**.
2. In the **Object Explorer** pane, select **Connect**→**Database Engine**.
3. In the **Connect to Server** dialog box, from the **Server name** drop-down list, select **SQL01** and **Connect**.

Scenario

Your database manager feels that the SalesOrderDetail table takes up larger storage size in the database and he wants the table to be optimized using data compression techniques. You are asked to analyze the actual savings in disk space as well as the best compression method for compressing the SalesOrderDetail table.

1. View the storage space of the SalesOrderDetail table.
 - a) In the **Object Explorer** pane, expand **SQL01**→**Databases**→**AdventureWorks2012** to view the **AdventureWorks2012** database node.
 - b) Expand the **Tables** folder, scroll down and select **Sales.SalesOrderDetail** to select the **SalesOrderDetail** table.
 - c) Right-click the **SalesOrderDetail** table and select **Properties**.
 - d) In the Table Properties window, in the **Select a page** section, select **Storage**.
 - e) On the **Storage** page, in the **Compression** section, verify that the **Compression type** value is set to **None**.
 - f) On the **Storage** page, in the **General** section, observe the **Data space** and **Index space** values.

Compression	
Compression type	None
Filegroups	
FILESTREAM filegroup	
Filegroup	PRIMARY
Text filegroup	
Table is partitioned	False
General	
Vardecimal storage format is enabled	False
Data space	9.664 MB
Index space	6.109 MB
Row count	121317

- g) Select **Cancel** to close the Table Properties – SalesOrderDetail window.
2. Determine the compression type.
 - a) Open a **Query Editor** pane, and type the space saving stored procedure `sp_estimate_data_compression_savings` followed by the database schema name **Sales**.
`sp_estimate_data_compression_savings 'Sales',`
 - b) Type the table name as **SalesOrderDetail** and then set the index and partition values as null, and then assign the compression type as row.
`sp_estimate_data_compression_savings 'Sales', 'SalesOrderDetail', null, null, 'ROW';`

- c) Execute the stored procedure.
- d) On the **Results** tab, observe the size of the table using the current and requested compression settings.

The screenshot shows a SQL query window titled "SQLQuery2.sql - SQ...Administrator (53)*". The query executed is:

```
sp_estimate_data_compression_savings 'Sales',
'SalesOrderDetail', null, null, 'ROW';
```

The results pane displays a table with the following data:

	object_name	schema_name	index_id	partition_number	size_with_current
1	SalesOrderDetail	Sales	1	1	10040
2	SalesOrderDetail	Sales	2	1	3776
3	SalesOrderDetail	Sales	3	1	2336

- e) In the **Query Editor** pane, change the compression type parameter in the stored procedure to Page.

```
sp_estimate_data_compression_savings 'Sales', 'SalesOrderDetail', null,
null, 'PAGE';
```

- f) Execute the stored procedure to compress the SalesOrderDetail table.

- g) On the **Results** tab, observe that using the PAGE compression gives significantly better compression performance than row-level compression.

The screenshot shows two windows side-by-side. The left window is titled "SQLQuery17.sql - S...Administrator (68)*" and the right window is titled "SQLQuery15.sql - S...Administrator (7)". Both windows show the same query:

```
sp_estimate_data_compression_savings
'Sales', 'SalesOrderDetail', null, null, 'PAGE';
```

The results pane displays a table with the following data:

	object_name	schema_name	index_id	partition_number	size_with_current
1	SalesOrderDetail	Sales	2	1	3776
2	SalesOrderDetail	Sales	1	1	4768
3	SalesOrderDetail	Sales	3	1	2336

3. Compress the size of the SalesOrderDetail table using page-level compression.

- a) Open a **Query Editor** pane, enter the ALTER TABLE statement followed by the schema name Sales and the table name SalesOrderDetail.

```
ALTER TABLE Sales.SalesOrderDetail
```

- b) Enter the REBUILD PARTITION keyword with ALL keyword, to compress all components of the SalesOrderDetail table.

```
ALTER TABLE Sales.SalesOrderDetail
REBUILD PARTITION = ALL
```

- c) Enter the WITH keyword followed by the data compression type as PAGE.

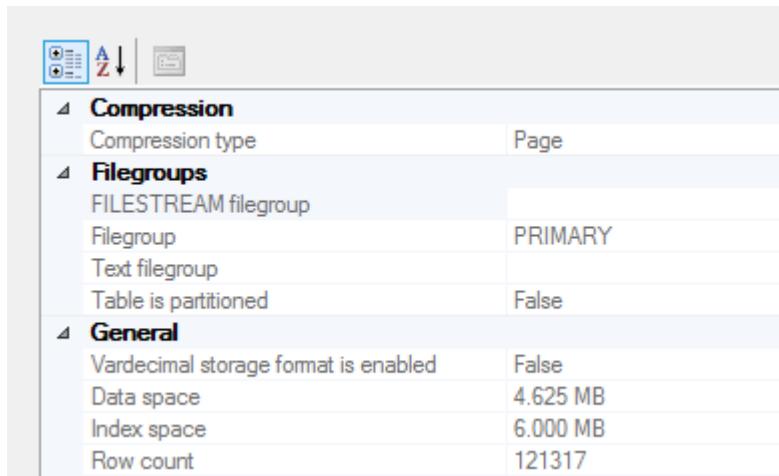
```
ALTER TABLE Sales.SalesOrderDetail
REBUILD PARTITION = ALL
WITH (DATA_COMPRESSION=PAGE);
```

- d) Execute the ALTER TABLE statement to compress the SalesOrderDetail table.

- e) On the **Messages** tab, observe that the message stating 'Command(s) completed successfully.' is displayed.

4. View the size of the compressed SalesOrderDetail table.

- In the **Object Explorer** pane, right-click the **Sales.SalesOrderDetail** table and from the fly-out pane, select **Properties**.
- In the Table Properties window, in the **Select a page** section, select **Storage**.
- In the right pane, observe that **Compression** is set to **Page** and the **Data Space** is now compressed to **4.625 MB**, which is comparatively lower than the earlier data space of 9.664 MB.



- In the Table Properties window, observe that the index space has not decreased. This is because non-clustered indexes do not inherit compression settings; you need to manually rebuild them with compression.
- Select **OK** to close the Table Properties – SalesOrderDetail window.
- Close the **Query Editor** panes without saving the query.

Summary

In this lesson, you applied encryption and compression to your databases. Encrypting the data in the database will enable you to protect data against unauthorized access and data theft. In addition, data compression will ensure that your database disk space is optimized.



Use the review questions provided to generate discussion among the participants.

When should TDE be used as a security measure on databases?

A: Answers will vary, but may include: to protect data from being accessed by unauthorized users, to prevent stolen data from being read by other users, and to protect the confidential information in the database.

What is the disadvantage of using encrypted keys and certificate using data transactions?

A: Answers will vary, but may include: the encrypted data may become unrecoverable if the keys or certificates get corrupt or lost. It is always recommended to backup your keys and certificates after every valuable data transactions.



Note: Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

8

Working with Indexes and Log Files

Lesson Time: 2 hours, 45 minutes

Lesson Objectives

In this lesson, you will:

- Work with full text indexes.
- Configure advanced storage and indexing settings
- Configure log files.

Lesson Introduction

You have configured SQL Server® by installing and migrating the required settings to SQL Server 2012. When working with large volume of organizational data, you need to ensure that the critical information's are easily accessible. You can maintain data accessibility by managing the database server instance using the various tools, such as auto shrink, auto close and recovery models, available in SQL Server 2012. Familiarizing yourself with indexes will enable you to efficiently administer your organizational data. In this lesson, you will work with indexes and log files.

TOPIC A

Work with Full Text Indexes

When working with organizational data, there may be circumstances where you may need to search for a particular data on the database. Exploring the database to search for a specific data is a time-consuming and tedious task. SQL Server 2012 provides you a search feature named full text indexes to search for words or phrases in tables quickly which helps speed up textual searches significantly. In this topic you will, work with full text indexes.



Indexes

Indexes allow the database to find the required files in the table without searching the entire table. SQL Server has many different types of indexes namely clustered, nonclustered, spatial, and XML. Most of these indexes work in a standard B or B+ Tree structure that allows searches or queries to quickly scan the indexed column based on “blocks” or “pages” of information.

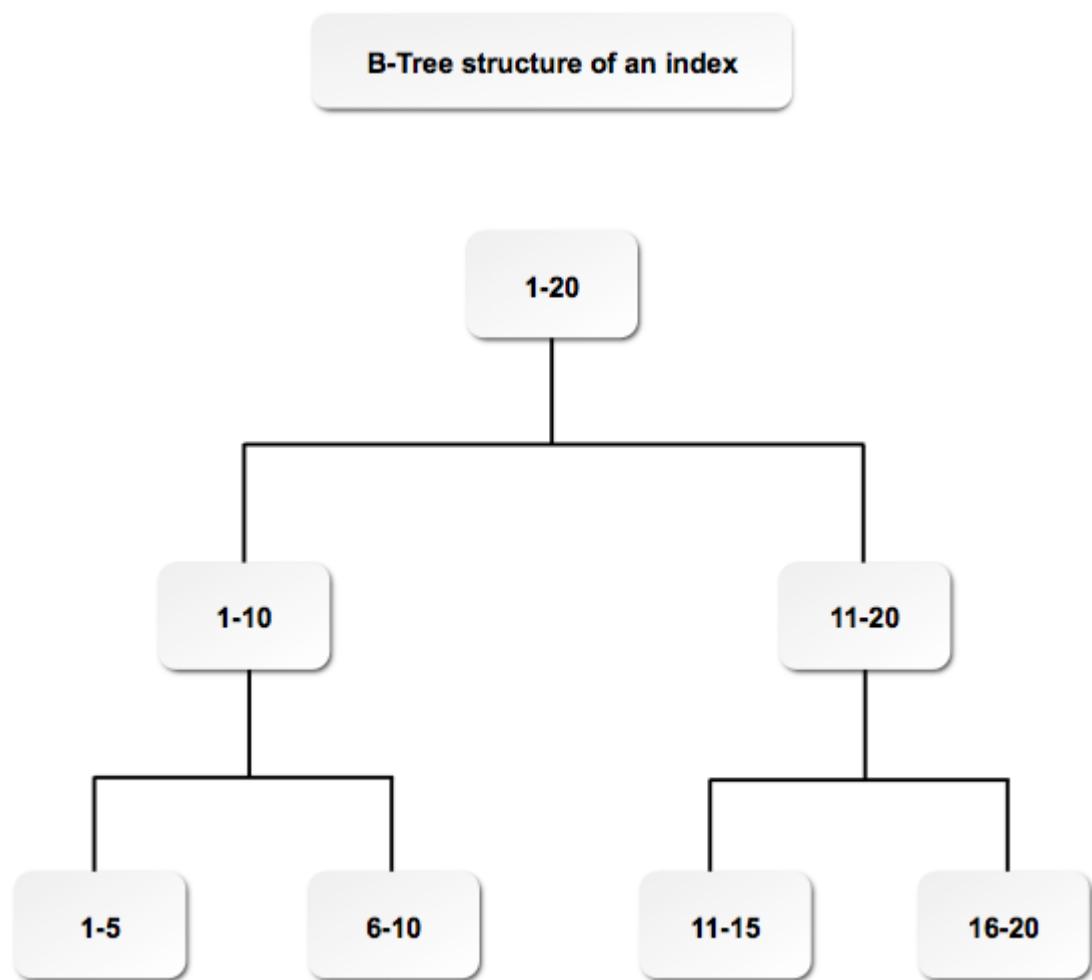


Figure 8-1: Indexing structure for a table with 20 rows.

However, when you wish to search for a word or phrase within a column containing a large volume of character data, such as TEXT or VARCHAR(MAX) formats, it is noted that TEXT or

VARCHAR(MAX) formats are not supported, instead the engine does a complete table scan to find the required set of words or phrases.

This type of searching will also fail if you search for a specific form of a word. For instance, you might want to search for all forms of the word “eat”, including “eating,” “ate,” and “eaten.” The normal search query will however miss this type of search. You can perform such granular check of data using the Full Text Search and Full Text Indexing methods.

FTI

Full Text Indexing (FTI) allows you to search for a particular word or phrases in a data table. The FTI feature allows you to create a token of all non-noise words and maintains their positions in terms of the rows that contain those words. *Noise words* are defined as the common set of words that really don't need to be ever searched or used—for instance “the,” “and,” “or,” and “not.”

Full Text Indexing is created under a grouped set called a *Full Text Catalog* where you first need to create one or more catalogs in a database and add Full Text Indexes in it. In addition, each table in a database can have only one FTI with additional parameters in it.

For instance, in a database if a row contains the phrase “The dog went home,” then FTI will contain the three words “dog,” “went,” and “home” with each of them pointing to that particular row. If another row contains the sentence “A dog ate the cat” the words that would reference in the database will be “dog,” “ate,” and “cat.” The word “dog” now would point to both the original row as well as the second example row. Therefore, if the word “dog” is ever searched in the database, the FTI will immediately return the two rows that contain it.



Note: For more information about FTI, check out the **LearnTO Search with Full Text Indexing** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.

Features of FTI

FTI has a number of options and methods of utilization. When you create an index, you can decide whether you wish to allow changes made to the table columns to be automatically updated in the FTI or whether you will do it manually. While tracking automatically does sound good, it really depends upon the amount of data in the table column that you have. If there are constant changes occurring, you might be better off performing the re-population or updates on a schedule—which is another option available for this type of index.

Another great feature of FTI is the ability to index content within standard document types such as Word documents, Excel spreadsheets, PowerPoint presentations, and Adobe PDFs, which are stored in a column in each row. All you need is a different column in the table that contains the file extension for the file type in that row. Simply specify this column along with the column that contains the file and SQL Server will allow you to search within those documents as well.

FTI also uses thesauruses and discarded strings called *stop words* which are supplied by the system for filtering out words and their forms. The stop words are used to prevent the FTI from over loading with unnecessary word searches. The FTI also provides you the ability to manage a custom list for thesaurus words as well as stop words to filter out from the index.



You may want to show **LearnTO Search with Full Text Indexing** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.

Full Text Querying

Full text querying is used when you want to perform linguistic searches on data of the full-text indexes. You can use either the CONTAINS or FREETEXT clauses in the WHERE clause of a SELECT statement to find the required text from a database. Both these full text querying clauses allows you to perform advanced database querying.

There are a few terms that need to be noted while using Full Text Querying. Some of these terms are listed in the table.

Terms	Description
Simple Terms	These terms use a word or the exact phrase to search. This search is similar to using a normal query with an “=” or “LIKE” clause.
Prefix Terms	These terms use a word or phrase that begins with the specified term(s).
Generation Terms	These terms are the inflected forms of the specified word.
Proximity Terms	These terms are used where the words are not together as a phrase but are “close” to each other.
Thesaurus Terms	These terms use synonyms for a specified word.
Weighted Terms	These terms are used where each word or phrase has a weight assigned for searching.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Work with Full Text Indexes

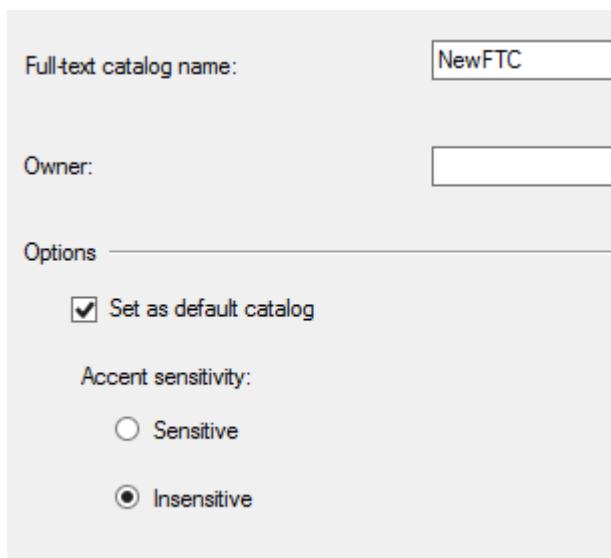
ACTIVITY 8–1

Setting Up Full Text Indexing

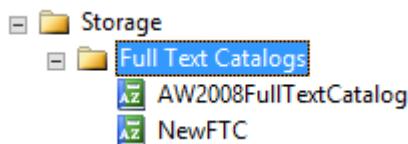
Scenario

The Sales team of Adventure Works Cycles wants to allow customers to search their products not only by name but also by any word in the description. You decide to implement FTI on the product and description tables of the database.

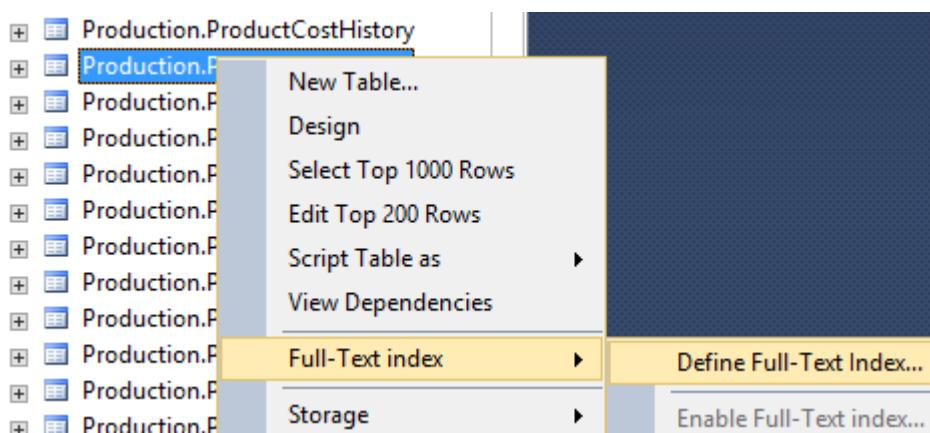
1. Set up a Full-Text Catalog.
 - a) In the **Object Explorer** pane, in the **Databases** folder, expand **AdventureWorks2012→Storage**.
 - b) In the **Storage** folder, right-click on the **Full Text Catalogs** node, and select **New Full-Text Catalog**.
 - c) In the New Full-Text Catalog - AdventureWorks2012 window, in the **Full-text catalog name** text box type **NewFTC** and in the **Options** section, check the **Set as default catalog** check box.
 - d) In the **Accent sensitivity** section, select the **Inensitive** option to ignore the case and accent differences.



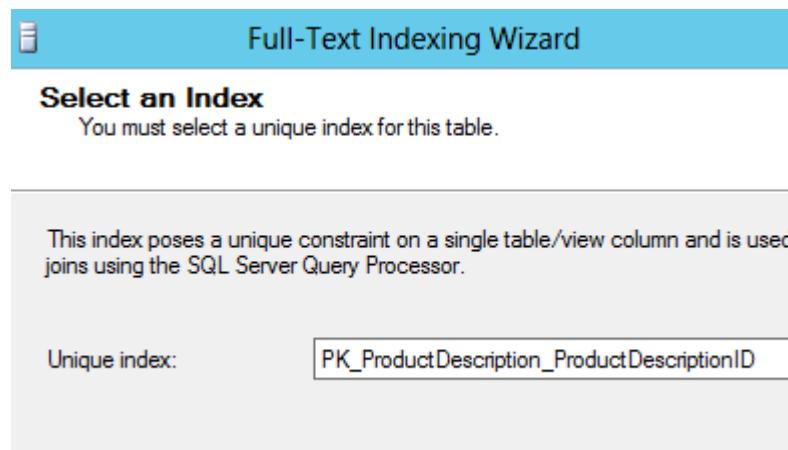
- e) In the New Full-Text Catalog-AdventureWorks2012 window, select **OK** to save the full text catalog.
- f) In the **Object Explorer** pane, expand **Full Text Catalog** and observe that the **NewFTC** is displayed.



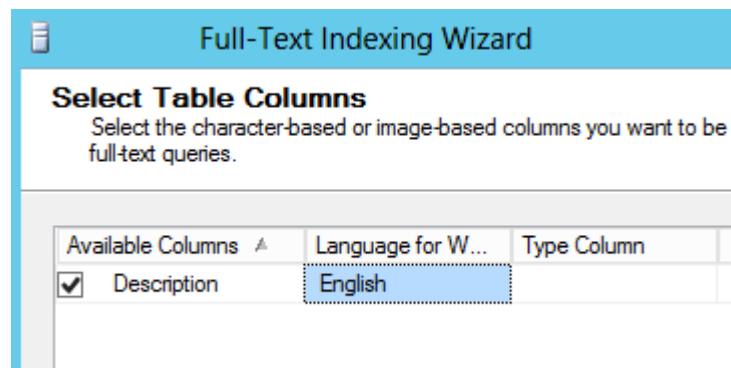
2. Set up a Full-Text Index.
 - a) In the **Object Explorer** pane, expand the **Tables** folder.
 - b) In the **Tables** folder, right-click on the **Production.ProductDescription** table, from the fly-out pane, select **Full-Text index→Define Full Text Index**.



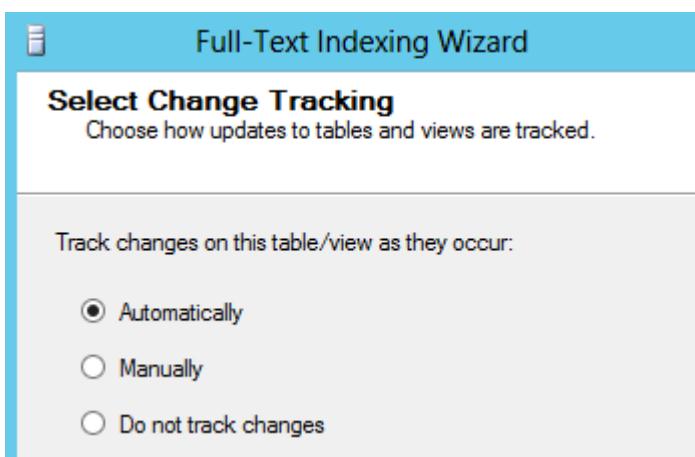
- c) In the **Full-Text Indexing Wizard**, select **Next** to proceed with the installation process.
- d) On the **Select an Index** page, in the **Unique Index drop-down** list ensure that **PK_ProductDescription_ProductDescriptionID** is selected, and select **Next** to proceed with the installation process.



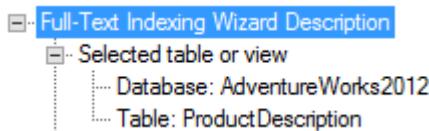
- e) On the **Select table Columns** page, in the **Available Columns** list, check the **Description** check box and in the **Language for Word Breaker** drop-down list, select **English** to identify the appropriate word breakers for the index.



- f) On the **Select Table Columns** page, select **Next** to proceed with the installation process.
- g) On the **Select Change Tracking** page, in the **Track changes on the table/view as they occur** section, verify that **Automatically** option is selected.



- h) On the **Select Change Tracking** page, select **Next** to proceed with the installation process.
- i) On the **Select Catalog, Index Filegroup, and Stoplist** page, in the **Select full-text catalog** drop-down list verify “**NewFTC**” is selected as the default catalog in this database.
- j) In the **Select index filegroup** drop-down list, ensure that **<default>** is selected and, in the **Select full-text stoplist** drop-down list ensure that **<system>** is selected.
- k) On the **Select Catalog, Index Filegroup, and Stoplist** page, select **Next** to proceed with the installation process.
- l) On the **Define Population Schedules (Optional)** page, select **Next** to skip the population schedule screen and proceed with the installation process.
- m) On the **Full-Text Indexing Wizard Description** page, in the **Full-Text Indexing Wizard Description** node, expand the **Selected table or view** node and ensure that the **AdventureWorks2012** and **ProductDescription** are selected.



- n) On the **Full-Text Indexing Wizard Description** page, select **Finish** to create the index and populate it for the first time.
 - o) In the **Full-Text Indexing Wizard Progress** page, observe that you have successfully created the FTI on the **AdventureWorks2012** database. Select **Close** to complete the installation process.
3. Display the products that contain “mountain” in their name from the **Production.ProductDescription** table.
- a) Open a **Query Editor** pane and type the **SELECT** statement to display all records from the **Production.ProductDescription** table.
- ```

SELECT *
FROM Production.ProductDescription

```
- b) Enter a **WHERE** clause along with a **CONTAINS** keyword to search for the term “mountain” in the **Description** column of the **Production.ProductDescription** table.
- ```

SELECT *
FROM Production.ProductDescription
WHERE CONTAINS (Description, 'mountain');
  
```
- c) Select **Execute** to execute the **SELECT** statement and display the products that matches the given criteria.
 - d) On the **Results** tab, observe that the list of records appear from the **Production.ProductDescription** table that contain the word “mountain” in them.

SQLQuery5.sql - SQ...Administrator (64)* X

```
SELECT *
FROM Production.ProductDescription
WHERE CONTAINS>Description, 'mountain');
```

100 % < Results Messages

Prod.	Description
1 128	Serious back-country riding. Perfect for all levels of competition. Uses the same HL Frame as the Mountain
2 168	Top-of-the-line competition mountain bike. Performance-enhancing options include the innovative HL Fram
3 647	Each frame is hand-crafted in optimum diameter and wall-thickness required of a premium mountain frame. I
4 686	Replacement mountain wheel for entry-level rider.
5 687	Replacement mountain wheel for the casual to serious rider.
6 688	High-performance mountain replacement wheel.
7 867	Replacement rear mountain wheel for entry-level rider.

4. Display the products that contain "mountain or wheel" in their name from the Production.ProductDescription table.
 - a) Modify the WHERE clause along with a CONTAINS keyword to search for the term "mountain" or "wheel" in the Description column of the Production.ProductDescription table.

```
SELECT *
FROM Production.ProductDescription
WHERE CONTAINS>Description, 'mountain or wheel');
```

- b) Select **Execute** to execute the SELECT statement and search and display the products that matches the given criteria.
 - c) On the **Results** tab, observe that the list of records appear from the Production.ProductDescription table that contain either the word "mountain" or "wheel" in them.

SQLQuery6.sql - SQ...Administrator (69)* X

```
SELECT *
FROM Production.ProductDescription
WHERE CONTAINS>Description, 'mountain or wheel');
```

100 % < Results Messages

Pro...	Description
1 128	Serious back-country riding. Perfect for all levels of competition. Uses the same HL Frame as the Mountain-
2 168	Top-of-the-line competition mountain bike. Performance-enhancing options include the innovative HL Fram
3 647	Each frame is hand-crafted in optimum diameter and wall-thickness required of a premium mountain frame re
4 686	Replacement mountain wheel for entry-level rider.
5 687	Replacement mountain wheel for the casual to serious rider.
6 688	High-performance mountain replacement wheel.
7 689	Replacement road front wheel for entry-level cyclist.

5. Display the products that contain the keywords "mountain" and "frame" with a distance of words being "2" from the Production.ProductDescription table.

- a) Modify the WHERE clause along with a CONTAINS keyword to search for the term “mountain” and “frame” with the distance between the words being “2” in the Description column of the Production.ProductDescription table.

```
SELECT *
FROM Production.ProductDescription
WHERE CONTAINS>Description, 'NEAR((mountain, frame), 2)')
```

- b) Select **Execute** to execute the SELECT statement to display the products that matches the given criteria.
c) On the **Results** tab, observe that the rows that are affected will appear from the Production.ProductDescription table that contain the words “mountain” and “frame” in them with the distance of the words being two, i.e. in the sentence, there are just two words that separate the words “mountain” and “frame”.

```
SQLQuery7.sql - SQ...Administrator (71)*
SELECT *
FROM Production.ProductDescription
WHERE CONTAINS>Description, 'NEAR((mountain, frame), 2)')

100 % < Results Messages
Pro... Description
1 128 Serious back-country riding. Uses the same HL Frame as the Mountain-10
2 647 Each frame is hand-crafted in the required of a premium mountain frame to t
```

6. Display the products that contain the keywords “mountain” and “frame” with a distance of words being “6” from the Production.ProductDescription table.

- a) Modify the WHERE clause along with a CONTAINS keyword to search for the term “mountain” and “frame” with the distance between the words being “6” in the Description column of the Production.ProductDescription table.

```
SELECT *
FROM Production.ProductDescription
WHERE CONTAINS>Description, 'NEAR((mountain, frame), 6)')
```

- b) Execute the SELECT statement to display the products that matches the given criteria.
c) On the **Results** tab, observe that the rows that are affected will appear from the Production.ProductDescription table that contain the words “mountain” and “frame” in them with the distance of the words being six, i.e. in the sentence, there are six words that separate the words “mountain” and “frame”.

The screenshot shows the SSMS interface with a query window titled "SQLQuery8.sql - SQ...Administrator (72)*". The query is:

```
SELECT *
FROM Production.ProductDescription
WHERE CONTAINS>Description, 'NEAR((mountain, frame), 6)'
```

The results pane displays four rows of data:

	Description
1	riding. Perfect for all levels of competition. Uses the same HL Frame as the Mountain-100.
2	Bothell facility to the optimum diameter and wall thickness required of a premium mountain frame. The
3	المستخدم في طراز HL Frame لغاية المحترفين في المناطق الرئيسية. دراجة مثلى لجميع مت
4	זהה לו המצויה בדגם HL מסגרת HL Frame (1). מתחייבים בכל רמה של חזרות.

- d) Close the **Query Editor** pane without saving the query.

ACTIVITY 8-2

Configuring Full Text Indexing

Scenario

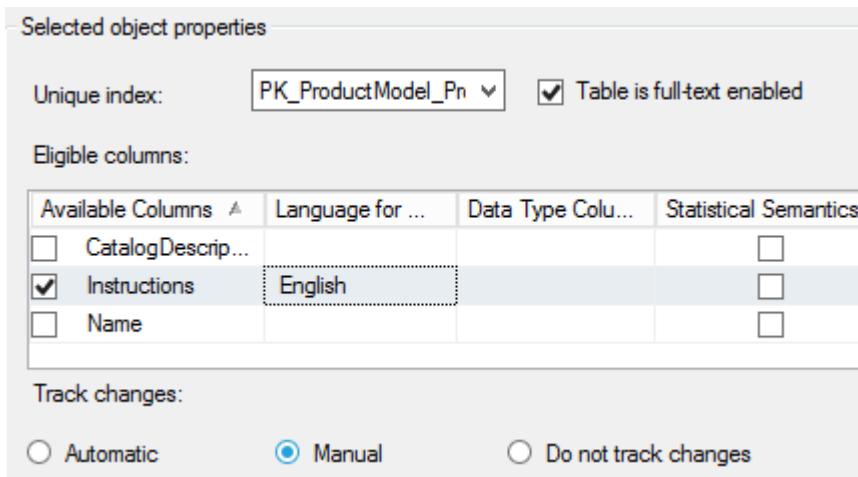
As database administrator of Adventure Works Cycles, you want to enable quick searches of instructions for using the product in the database, it is required to add more tables and columns to the product catalog. In order to perform this task, you will need to configure the catalog with a new index and rebuild it.

1. Configure FTI on the AdventureWorks2012 database.
 - a) In the **Object Explorer** pane, right-click **SQL01** and from the fly-out pane and select **Refresh**.
 - b) In the **Databases** folder, expand **AdventureWorks2012** and then expand **Storage→Full Text Catalogs**.
 - c) In the **Full Text Catalogs** node, right-click on **NewFTC** and from the fly-out pane, select **Properties**.
 - d) In the Full-Text Catalog Properties window, on the **Select a page** section, select **Tables/Views** page.
 - e) In the **All eligible table/view objects in this database** section, in the **Object Name** list box scroll down to find the **Production.ProductModel** table and select it.
 - f) Select the button to move the **Production.ProductModel** table to the **Table/View objects assigned to the catalog** list on the right.
 - g) In the **Table/View objects assigned to the catalog** list, select **Production.ProductModel**.

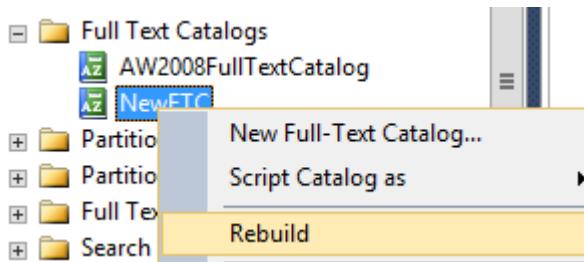


Note: In the **Select Object Properties** pane, in the “Unique index” option drop-down list, ensure that the default column index **PK_ProductDescription_ProductDescriptionID** is selected.

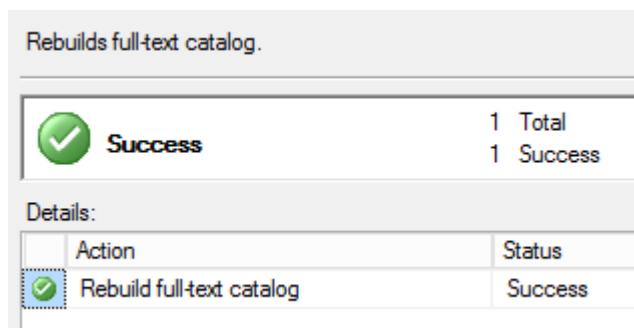
- h) In the **Eligible columns** section, in the list of available columns, select the **Instructions** column and in the **Language for Word Breaker** drop-down list, select **English** as the language to identify the appropriate word breakers in the index.
- i) In the **Track changes** section, select the **Manual** option. Select **OK** to save the settings that were configured in the **NewFTC** tables/views option of the catalog.



- j) In the **Full-Text Catalogs** folder, right-click **NewFTC** and from the fly-out pane, select **Rebuild** to restructure the full text catalog.



- k) In the **Rebuild Full-Text Catalog** dialog box, if you are prompted to delete and rebuild the full-text catalog, select **OK** to rebuild the catalog.
- l) In the **Rebuild Full-Text Catalog** dialog box, verify that the rebuild was success. Select **Close** to complete the rebuild.



2. Which format is not supported while searching for a column containing a large volume of character data?
 - VARCHAR(MAX)
 - NVARCHAR
 - VARCHAR
 - NCHAR
3. Which of the following are referred to as noise words? (Choose three.)
 - The
 - An
 - Or
 - And
 - A

TOPIC B

Configure Advanced Storage and Indexing Settings

You have configured FTI on a specific table. When working with organizational data, you may be dumped with a huge volume of data and as database administrator it is your prime duty to search and obtain the required information at any point in time. The advanced storage and indexing settings of SQL Server allows you to manage and control your organizational data. In this topic, you will configure advanced storage and indexing settings.

Column Store Indexes

Column store indexes store column data for the columns mentioned in the index, instead of storing rows of data in the index. These indexes compress the data significantly. Each page in the index simply stores a different column if they are specified.

In traditional database systems, data is stored in rows in the table and when a query comes in, the index and the row data work together to retrieve and present the values. However, in data warehousing and analysis scenarios, this is not the way that it is done.

In Data Warehouse and Analysis scenarios, usually a summarization of a fixed set of columns is required. For instance, in a sales table, users might want to know the sum or average of sales over a week, month or year. Each time this query is run, in normal cases, indexes will work across the rows that are stored as pages of data in the table and retrieve it and calculate the summarization. This is unfortunately a performance hit on the database engine.



Column Store Indexes

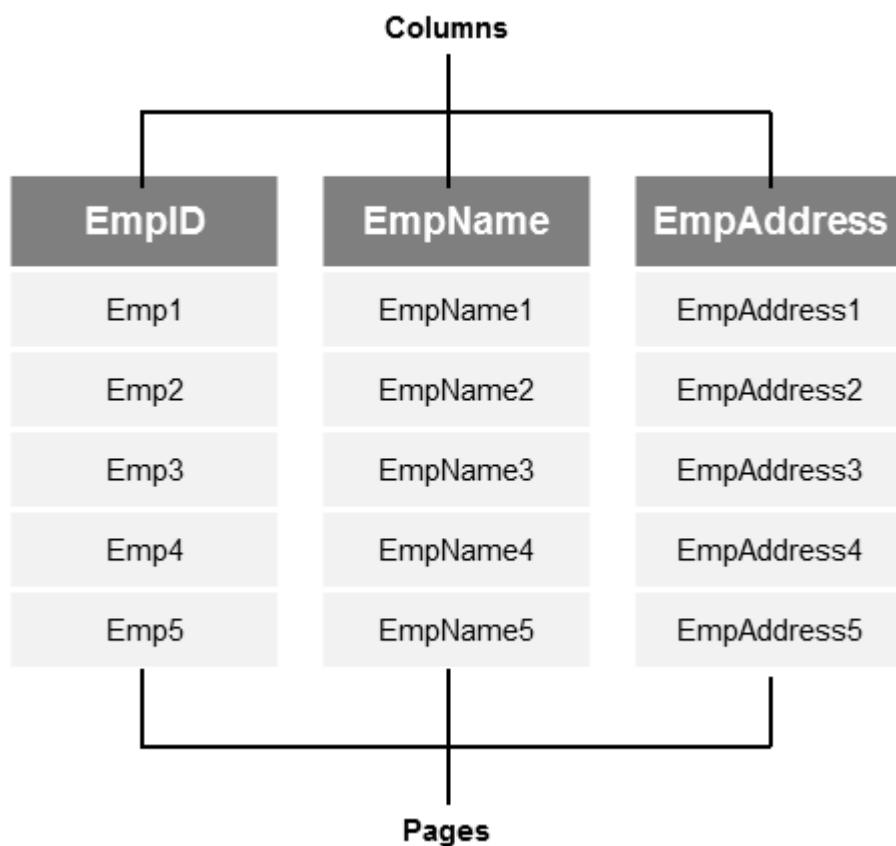


Figure 8–2: A data table with column store index.

Advantages and Disadvantages of Using a Column Store Index

The advantage of using the column store indexes is that only columns required in a query will be indexed, whereas the rest of the table components will be simply ignored. Tons of performance optimizations have been done on the SQL Server 2012 indexes feature so that CPU usage is minimal.

However, the one important restriction is that tables with column store indexes cannot be updated in SQL Server 2012, which means you can select and insert data but cannot update existing data as that would require the entire column to be rebuilt. In order to do this, you need to drop the index, perform the update and then recreate the index again if required. Once the column store index is created, it works and feels just like a normal index to the end user. No change in any data retrieval code is required.

FILESTREAM



FILESTREAM objects are file objects that are stored on the file system but referenced within a special VARBINARY(MAX) column with the FILESTREAM option. Standard FTI will work on this FILESTREAM column as well, even though the contents of the file are actually not within the database. SQL Server 2012 has the ability to read the contents of different types of document files directly.

Example

A SELECT statement to display the list of file types supported by SQL Server is as follows.

```
SELECT * FROM sys.fulltext_document_types;
```

The sys.fulltext_document_types keyword in the SELECT statement will display the list of all file extensions that the SQL Server engine can directly parse and perform FTI on. The result set will

include large number of file types, including Microsoft Word, Excel, and PowerPoint, but only of the older 2003 versions with extensions as .doc, .xls, and .ppt, respectively.

The screenshot shows a SQL query window titled "SQLQuery11.sql - S...Administrator (54)*". The query is:

```
SELECT * FROM sys.fulltext_document_types;
```

The results pane displays a table with five rows, showing document types and their corresponding class IDs and paths:

	document_type	class_id	path
1	.ascx	E0CA5340-4534-11CF-B952-00AA0051FE20	C:\Program Files\Microsoft SQL Server
2	.asm	C7310720-AC80-11D1-8DF3-00C04FB6EF4F	C:\Program Files\Microsoft SQL Server
3	.asp	E0CA5340-4534-11CF-B952-00AA0051FE20	C:\Program Files\Microsoft SQL Server
4	.aspx	E0CA5340-4534-11CF-B952-00AA0051FE20	C:\Program Files\Microsoft SQL Server
5	.bat	C7310720-AC80-11D1-8DF3-00C04FB6EF4F	C:\Program Files\Microsoft SQL Server

Figure 8-3: List of document types with extensions displayed by using the sys.fulltext_document_types command.

You can add new file extensions that it can parse by including the appropriate iFilters in the system and registering them with SQL Server. This includes newer Office documents, Adobe PDF, and other types that you might want. In order to include an appropriate iFilter within SQL Server, you have to contact the developer of the file type for storing such files and to be able to use Full Text Search on them as well.

File Table

FileTable is a new feature in SQL Server 2012 that allows administrators to point to a standard file system folder such as the hard disk or *Universal Naming Convention (UNC)* path. In addition, it allows SQL Server to automatically populate the table with information about the files and directories within them. You can add and change files both within SQL Server or directly on the file system and have the changes immediately reflect in the table as well.

The advantage of using a FileTable is that you can simply configure a FileTable and point to a location and SQL Server does all the work of adding and maintaining the list of files there. You can also have other client applications directly to read and write the files in the folder. And finally, it becomes a simple task to add FTI on the folder and its contents enabling quick and powerful query capabilities with the contents of the files in that folder through SQL Server.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Configure Advanced Storage and Indexing Settings

ACTIVITY 8-3

Configuring Column Store

Before You Begin

Ensure that **AdventureWorks2012** database is selected.

Scenario

A senior manager in your team feels that the performance of a SalesOrderDetail table with non-clustered index is very low when compared with other tables in the databases. You are asked to perform a data analysis in that table. You decide to configure a columnstore index in the SalesOrderDetail table and evaluate its performance.

1. Create a copy of a table to perform analysis.

- a) Open a **Query Editor** pane and enter the SELECT INTO statement followed by the destination table name as Sales.SalesOrderDetailForAnalysis.

```
SELECT * INTO Sales.SalesOrderDetailForAnalysis
```

- b) Type the FROM clause followed by the source table name as Sales.SalesOrderDetail.

```
SELECT * INTO Sales.SalesOrderDetailForAnalysis FROM
Sales.SalesOrderDetail;
```

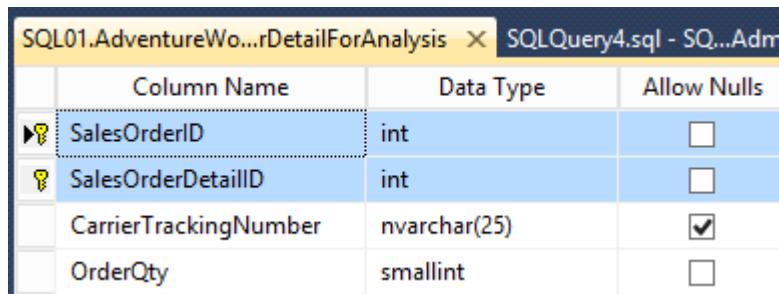
- c) Execute the SELECT INTO statement to create a copy of the Sales.SalesOrderDetail table.

- d) On the Messages tab, observe that the message "(121317 rows(s) affected)" is displayed.

- e) Close the **Query Editor** pane without saving the query.

2. Create and verify the Primary Key for the column store index.

- a) In the **Object Explorer** pane, refresh the **AdventureWorks2012** database and expand the **Tables** folder and observe that the newly created table **Sales.SalesOrderDetailForAnalysis** is displayed.
- b) In the **Tables** folder, right-click the **Sales.SalesOrderDetailForAnalysis** table and select **Design**.
- c) In the **Design** pane, in the **Column Name** list, right-click **SalesOrderID** and from the fly-out pane select **Set Primary Key**. Right-click **SalesOrderDetailID** and from the fly-out pane select **Set Primary Key** to set the Column Store index for both the columns.



Column Name	Data Type	Allow Nulls
SalesOrderID	int	<input type="checkbox"/>
SalesOrderDetailID	int	<input type="checkbox"/>
CarrierTrackingNumber	nvarchar(25)	<input checked="" type="checkbox"/>
OrderQty	smallint	<input type="checkbox"/>

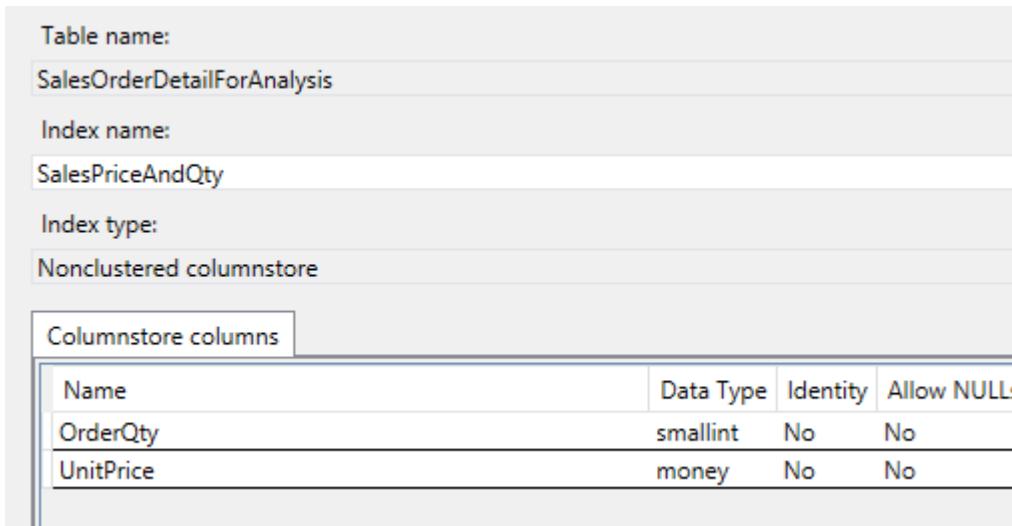
- d) On the **SQL Editor** toolbar, select the **Save** button to save the design. Once the design is saved, close the design window.



Note: To save the design you can also press **Ctrl+S** on the keyboard.

3. Create a new Column Store Index.

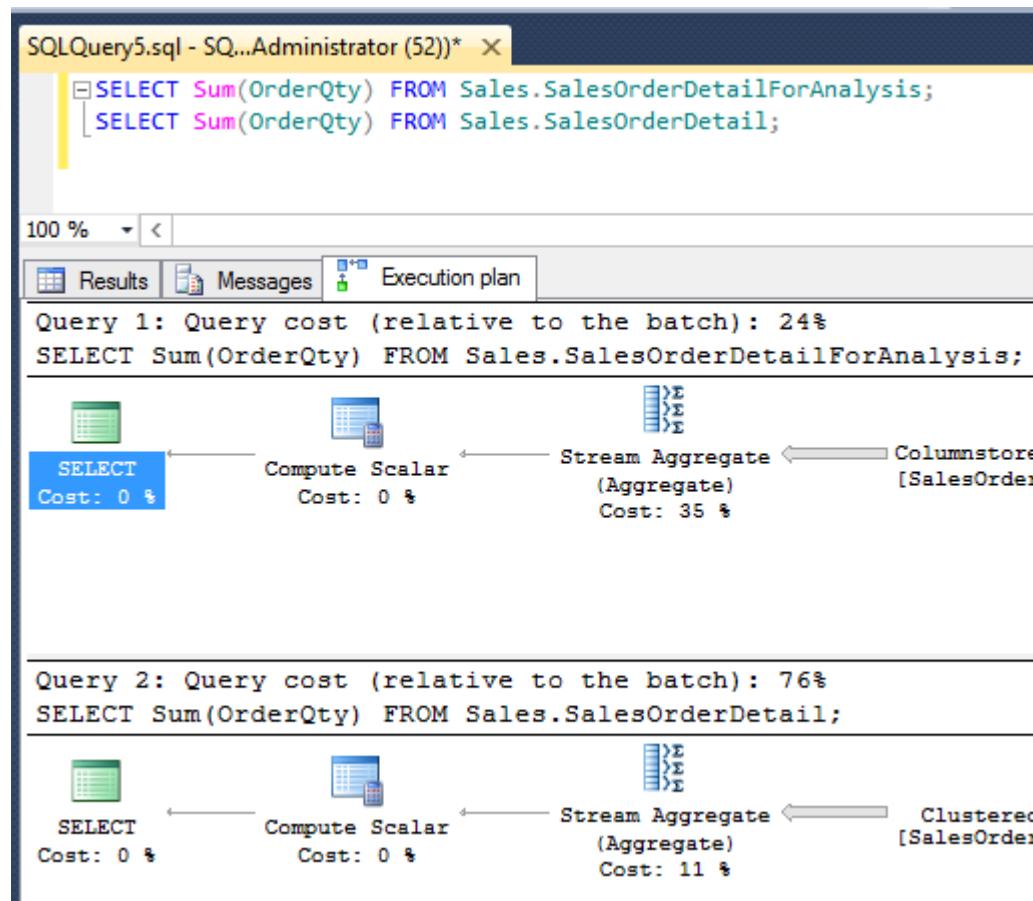
- In the **Object Explorer** pane, expand **Sales.SalesOrderDetailForAnalysis** table and right-click the **Indexes** node, from the fly-out pane select **New-Index →Non-Clustered Columnstore Index**.
- In the New Index window, in the **Table name** text box, ensure that the table name **SalesOrderDetailForAnalysis** is specified.
- In the New Index window, in the **Index name** text box, type **SalesPriceAndQty** as the index name.
- In the **Columnstore columns** section, select the **Add** button to add the desired table columns to the index.
- In the Select Columns from Sales.SalesOrderDetailForAnalysis window, check the **OrderQty** and **UnitPrice** check boxes and select **OK** to add them to the index.
- In the New Index window, verify that the **OrderQty** and **UnitPrice** table columns are listed in the **Columnstore columns** section and select **OK** to save the new Column Store index.



- Execute the SELECT statement to find the performance of the SalesOrderDetailForAnalysis and SalesOrderDetail tables.
 - Open a **Query Editor** pane, enter the SELECT statement followed by the SUM keyword along with OrderQty column to display the sum of OrderQTY column from the **Sales.SalesOrderDetailForAnalysis** table.


```
SELECT Sum(OrderQty) FROM Sales.SalesOrderDetailForAnalysis;
```
 - Enter another SELECT statement followed by the SUM keyword along with OrderQty column to display the sum of OrderQTY column from the **Sales.SalesOrderDetail** table.


```
SELECT Sum(OrderQty) FROM Sales.SalesOrderDetailForAnalysis;
SELECT Sum(OrderQty) FROM Sales.SalesOrderDetail;
```
 - On the **SQL Editor** toolbar, select the **Include Actual Execution Plan** button to activate the **Execution Plan** tab.
 - On the **SQL Editor** toolbar, select **Execute** to see the difference that the column store index has on the performance.
 - View the execution plan with both the column store indexes performance in the **Execution plan** tab. Observe that the first query uses the Column Store Index to scan for the summarization; while the second query uses the Clustered Index. The difference in performance using the Column Store Index and the Clustered Index is 24% to 76%, which indicates that the query with the column store index is much more efficient than query with Clustered index.



- f) Close the **Query Editor** pane without saving the query.

ACTIVITY 8-4

Configuring FILESTREAM for Additional File Types

Scenario

The sales team of Adventure Works Cycles wants to upload MS Office 2010 documents into their database and want to be able to search their contents at any point in time. However, SQL Server only allows searching contents from MS Office 2003 type documents by default. You are asked to configure the FILESTREAM so that these additional file types can be parsed.

1. Select full text document types for the database.

- a) Open a **Query Editor** pane and enter the SELECT statement to display the document types available in the server.

```
SELECT * FROM sys.fulltext_document_types;
```

- b) Execute the SELECT statement to display the available document types.

- c) On the **Results** tab, observe that Microsoft Office 2003 documents such as .doc, .xls, and .ppt will be supported by default. Notice that the file formats supported by the latest versions of Microsoft Office 2007, 2010, 2013 (.DOCX, .XLSX, .PPTX) will not be displayed in the list, indicating that these file formats were not supported.

	document_type	class_id
10	.def	C7310720-AC80-11D1-8DF3-00C04FB6EF4F
11	.dic	C7310720-AC80-11D1-8DF3-00C04FB6EF4F
12	.doc	F07F3920-7B8C-11CF-9BE8-00AA004B9986
13	.dot	F07F3920-7B8C-11CF-9BE8-00AA004B9986



Note: In the “Results” tab scroll down and check for the “.doc” files and locate it.

- d) Close the **Query Editor** pane without saving the query.
2. Download and install the Microsoft Office2012 Filter pack on the working system.
 - a) Open the Internet Explorer, navigate to the link <http://www.microsoft.com/en-us/download/details.aspx?id=17062> and download the **Microsoft Office2012 Filter pack 64bit** onto the working machine.
 - b) In the Downloads window, right-click the download link, from the fly-out pane, select **Open** to run the setup.
 - c) If necessary, in the **Open File – Security Warning** dialog box, select **RUN** to start **Microsoft Office2012 Filter pack** installation.
 - d) In the **Welcome to the Microsoft Filter Pack 2.0 Setup Wizard**, select **Next** to continue with the installation.

- e) On the **End-User License Agreement** page, check the **I accept the terms in the License Agreement** check box and select **Next** to continue with the installation.



Note: The installation might take a few seconds.

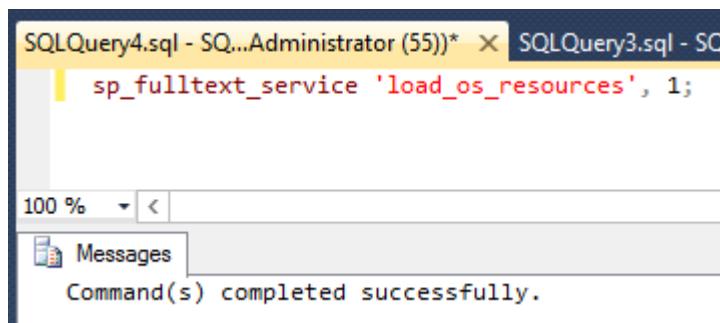
- f) Once the installation completes, select **OK** to complete the **Microsoft Filter Pack 2.0 Setup Wizard**.

3. Change the server role properties.

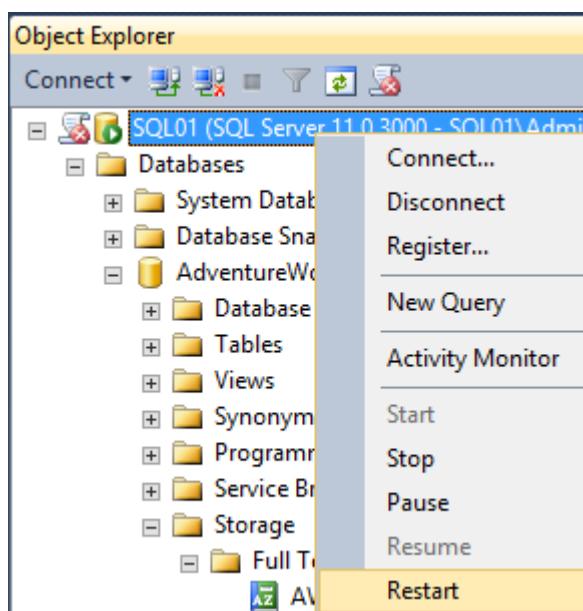
- a) Go to the SSMS and on the **Query Editor** pane, enter the `sp_fulltext_service` command followed by the resource name to load the iFilters and word breakers registered within Windows.

```
sp_fulltext_service 'load_os_resources', 1
```

- b) On the **SQL Editor** toolbar, select **Execute** and view the output on the **Messages** tab.



- c) Close the **Query Editor** pane without saving the query.
d) In the **Object Explorer** pane, right-click on the server node and select **Restart**.



- e) In the **Microsoft SQL Server Management Studio** dialog box, select **Yes** if prompted to restart the server service.



Note: You need to wait for the SQL Server instance to stop and start the process again.

4. Verify the registered file types.

- a) Open a **Query Editor** pane, re-enter the `SELECT` statement to list the registered document filetypes.

```
SELECT * FROM sys.fulltext_document_types;
```

- b) On the **SQL Editor** toolbar, select **Execute** to view the registered file types.
- c) On the **Results** tab, observe that the new file types—.DOCX, .XLSX, and .PPTX—will be displayed in the result set. This indicates that the FILESTREAM feature is configured to display advanced file types.

The screenshot shows the SSMS Query Editor window with the title bar "SQLQuery2.sql - SQ...Administrator (51)*". The query "SELECT * FROM sys.fulltext_document_types;" is entered in the query pane. The results pane displays a table with two columns: "document_type" and "class_id". The data rows are:

	document_type	class_id
25	.dic	C7310720-AC80-11D1-8DF3-00C04FB6EF4F
26	.doc	F07F3920-7B8C-11CF-9BE8-00AA004B9986
27	.docm	5A98B233-3C59-4B31-944C-0E560D85E6C3
28	.docx	5A98B233-3C59-4B31-944C-0E560D85E6C3
29	.dos	C1243CA0-BF96-11CD-B579-08002B30BFEB
30	.dot	F07F3920-7B8C-11CF-9BE8-00AA004B9986

- d) Close the **Query Editor** pane without saving the query.

ACTIVITY 8-5

Configuring File Table

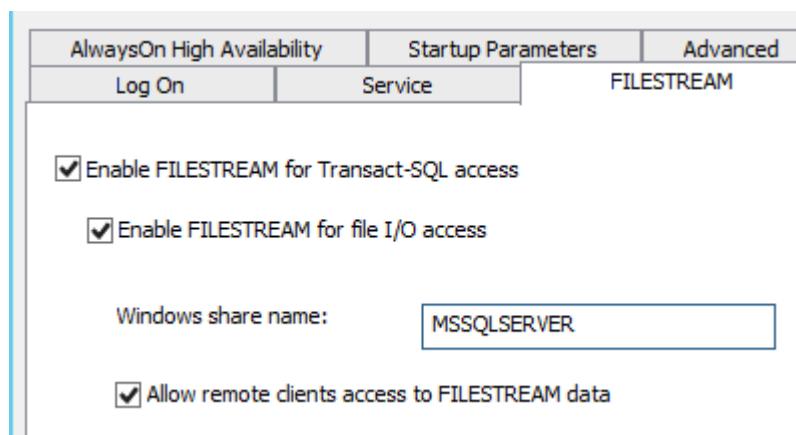
Before You Begin

Open a **Query Editor** pane and ensure that the **SalesDB** database is selected.

Scenario

The sales team of Adventure Works Cycles wants to store some financial documents in a particular location in the SalesDB database. You are asked to set up a standard file system folder for the sales team to store their files in. You decide to create a table with the FileTable feature to perform this task.

1. Enable FILESTREAM on the server instance.
 - a) In the desktop interface of Windows Server 2012, move your mouse pointer to the bottom-left corner of the screen and select **Metro Interface** to view the Start screen.
 - b) In the **Metro Interface** of Windows Server 2012, select **SQL Server Configuration Manager** to launch it.
 - c) In the SQL Server Confirmation Manager window, in the right pane, double-click the **SQL Server (MSSQLServer)** instance to open the Properties window.
 - d) In the SQL Server (MSSQLServer) Properties window, select the **FILESTREAM** tab.
 - e) On the **FILESTREAM** tab, check the following check boxes to enable the FILESTREAM feature in the server.
 - Enable FILESTREAM for Transact-SQL access
 - Enable FILESTREAM for file I/O access
 - Allow remote clients access to FILESTREAM data



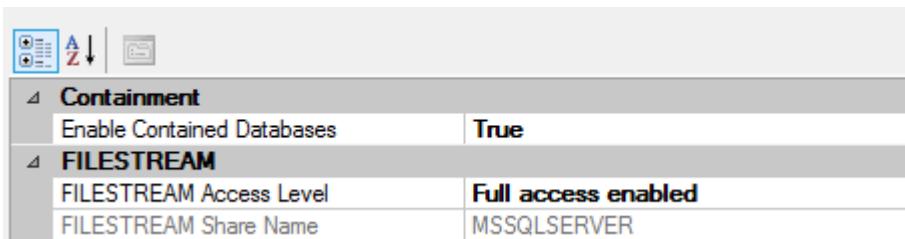
- f) In the SQL Server Properties window, select **OK** to enable the FILESTREAM feature.
- g) In the SQL Server Configuration Manager window, right-click the **SQL Server (MSSQLServer)** server instance and select **Restart**.



Note: You need to wait for the SQL Server instance to stop and start the process again

- h) Close the SQL Server Configuration Manager window after the restart process is completed.
2. Grant full access to the server.

- Get back to the SSMS and in the **Object Explorer** pane refresh the **SQL01** node, right-click on the server name and from the fly-out pane, select **Properties**.
- In the Server Properties window, on the **Select a Page** section, select **Advanced** page.
- In the **FILESTREAM** section, select **FILESTREAM Access Level** and from the drop-down list, select **Full access enabled** to grant full access to the FILESTREAM.



- In the Server Properties window, select **OK** to save the FILESTREAM access settings. When prompted for restart, select **OK**.
- Set the FILESTREAM Non-Transacted level.
 - On the **Object Explorer** pane, expand the **Databases** folder, right-click **SalesDB** and from the fly-out pane, select **Properties**.
 - In the Database Properties-SalesDB window, in the **Select a Page** section, select the **Options** page.
 - In the **Other options** section, in the **FILESTREAM** section, select **FILESTREAM Non-Transacted Access** and from the drop-down list, select **Full** to grant full access to the FILESTREAM. Select **OK** to enable the FILESTREAM.
 - In the **Open Connections** dialog box, select **Yes** if prompted to change the database properties and close all other connections to the database.
 - Execute the commands to set FILESTREAM access on the database.
 - Open a **Query Editor** pane, enter the ALTER DATABASE statement followed by the database name SalesDB.


```
ALTER DATABASE SalesDB
```
 - Enter the SET FILESTREAM keyword, along with the NON_TRANSACTED_ACCESS command with FULL to provide full filestream access to the SalesDB database.


```
SET FILESTREAM ( NON_TRANSACTED_ACCESS = FULL,
```
 - Type the DIRECTORY NAME keyword along with the directory name as N Files.


```
ALTER DATABASE SalesDB
SET FILESTREAM( NON_TRANSACTED_ACCESS = FULL, DIRECTORY_NAME = N'Files' );
```
 - Select **Execute** to execute the ALTER statement and set the FILESTREAM access to the SalesDB database.
 - On the **Messages** toolbar, observe the message "Command(s) completed successfully." is displayed.

- Alter a database and add a file to the file group.
 - Open a **Query Editor** pane and enter the ALTER DATABASE statement followed by the database name SalesDB.


```
ALTER DATABASE SalesDB
```
 - Enter the ADD FILEGROUP keyword followed by the filegroup name as FSFiles along with the CONTAINS FILESTREAM keyword.


```
ADD FILEGROUP FSFiles CONTAINS FILESTREAM;
```
 - Enter another ALTER DATABASE statement followed by the database name SalesDB.


```
ALTER DATABASE SalesDB
ADD FILEGROUP FSFiles CONTAINS FILESTREAM;
ALTER DATABASE SalesDB
```

- d) Enter the ADD FILE keyword followed by the name of the file as FSFiles along with the FILENAME path as c:\Temp\FS to access the filegroup.

```
ALTER DATABASE SalesDB
ADD FILEGROUP FSFiles CONTAINS FILESTREAM;
ALTER DATABASE SalesDB
ADD FILE (Name='FSFiles', FILENAME='c:\Temp\FS')
```

- e) Enter the TO FILEGROUP keyword followed by the filegroup name FSFiles.

```
"...
ALTER DATABASE SalesDB
ADD FILE (Name='FSFiles', FILENAME='c:\Temp\FS')
TO FILEGROUP FSFiles;
```

- f) Execute the ALTER statement and set a new filegroup for all file streams on the SalesDB database.
g) Close the **Query Editor** pane without saving the query.

6. Create a new FileTable called ProductDocuments.

- a) Open a new **Query Editor** pane and enter the CREATE TABLE statement followed by the table name ProductDocuments with alias name as FILETABLE.

```
CREATE TABLE ProductDocuments AS FILETABLE
```

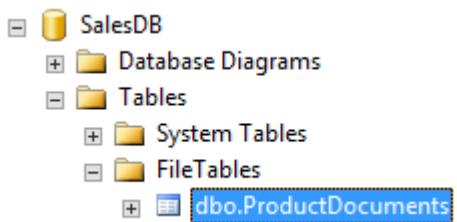
- b) Enter the WITH clause followed by the FILETABLE_DIRECTORY name as ProductDocuments and FILETABLE_COLLATE_FILENAME as database_default.

```
CREATE TABLE ProductDocuments AS FILETABLE
WITH
(
FILETABLE_DIRECTORY = 'ProductDocuments',
FILETABLE_COLLATE_FILENAME = database_default
)
```

- c) Select **Execute** to execute the CREATE TABLE statement to create a new FileTable called ProductDocuments.
d) Close the **Query Editor** panes without saving the queries.

7. Create a new file in the FileTable folder

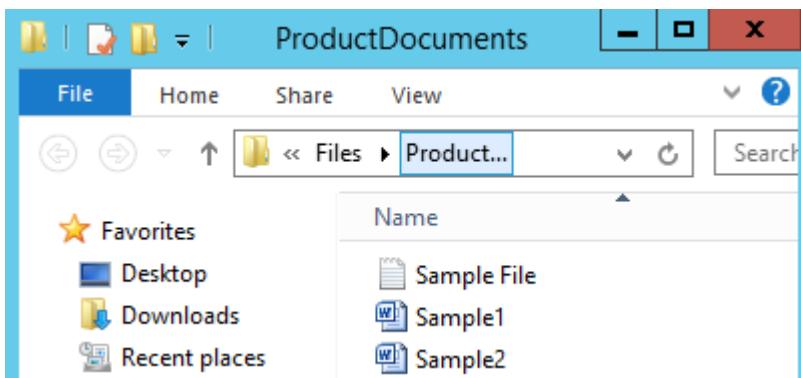
- a) In the **Object Explorer** pane, expand the **SalesDB** → **Tables** → **FileTables** folder.
b) In the **FileTables** folder, right-click and select **Refresh** and observe that the newly created **dbo.ProductDocuments** table is listed.



- c) In the **FileTables** folder, right-click the **dbo.ProductDocuments** table, from the fly-out pane, select **Explore FileTable Directory**.
d) In the Product Documents window, copy and paste the documents named "Sample1.doc" and "Sample2.doc" from C:\093108Data\Working with Indexes and Log Files folder.
e) In the Product Documents window, right-click on the blank space and from the fly-out pane, select **New→Text document** and name the new file as **Sample File** in the same window. These files are created in order to perform a search in the FILESTREAM.
f) Close the Product Documents window once the files are added.



Note: Observe that the files that are dispersed into the ProductDocuments folder are displayed in the result set.



- g) In the SSMS, open a **Query Editor** pane and enter the SELECT statement to display the records in the ProductDocuments table.

```
SELECT * FROM ProductDocuments;
```

- h) Execute the SELECT statement to verify the insertion of the folders.
i) On the **Results** tab, observe that the documents that you inserted into the ProductDocuments folder is inserted and displayed in the result set.

The screenshot shows the SSMS interface with a query window titled "SQLQuery17.sql - S...Administrator (59)*". The query entered is:

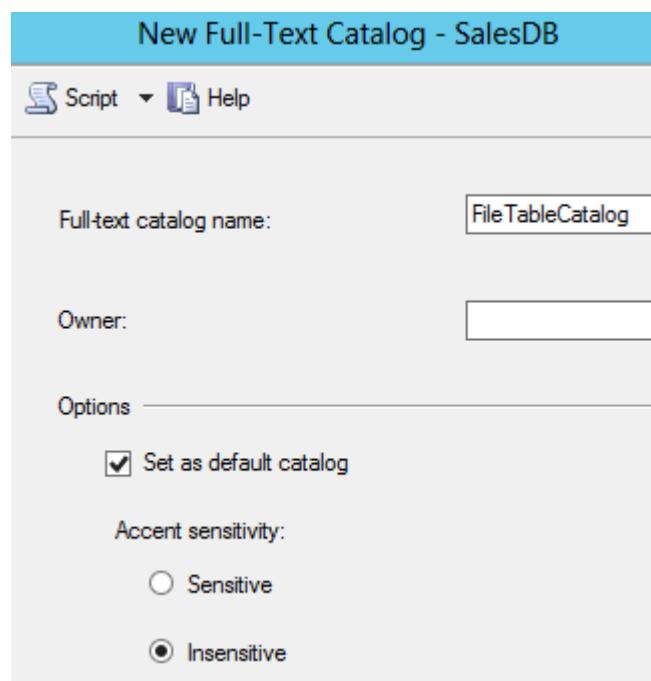
```
SELECT * FROM ProductDocuments;
```

The results pane shows the output of the query:

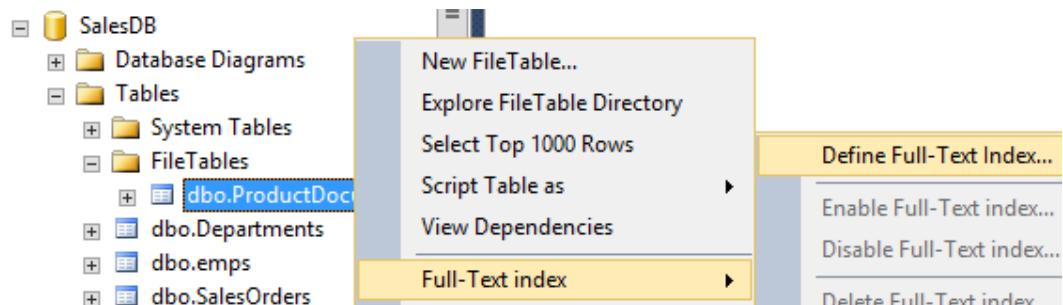
	stream_id	file_stream	name
1	F5B17519-0591-E211-...	0x504B030414000600...	Sample1.docx
2	F7B17519-0591-E211-...	0x504B030414000600...	Sample2.docx
3	F9B17519-0591-E211-...	0x	SampleFile.txt

8. Create Full-Text Catalog.

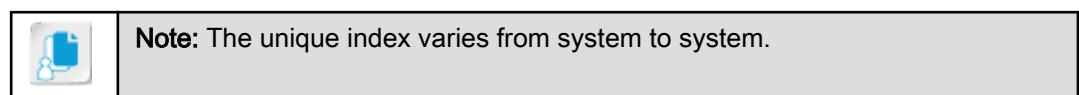
- In the **Object Explorer** pane, right-click on the server node and select **Refresh** and expand the **Databases** folder.
- In the **SalesDB** folder, expand the **Storage** folder.
- Right-click the **Full-Text Catalogs** folder and from the fly-out pane, select **New Full-Text Catalog**.
- In the **New Full-Text Catalog** dialog box, in the **Full-text catalog name** text box, type **FileTableCatalog**.
- In the **Options** section, check the **Set as default catalog** check box to set this catalog as default.
- In the **Accent sensitivity** section, select **Insensitive**. Select **OK** to save the settings.



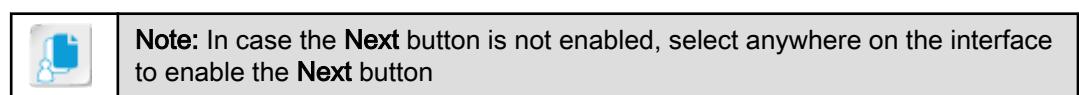
9. Set up a Full-text index.
- In the **SalesDB** folder, expand the **Tables->FileTables** table.
 - In the **FileTables** table, right-click **dbo.ProductDocuments** and from the fly-out pane, select **Full-Text index->Define Full-Text Index**.



- In the **Full-Text Indexing Wizard**, select **Next** to proceed with the installation process.
- On the **Select an Index** page, in the **Unique Index** drop-down list, ensure **PK_ProductID_5A5B77D547F031E0** is selected, and select **Next** to proceed with the installation process.



- On the **Select Table Columns** page, in the **Available Columns** list, check the **file_stream** check box, in the **Type Column** drop-down list, select **file_type**, and select **Next** to proceed with the installation process.



Available Columns ▾	Language for W...	Type Column
<input checked="" type="checkbox"/> file_stream		file_type
<input type="checkbox"/> file_type		
<input type="checkbox"/> name		

- f) On the **Select Change Tracking** page, in the **Track changes on the table/view as they occur** section, ensure **Automatically** is selected.

Track changes on this table/view as they occur:

Automatically

Manually

Do not track changes

- g) On the **Select Change Tracking** page, select **Next** to proceed with the installation process.
- h) On the **Select Catalog, Index Filegroup, and Stoplist** page, in the **Select full-text catalog** drop-down list, ensure that **FileTableCatalog** is selected as the default catalog in this database.
- i) In the **Select index filegroup** drop-down list, ensure that the **<default>** option is selected and in the **Select full-text stoplist** drop-down list, ensure that the **<system>** option is selected.

Full-Text Indexing Wizard

Select Catalog, Index Filegroup, and Stoplist

You must select an existing full-text catalog or create one for this database.

If this is a large table or view, or if the chosen full-text catalog has indexed other tables, consider assigning this table or view its own full-text catalog.

Select full-text catalog:

Create a new catalog

New catalog

Name:

Set as default catalog

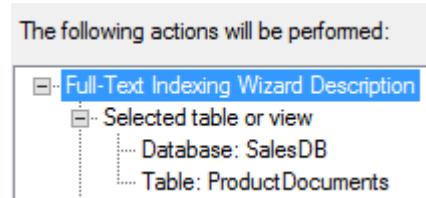
Accent sensitivity: Sensitive Insensitive

Select index filegroup:

Select full-text stoplist:

- j) On the **Select Catalog, Index Filegroup, and Stoplist** page, select **Next** to proceed with the installation process.
- k) On the **Define Population Schedules (Optional)** page, select **Next** to skip the population schedule screen and proceed with the installation process.

- i) On the **Full-Text Indexing Wizard Description** page, in the **Full-Text Indexing Wizard Description** node, expand the **Selected table or view** node to explore if **SalesDB** is the selected database and **ProductDocuments** is the table to which the FTI is set up.
- m) On the **Full-Text Indexing Wizard Description** page, select **Finish** to create the index.
- n) On the **Full-Text Indexing Wizard Process** page, observe that you have successfully created the FTI on the **SalesDB** database and the **ProductDocuments** table. Select **Close** to complete the installation process.



10. Display the file name and file type of the documents in the ProductDocuments table.

- a) Open a **Query Editor** pane and type the SELECT statement to display the name and file type columns from the ProductDocuments table.

```
SELECT name, file_type FROM ProductDocuments
```

- b) Enter the WHERE CONTAINS clause along with the file_stream and server keywords to display the files that contain filestream in the server.

```
SELECT name, file_type FROM ProductDocuments WHERE CONTAINS(file_stream, 'Server');
```

- c) Execute the SELECT statement to view the name and file_type columns.

- d) On the **Results** tab, observe that the results show the filename and file type that contain the keyword **Server** from the filestream that you were searching for from the ProductDocuments table.

	name	file_type
1	Sample1.docx	docx
2	Sample2.docx	docx

- e) Close the **Query Editor** pane without saving the query.

TOPIC C

Configure Log Files

You have configured advanced storage and indexing settings to the database. When working with databases, there may be circumstances where you would need to monitor the requests made to the server. In such instances, it's recommended to record the changes that are being made in the server and keep them for later use. Familiarizing yourself with configuring log files will enable you to track and monitor data transaction in your databases. In this topic, you will configure log files.

Log Files

Log files record the database transactions in the server and are used to monitor and ensure data integrity in the event of system restoration. When any transaction is carried out in a SQL Server database, the log file will record the changes and once the transactions are complete, the logs will be written back to the database. If the write is successful, then the log file will be truncated.



SQL Server actually uses a concept of *Virtual Log Files (VLFs)* within the log file. These are basically equal size “blocks” within the real log file, which are allocated and reused as and when transactions are added and truncated.

SQL Server starts using VLFs from the beginning of the physical log file. In the process wherein the transactions are being committed or rolled back, the content within VLFs will get truncated while current VLFs will be used. A *Log Sequence Number (LSN)* is a unique id which maintains the position of the last current transaction that can be committed or rolled back.

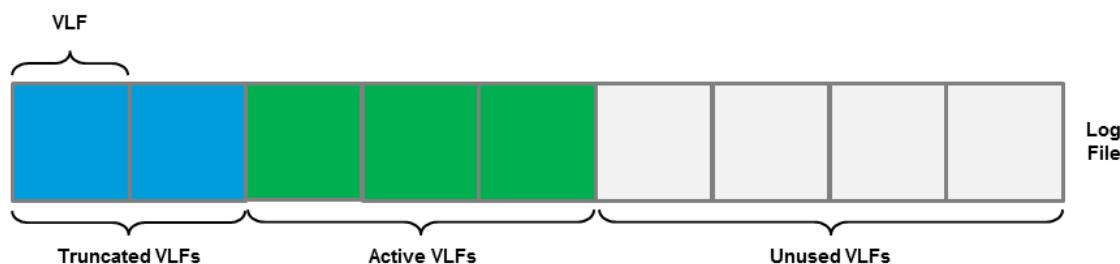


Figure 8-4: Virtual log files in a physical log file.

Once the system reaches the end of the file, it will usually return back to the beginning and start reusing the truncated VLFs in the file. However, if transactions are still open from the beginning of the log, a log backup (that causes a log truncate) is not done, or the transaction requires more space than that the free VLFs in the log can handle, and then the VLFs are unable to be reused. Instead, the engine will go ahead and increase the size of the physical log file, and therefore increase the size or number of VLFs. This is basically done using the Auto Grow settings on the log.

The DBCC SHRINKFILE Command

DBCC SHRINKFILE is a command where extremely advanced options are used to shrink the size of a log file from a database. You can resize a file to a size that is lesser than its previous size. Utmost care and caution should be taken while using this command on the database.

The syntax to use DBCC SHRINKFILE is as follows.

```
DBCCSHRINKFILE(<file_name or file_id>, target_size)
```

Example

A DBCC SHRINKFILE statement to shrink the log file of the SalesDB database to 2 MB is as follows.

```
DBCC SHRINKFILE (SalesDB_Log, 2);
```

Need to Cycle Error Logs

Cycling log files is done when you want to use multiple log files on each database. While you can create more than one log file for any database, the database will use one log file at any point of time. You can however cycle through these log files in a *round-robin* approach if you wish to balance the log files in the database. This will however make recovery procedures much more complex.

It is recommended that you use only one transaction log file per database. The only case where you may want to add multiple log files is when changing the location of the log file in a running database. Therefore, by adding a new location to the log file, all new transactions will go the moved log file and after a while you can take a backup of the old log and then remove it completely.

Round-Robin Approach

The round-robin approach helps distribute the load equally across all databases. It changes the database endpoint for every new connection consistently.

Log File Growth Properties



Log File Growth Properties

The log files have a few important properties that help record the activities performed in the server and control the growth of the log file.

Property	Used To
Enable Autogrowth	Enable the log file to grow automatically when SQL Server runs out of physical space to add transaction log entries. If this setting is off, when the log file grows beyond its size the SQL Server cannot write to the log, it will throw an error and stop the data transaction.
File Growth	Specify the size that the log file can grow. You can specify this in percentages or absolute values of megabytes.
Maximum file size	Limit the maximum file size of the log, you can set this option to limit the file size of the log to a specific size in MB or you can simply select the Unlimited option.

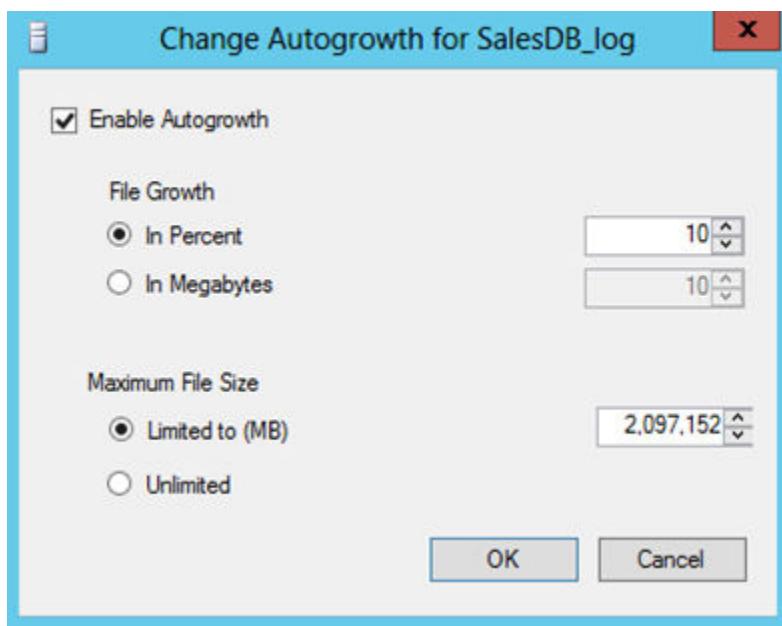


Figure 8-5: The *Change Autogrowth* dialog box with log file properties.

A careful planning of the log file property settings is required, especially for large, transactional databases. This is because if transactions come in very quickly and older ones are not yet truncated, the system will be unable to find free space and start growing the file.

A large, unmanaged transaction log file will not only take longer time to backup, but can also suddenly fill up the entire disk space of the system. This is why very detailed preparation is required in such cases.

Trace Flags

Trace Flags are values used for tracing certain issues that might occur in SQL Server. There are a number of trace flags that are depicted using numbers, such as 260, 1204, and 3205, that stand for a single system or session wide property that allows you to temporarily change the behavior of the system. Trace flags can be used within a single session or globally when the server starts up.



Note: Microsoft will be deprecating the Trace Flags feature in the future versions of SQL Server. So use them sparingly in your SQL programs.

The DBCC TRACE ON and DBCC TRACE OFF commands are used to enable and disable the trace flag feature in SQL Server.

The syntax to enable or disable trace flags feature is as follows.

```
DBCC TRACE [ON | OFF] (<traceflag> [, -1]);
```

In this syntax, the <traceflag> is the number of the trace flag you wish to enable or disable. You can use the option “-1” if you wish to enable or disable the trace file globally. Leaving out this option will reflect for the current session only.

Some of the commonly used trace flag values are listed in the table.

Trace Flag Value	Used To
260	Print the DLL’s versioning information.
3205	Disable the hard disk compression for tape drivers.
1204 and 1222	Display the information about the type of locks included in a deadlock issue.

Trace Flag Value	Used To
1211	Disable log escalation issues that occur due to memory pressure.
4199	Control changes made by multiple query optimizers.

Example

The command to enable the trace log feature is as follows.

```
DBCC TRACEON(1204, -1);
```

This statement will enable the trace flag feature and the trace value 1204 will display the information about the type of lock used in a deadlock issue.

Deadlock Issues



Deadlock Issues

Deadlocks are a special error condition in SQL Server where two different applications or connections hold locks on two separate objects and are waiting for the object held by the other to be released. Because this is a circular dependency, the lock condition becomes indefinite.

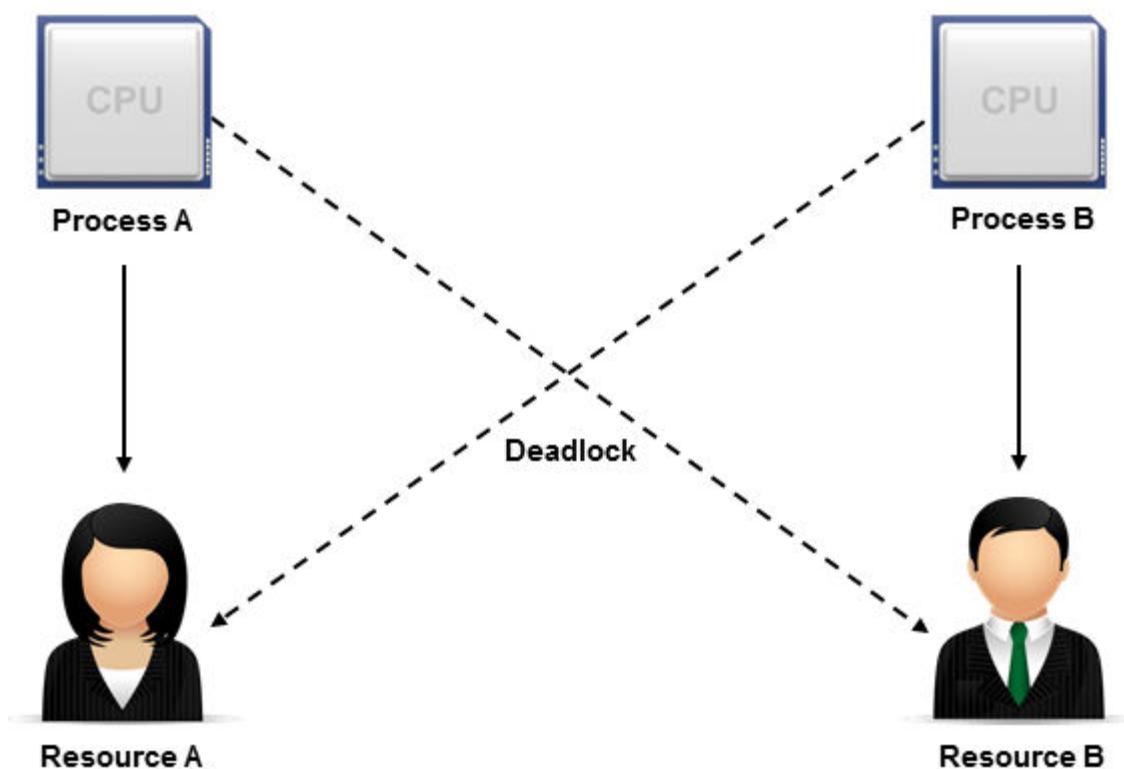


Figure 8-6: A deadlock condition between two resources.

SQL Server handles this deadlock condition by selecting one of the transactions and aborting it, therefore rolling it back so that the other one can continue. SQL Server uses the one with the minimum transaction impact and makes it the “victim.” The other one works from that point on.

You can change the priority of the deadlock of sessions using the SET DEADLOCK_PRIORITY values to LOW, NORMAL, or HIGH or a value between -10 to +10. You can also use the trace flag numbers 1204 and 1222 to get more information regarding the deadlocks.

Live Lock

Live Locks are situations where a request for an exclusive lock on a resource is continuously denied to a process since overlapping shared locks keep getting access.

For instance, Process A gets a shared lock on a resource. Process B requests an exclusive lock on the same will be kept in a waiting state. Then Process C asks for a shared lock on the same and gets it. Process A releases the lock but process B still cannot get it since now Process C is holding the shared lock and so on.



Live Lock

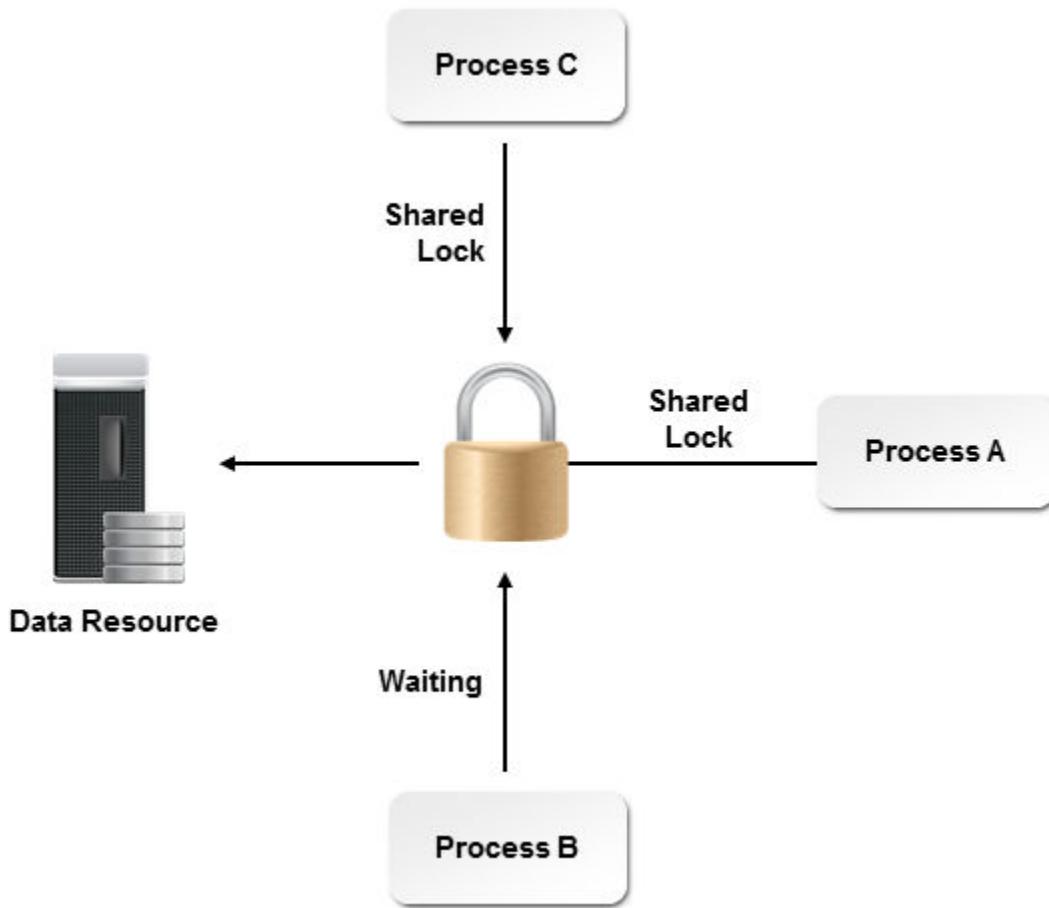


Figure 8-7: Live Locks shared with processes in a data resource.

However, in SQL Server 2012 Live Locks are handled automatically. If the system detects this situation it denies any more shared locks after the fourth such request. The process requesting the exclusive lock will then get the lock.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Configure Log Files

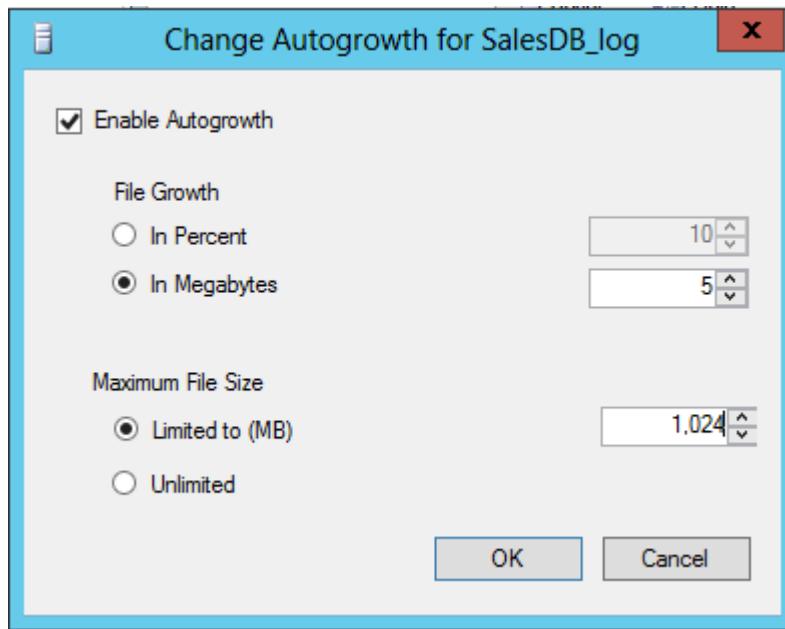
ACTIVITY 8-6

Changing Log File Growth Properties

Scenario

The products of Adventure Works Cycles have been put online on the company website for product promotion and sales. Because of the attractive offer provided by the company, large volume of transactions started to happen quickly. You are asked to ensure that the users were able to get the required products without any errors. You decide to plan for an adequate log file to record and monitor the data transactions.

1. Configure the log file growth properties in the SalesDB database.
 - a) In the **Object Explorer** pane, expand the **Databases** folder.
 - b) In the **Databases** folder, right-click the **SalesDB** database and select **Properties**.
 - c) In the Database Properties-SalesDB window, on the **Select a page** section, selects the **Files** page.
 - d) On the **Files** page, in the **Databases files** section, select the **SalesDB_log** file, and in the **Autogrowth / Maxsize** column select the **ellipsis (...)** button.
 - e) In the **Change Autogrowth for SalesDB_log** dialog box, ensure that the **Enable Autogrowth** check box is selected.
 - f) In the **File Growth** section, select the **In Megabytes** option and set the file size to **5MB**.
 - g) In the **Maximum File Size** section, ensure that the **Limited to (MB)** is selected and type **1024** in the text box. Select **OK** to save the log settings.



- h) In the Database Properties - SalesDB window, select **OK** to save the database properties and close the window.
2. Which option maintains the position of the last current transaction in a physical log file?
 - Virtual Log Files
 - Log Sequence Number
 - DBCC SHRINKFILE

- Truncated Virtual Log Files
3. Which of the following properties hold good in describing the attributes of a log file? (Choose three.)
- Enable Autogrowth
 - File Growth
 - Maximum file size
 - DBCC SHRINKFILE
 - Maximum Word count
-

ACTIVITY 8-7

Examining Deadlock Issues Using SQL Server Logs and Trace Flags

Before You Begin

Open a new Query Editor pane and ensure that the **SalesDB** database is selected.

Scenario

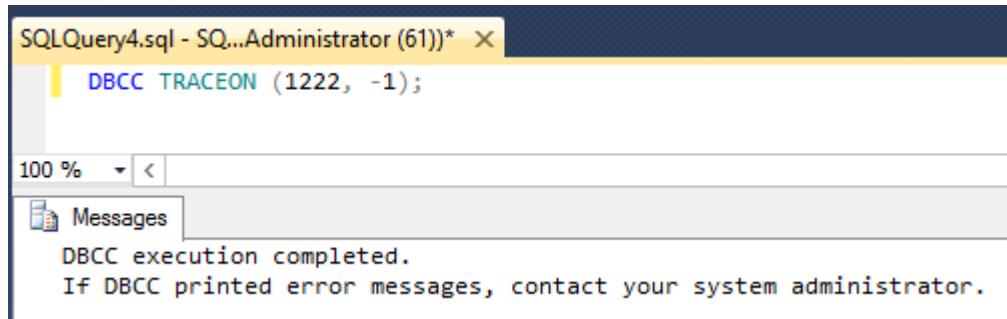
You manager wants you to train the database administration team on how to examine the deadlock issues that occur during data transaction. You decide to create two new tables and perform simultaneous data transaction in it to find the dead lock victim object by verifying the server log and trace flags.

1. Enable the trace flag feature.

- Open a Query Editor pane, enter the DBCC TRACEON command, followed by the trace value 1222 and -1 to enable the trace flag feature and display the information about the type of locks included in the deadlock issue.

```
DBCC TRACEON (1222, -1);
```

- Select **Execute** to execute the DBCC TRACEON command to turn on the trace flag feature globally.
- In the **Messages** tab, observe the message "DBCC execution completed. If DBCC printed error messages, contact your system administrator." is displayed.



- Close the **Query Editor** pane without saving the query.
- 2. Create a new table named dbo.departments and add few records to it.**
- Open a **Query Editor** pane and enter a CREATE TABLE statement followed by the table name **dbo.departments**.
- ```
CREATE TABLE [dbo].[Departments] (
```
- Enter the code to add the following columns in the **dbo.departments** table.
    - DepartmentID column with integer data type with identity of 1,1 and NOT NULL constraint.
    - DepartmentName column with varchar data type with maximum character length of 50 characters and NOT NULL constraint.
    - Description column with varchar data type with maximum character length of 150 characters and NULL constraint.

```
CREATE TABLE [dbo].[Departments] (
[DepartmentID] [int] IDENTITY(1,1) NOT NULL,
[DepartmentName] [varchar](50) NOT NULL,
```

```
[Description] [varchar](150) NULL
)
```

- c) Enter the INSERT INTO statements followed by the table name Departments.

```
"...
[Description] [varchar](150) NULL
)
INSERT INTO Departments
```

- d) Enter the VALUES keyword followed by the following data values that need to be inserted into the departments table.

- Department Name as HR, and Description as Human Resources.
- Department Name as IT, and Description as Information Technology.
- Department Name as DEV, and Description as Product Development.

```
"...
INSERT INTO Departments
VALUES
('HR', 'Human Resources'),
('IT', 'Information Technology'),
('Dev', 'Product Development');
```

- e) Select **Execute** to execute the INSERT INTO statement to create a table and add the required rows to it.

The screenshot shows the SQL Server Management Studio (SSMS) interface. A query window titled "SQLQuery19.sql - S...Administrator (51)\*" contains the following T-SQL code:

```
CREATE TABLE [dbo].[Departments](
 [DepartmentID] [int] IDENTITY(1,1) NOT NULL,
 [DepartmentName] [varchar](50) NOT NULL,
 [Description] [varchar](150) NULL
)
INSERT INTO Departments
VALUES
 ('HR', 'Human Resources'),
 ('IT', 'Information Technology'),
 ('Dev', 'Product Development');
```

The code is highlighted in blue and red. The status bar at the bottom left shows "100 %". Below the status bar is a "Messages" pane which displays "(3 row(s) affected)".

- f) Close the **Query Editor** pane without saving the query.

3. Create another new table named dbo.emps and add few records to it.

- a) Open a **Query Editor** pane and enter a CREATE TABLE statement followed by the table name dbo.emps.

```
CREATE TABLE [dbo].[emps] (
```

- b) Enter the code to add the following columns in the dbo.emps table.

- Empid column with integer data type and NOT NULL constraint.
- Name column with varchar data type with maximum character length of 50 characters and NOT NULL constraint.

```
CREATE TABLE [dbo].[emps] (
 [Empid] [int] NOT NULL,
```

```
[Name] [varchar](50) NOT NULL
)
```

- c) Enter the INSERT INTO statements followed by the table name emps.

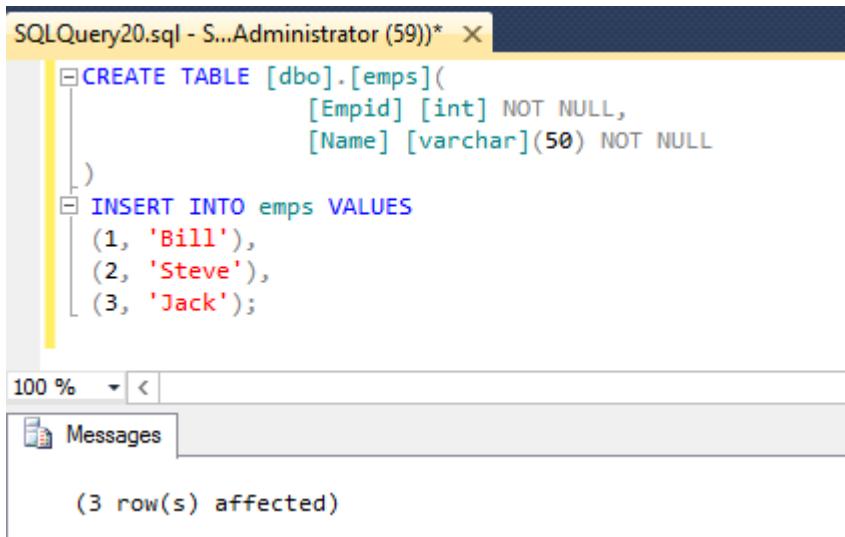
```
"...
[Name] [varchar](50) NOT NULL
)
INSERT INTO emps
```

- d) Enter the VALUES keyword followed by the following data values that need to be inserted into the departments table.

- Empid as 1 and Name as Bill.
- Empid as 2 and Name as Steve.
- Empid as 3 and Name as Jack.

```
"...
INSERT INTO emps
VALUES
(1, 'Bill'),
(2, 'Steve'),
(3, 'Jack');
```

- e) On the **SQL Editor** toolbar, select **Execute** to create a table and add the required rows to it.



- f) Close the **Query Editor** pane without saving the query.

4. Type the code to create a new transaction on the departments table.

- a) Open a **Query Editor** pane, enter the BEGIN TRAN and COMMIT TRAN keywords to create a new transaction.

```
BEGIN TRAN
COMMIT TRAN
```

- b) In between the TRAN keyword, enter the UPDATE statement followed by the table name emps.

```
BEGIN TRAN
UPDATE emps
COMMIT TRAN
```

- c) Type the SET keyword to set the Name field as bill1001.

```
BEGIN TRAN
UPDATE emps SET Name = 'bill1001'
COMMIT TRAN
```

- d) Type a WHERE clause followed by the EmpID as 1.

```
BEGIN TRAN
UPDATE emps SET Name = 'bill1001' WHERE EmpID = 1;
COMMIT TRAN
```

- e) Enter the WAITFORDELAY keyword and type the delay time as 00:00:05 with a message stating wait for 5 minutes.

```
"...
UPDATE emps SET Name = 'bill1001'
WHERE EmpID = 1;
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
COMMIT TRAN
```

- f) Enter another UPDATE statement followed by the table name emps.

```
"...
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE Departments
COMMIT TRAN
```

- g) Type the SET keyword to set the DepartmentName field as HumanResources 1001.

```
"...
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE Departments SET DepartmentName = 'Human Resources1001'
COMMIT TRAN
```

- h) In the UPDATE statement type another WHERE clause followed by the DepartmentID as 1.

```
BEGIN TRAN
UPDATE emps SET Name = 'bill1001' WHERE EmpID = 1;
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE Departments SET DepartmentName = 'Human Resources1001' WHERE
DepartmentID=1;
COMMIT TRAN
```

```
BEGIN TRAN
UPDATE emps SET Name = 'bill1001' WHERE EmpID = 1;
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE Departments SET DepartmentName = 'Human Resources1001'
WHERE DepartmentID=1;
COMMIT TRAN
```



**Note:** Do not execute this transaction as yet.

5. Create another new transaction on the emps table.

- a) Open a **Query Editor** pane, enter the BEGIN TRAN and COMMIT TRAN keyword to create a new transaction.

```
BEGIN TRAN
COMMIT TRAN
```

- b) In between the TRAN keyword, enter the UPDATE statement followed by the table name Departments.
- c) Type the SET keyword to set the DepartmentName field as Human Resources1001.

```
BEGIN TRAN
UPDATE Departments SET DepartmentName = 'Human Resources1001'
COMMIT TRAN
```

- d) Type a WHERE clause followed by the DepartmentID as 1.

```
BEGIN TRAN
UPDATE Departments SET DepartmentName = 'Human Resources1001' WHERE
DepartmentID=1;
COMMIT TRAN
```

- e) Enter WAITFOR DELAY and type the delay time as 00:00:05 with a message stating wait for 5 minutes.

```
"...
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
COMMIT TRAN
```

- f) Enter another UPDATE statement followed by the table name emps.

```
"...
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE emps
COMMIT TRAN
```

- g) Type the SET keyword to set the Name field as bill1001.

```
"...
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE emps SET Name = 'bill1001'
COMMIT TRAN
```

- h) Type another WHERE clause followed by the empid as 1.

```
'...
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE emps SET Name = 'bill1001' WHERE EmpID = 1;
COMMIT TRAN
```

- i) Select **Execute** to execute the TRAN statement to view the rows with the department and employee name.

```
BEGIN TRAN
UPDATE emps SET Name = 'bill1001' WHERE EmpID = 1;
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE Departments SET DepartmentName = 'Human Resources1001'
WHERE DepartmentID=1;
COMMIT TRAN
```

(1 row(s) affected)  
Msg 1205, Level 13, State 45, Line 4  
Transaction (Process ID 64) was deadlocked on lock resources  
with another process and has been chosen as the deadlock victim.  
Rerun the transaction.

- j) On the **Results** tab, observe that the table is getting updated.  
k) Immediately switch to the other window in which the query was typed previously, on the **SQL Editor** toolbar, select **Execute**. Observe that both queries are updated simultaneously but in opposite order, causing a deadlock.

```

SQLQuery22.sql - S...Administrator (54))*
SQLQuery21.sql - S...Administrator (58))*
```

```

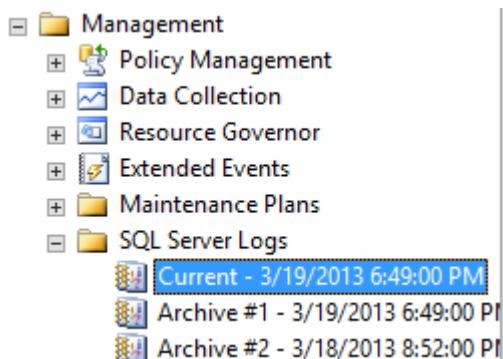
BEGIN TRAN
UPDATE emps SET Name = 'billi1001' WHERE EmpID = 1;
WAITFOR DELAY '00:00:05' -- Wait for 5 ms
UPDATE Departments SET DepartmentName = 'Human Resources1001'
WHERE DepartmentID=1;
COMMIT TRAN
```

100 % < Messages

(1 row(s) affected)

(1 row(s) affected)

- i) On the Object Explorer pane, expand Management →SQLServer Logs and select Current.



- m) In the Current log window, observe that the deadlock issue was detected and one of the processes was decided to be marked as the deadlock victim. Select Close to close the current log window.

| Log file summary: No filter applied |         |                                                    |
|-------------------------------------|---------|----------------------------------------------------|
| Date                                | Source  | Message                                            |
| 3/19/2013 7:32:51 PM                | spid22s | executionStack                                     |
| 3/19/2013 7:32:51 PM                | spid22s | process id=process270796cf8 taskpriority=0 logused |
| 3/19/2013 7:32:51 PM                | spid22s | process-list                                       |
| 3/19/2013 7:32:51 PM                | spid22s | deadlock victim=process270796cf8                   |
| 3/19/2013 7:32:51 PM                | spid22s | deadlock-list                                      |
| 3/19/2013 7:29:20 PM                | spid58  | DBCC TRACEON 1222, server process ID (SPID) 58     |

- n) Close the **Query Editor** panes without saving the queries.

## Summary

In this lesson, you worked with full text indexes to search for the required data in your database. You have also configured advanced storage and indexing settings to manage your organizational data systematically. In addition, you have also configured log files that will enable you to monitor the data transactions in the server, thereby enabling you to solve your database issues with ease.



Use the review questions provided to generate discussion among the participants.

### What are the major functionalities of full text indexing options?

**A:** Answers may vary, but will include: the ability to index content within standard document types such as Word documents, Excel spreadsheets, PowerPoint presentations, Adobe PDFs and to manage a custom list for thesaurus words as well as stop words.

### When does a deadlock issue occur?

**A:** Answers will vary, but may include: when two different applications or connections hold locks on two separate objects and are waiting for the object held by the other to be released.



**Note:** Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

9

# Working with Backup and Restore

**Lesson Time:** 2 hours

## Lesson Objectives

In this lesson, you will:

- Perform a backup.
- Restore a database.
- Import and export data.
- Configure the SQL Server agent.

## Lesson Introduction

You have configured the advanced indexing options and log files available in SQL Server® to retrieve required data. However, there may be instances, when the server itself stops working and data might be lost. In such cases, backing up the server will ensure data protection. SQL Server contains options for taking backups of the server periodically and restoring the databases from the backups in case of failures. SQL Server also allows you to transfer your database data using various data transfer techniques. Familiarizing yourself with the backup and restore options will enable you to protect your databases from unexpected failures. In this lesson, you will work with backup and restore options.

# TOPIC A

## Perform a Backup

Data in a server is crucial for any organization. It is important to ensure that a copy of the data is always maintained. The best approach to protect data from being lost is copying the required data and maintaining it offline. A better understanding about database backup will enable you to ensure that data is protected even in unexpected situations. In this topic, you will perform a backup.

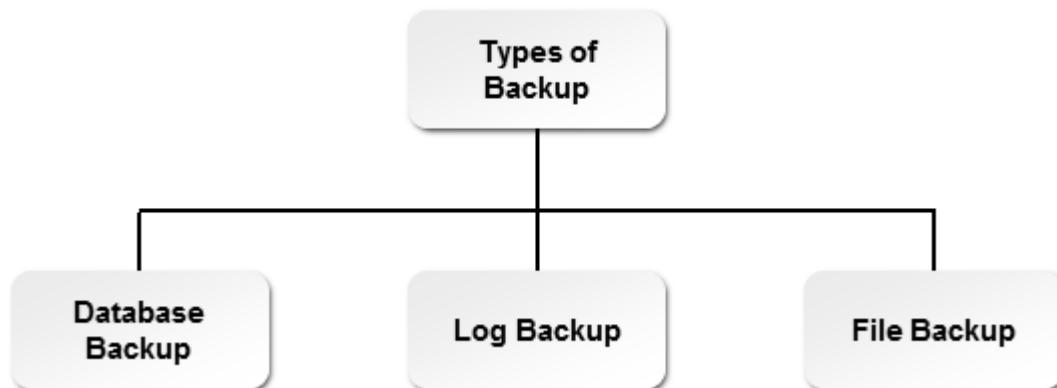
### Backup



Backup

A *backup* is a copy of the server data and is critical for database administration and maintenance. SQL Server allows you to take backups of a database as a single job, on a schedule, or manually.

The different types of backups that can be made from SQL Server are listed in the table.



*Figure 9–1: Types of backup.*

| Type of Backup  | Description                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------|
| Database backup | A backup of the database.                                                                        |
| Log backup      | A backup of the transaction log of the database. This basically truncates the log once complete. |
| File backup     | A backup of files or filegroups of a database.                                                   |

SQL Server uses the concept of shadow copies to perform data backup. That is, operations such as INSERT, UPDATE, and DELETE can happen during a backup operation. Shadow copies are copies of data at an instant of time. The only restriction here is that the database cannot be altered when the backup is processed. SQL Server 2012 also allows compressing the backup to save space.

### Database Backup



Database Backup

A *database backup* is the copy of a database in the server. It is the most common type of backup and is sometimes referred to as SQL Server backup. It is recommended that you take a backup of all the databases regularly.

The syntax to back up a database using transact SQL is as follows.

```

BACKUP DATABASE <database name>
TO DISK = '<back up file location>'
```

```
WITH FORMAT,
MEDIENAME = '<backup media name>',
NAME = '<name>;'
```

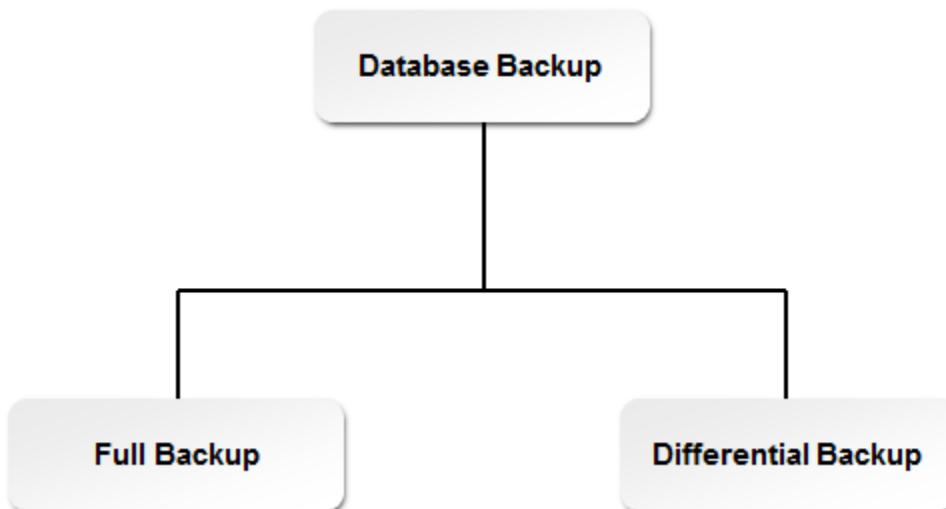
### Example

A BACKUP DATABASE statement to take a complete backup of the AdventureWorks2012 database is as follows.

```
BACKUP DATABASE AdventureWorks2012
TO DISK = 'C:\Temp\Backup\ADW2012.Bak'
WITH FORMAT,
MEDIENAME = 'ADWBackup',
NAME = 'Full Backup of AdventureWorks2012';
```

This code snippet will back up the AdventureWorks2012 database in the location c:\Temp\Backup with the name of the backup file as ADW2012.bak. The name of the media set in which the backup is stored is ADWBackup and the name of the backup is Full Backup of AdventureWorks2012.

The database backup can be done in two ways.



**Figure 9–2: Types of database backup.**

| Database Backup     | Description                                                                        |
|---------------------|------------------------------------------------------------------------------------|
| Full backup         | Taking an entire backup of the database and its log file.                          |
| Differential backup | Taking a backup of only the data that has been changed since the last full backup. |

### Copy-Only Backup

Copy-only backup is a backup that is not counted as a regular backup and is performed only for special purposes. Therefore, the data and log copied by the copy-only backup are also copied when the next regular backup is performed.



**Note:** For more information about database backup, check out the **LearnTO Back Up Databases** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.



You may want to show LearnTO **Back Up Databases** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.

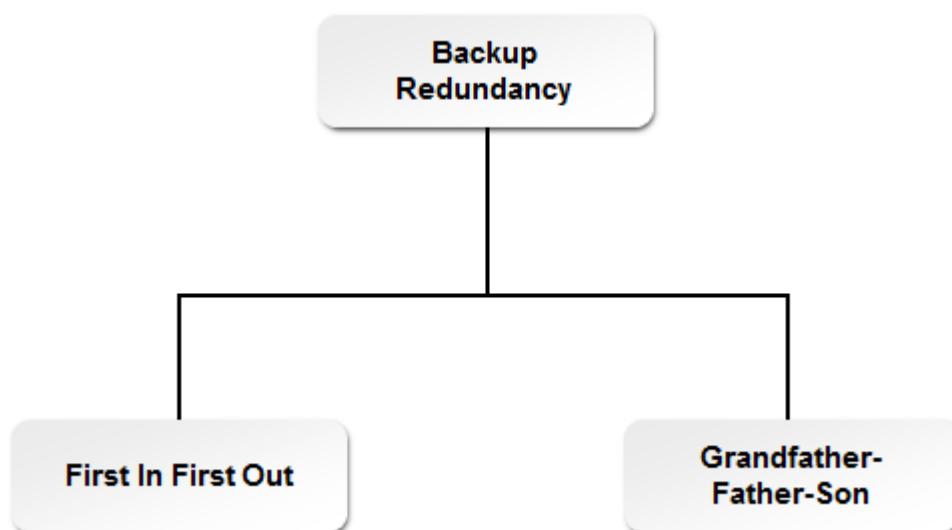


**Backup Redundancy**

## Backup Redundancy

*Backup redundancy* is a strategy to ensure that your backup is also backed up and that you can recover data to any point in time. Consider the case where you take a full backup every Friday of the week at 8:00 P.M. If the database crashes on the next Wednesday and you need to recover it, you can recover only the data that was present on the last Friday. The data from Monday until the database crash is lost.

The two commonly used backup redundancy strategies are as follows.

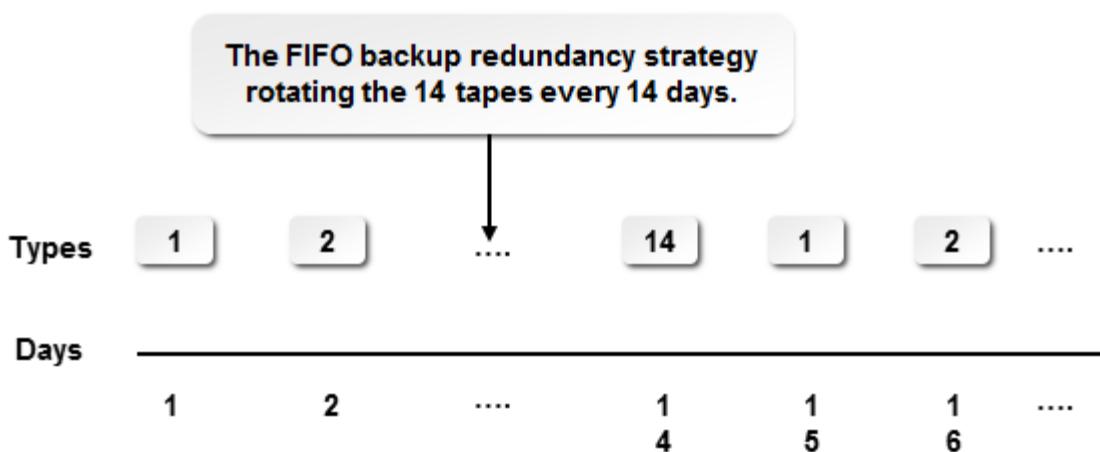


*Figure 9–3: Types of backup redundancy strategies.*



**FIFO**

In *First In First Out* (FIFO) strategy, you use multiple media sets for backups and rotate them over a period of time equal to the number of media sets. For instance, if you have 14 tapes, you will start using Tape 1 on day 1, Tape 2 on day 2, and so on until Tape 14 on day 14. On day 15, you will reuse the oldest of the tapes—that is, tape 1 again. You can go back and recover data at a point in time for the last 14 days.



*Figure 9–4: The FIFO backup redundancy strategy rotates 14 tapes every 14 days.*

## Grandfather–Father–Son

The *Grandfather–Father–Son* strategy defines three sets of backups – daily (son), weekly (father), and monthly (grandfather). Each set is rotated using the FIFO method and the latest backup of one set graduates to the next set. The latest daily (son) backup at the end of the week will become the weekly (father) backup; the latest weekly (father) backup will become the monthly (grandfather) backup at the end of the every month. At the end of the year, the latest grandfather backup should be taken offsite and stored securely.



Grandfather–Father–Son

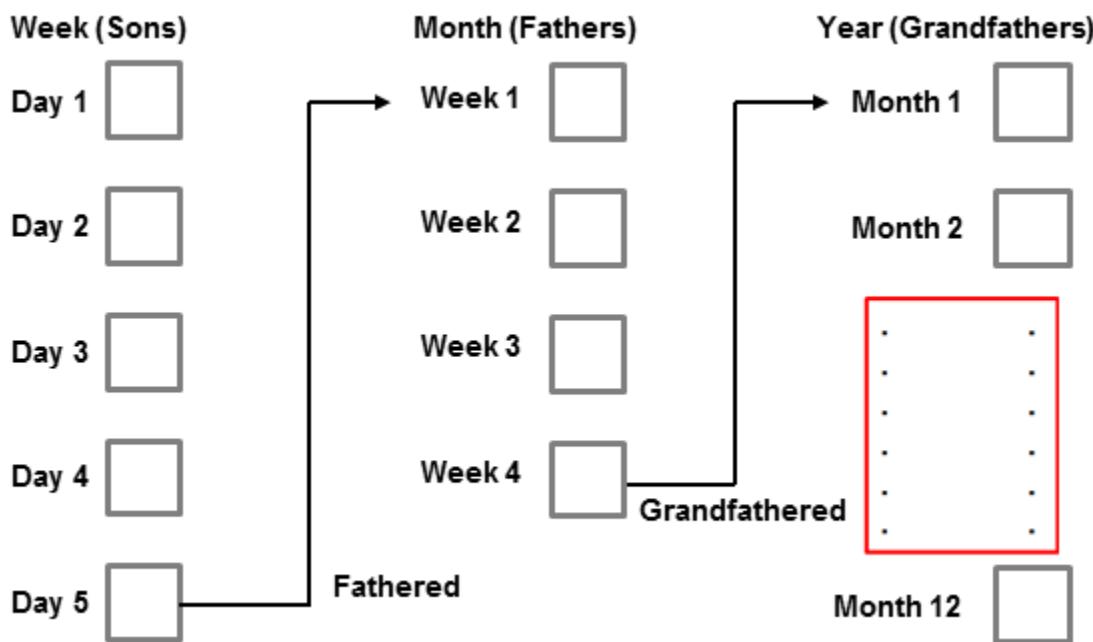


Figure 9–5: The *Grandfather–Father–Son* backup redundancy strategy graduates the latest backup of one set to the next set.

## Guidelines to Select a Backup Strategy

There are a few guidelines that you should follow when selecting a backup strategy.

1. Always have more than one backup, so that in case a backup media fails, you can use the other.
2. Create backups on a different media device than the one on which the database files are stored. This ensures that if the media of the drive fails, you don't lose the backup as well.
3. Use the FULL Recovery Model on your databases. This ensures the highest level of data recovery options in a backup.
4. Create a backup and media rotation schedule depending on the amount of data that changes in your database. Follow a standard FIFO strategy for normal databases and a 3-level Grandfather–Father–Son strategy for important databases.
5. Always test your backups to check if they have been copied correctly. You cannot recover databases if the backup is incomplete or corrupted.
6. In case of SQL Server that allows backing up the Transaction Log, use a combination of Full backups (weekly), Differential backups (daily), and Log backups (every hour or 15 minutes depending on data change rate). You can then restore the transaction logs at the end if required.



Discuss about the disk space used by the different types of backups with the students.

## Strategies to Protect Data if Backup Media Is Lost

It is important to protect the backup media from loss as well. There are two types of things you need to protect from.

- **Media or backup failure:** This happens when the backup media itself fails or the backup that you took on the media fails for some reason. You should always test your backups on a different machine as part of the backup procedure to verify that the media and the backup are working. Taking a copy of the verified backup and maintaining it offsite is also important to ensure that if any issue develops subsequently, you can still use the backup.
- **Loss of backup that contains sensitive data:** What happens if a full backup of your database with sensitive information is lost? You need to ensure that if the data falls into wrong hands, they should not be able to use it. You've already learned how to protect organizational data from this scenario. You can use TDE on the database so that the backups are also automatically encrypted with the same digital certificate. Even if the backup falls into the wrong hands, they will find it impossible to restore the database anywhere without the right key. Of course, this does mean that the database encryption key is maintained separately and extremely securely.



Ask the students to list the different types of backup media used in their organization.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Perform a Backup

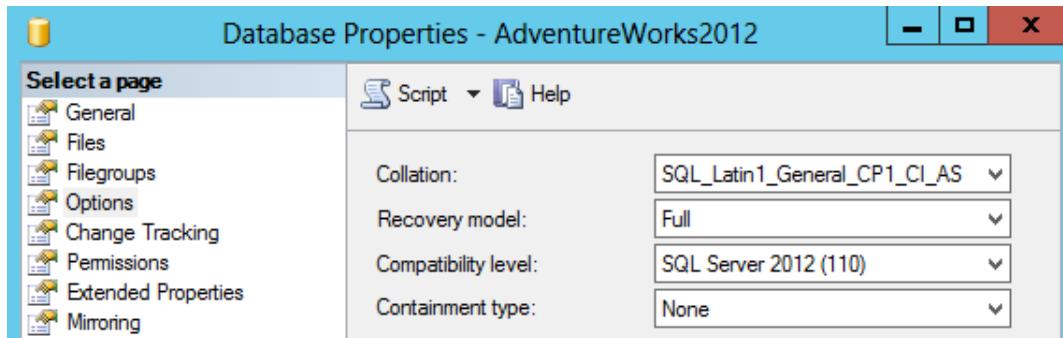
# ACTIVITY 9–1

## Backing Up an SQL Server Environment

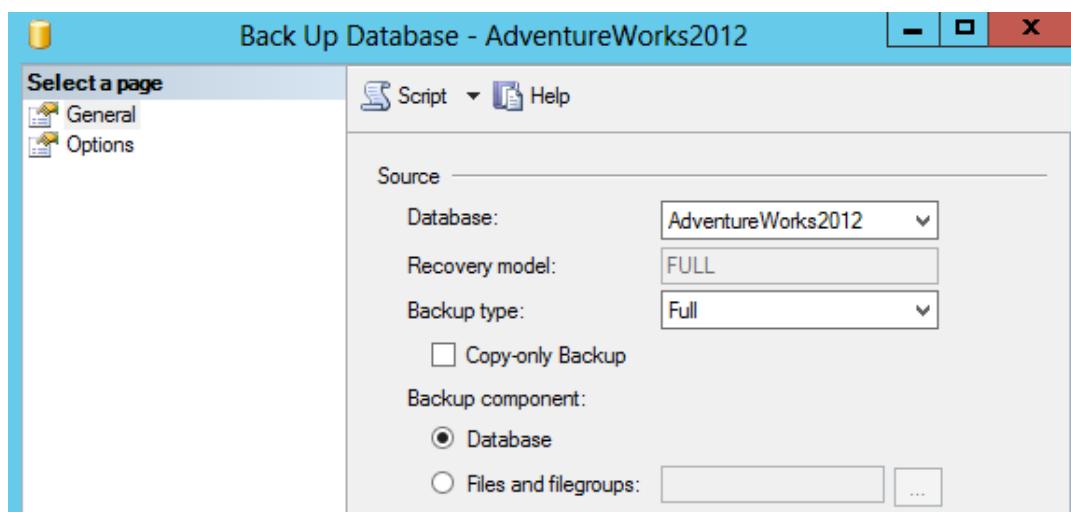
### Scenario

The Management of Adventure Works Cycles wants to ensure that data in AdventureWorks2012 is protected in case of system failures. You are asked to take a full backup of the main database.

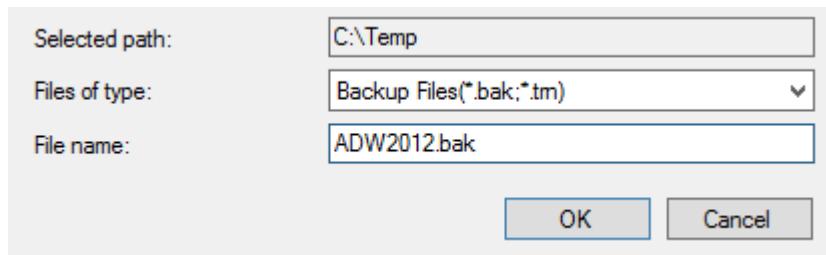
1. Set the database properties to perform a full backup.
  - a) In the **Object Explorer** pane, right-click **SQL01**→**Refresh**.
  - b) In the **Object Explorer** pane, expand **SQL01**→**Databases**.
  - c) Right-click **AdventureWorks2012** and select **Properties**.
  - d) In the Database Properties window, in the **Select a page** section, select the **Options** page.
  - e) From the **Recovery Model** drop-down list, if necessary, select **Full** and select **OK**.



2. Specify the storage location for the backup.
  - a) In the **Object Explorer** pane, right-click **AdventureWorks2012** and select **Tasks**→**Back Up**.
  - b) In the Back Up Database window, in the **Destination** section, select **Remove** to delete the current backup location.
  - c) In the Back Up Database window, from the **Database** drop-down list, ensure that **AdventureWorks 2012** is selected and from the **Backup type** drop-down list, **Full** is selected.
  - d) In the **Backup component** section, ensure that the **Database** option is selected



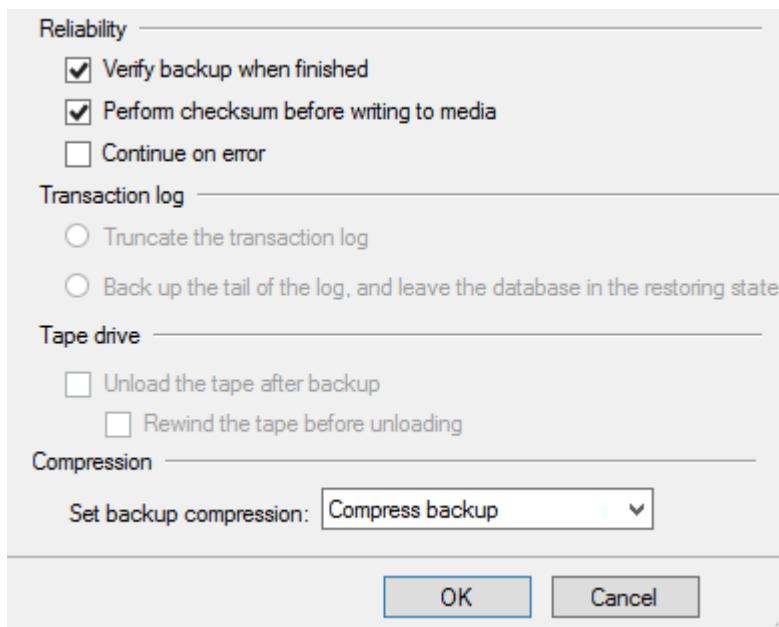
- e) In the Back Up Database window, in the **Destination** section, from the **Back up to** options, ensure that the **Disk** option is selected and select the **Add** button to add a new destination.
- f) In the **Select Backup Destination** dialog box, in the **File name** text box, select the **ellipsis (...)** button.
- g) In the Locate Database Files window, navigate to **c:\Temp** and in the **File name** text box, type **ADW2012.bak** and select **OK**.



- h) In the **Select Backup Destination** dialog box, in the **File name** text box, ensure that the file path **c:\Temp\ADW2012.bak** is displayed and select **OK**.
3. Set the reliability and compression settings to the backup.
- a) In the Back Up Database window, in the **Select a Page** section, select **Options**.
  - b) In the Back Up Database window, in the **Reliability** section, check the **Verify backup when finished** and **Perform checksum before writing to media** check boxes.
  - c) In the **Compression** section, from the **Set backup compression** drop-down list, select **Compress backup** and select **OK** to start the back up.



Select one or two participants and ask them to summarize the checksum option they have set in the reliability section.



- d) Wait for the backup to complete and in the message box that displays the success message, select OK.
4. Verify that the backup file is created in the specified location.
- Open Windows Explorer and navigate to C:\Temp.
  - In the C:\Temp folder, verify that the backup file **ADW2012.bak** is created and its size is lesser than that of the AdventureWorks2012 database because you performed a compressed backup.
  - Close the Windows Explorer window.



Verify that all the participants were able to observe that the size of the backup file ADW2012.bak is lesser than the size of the actual database.

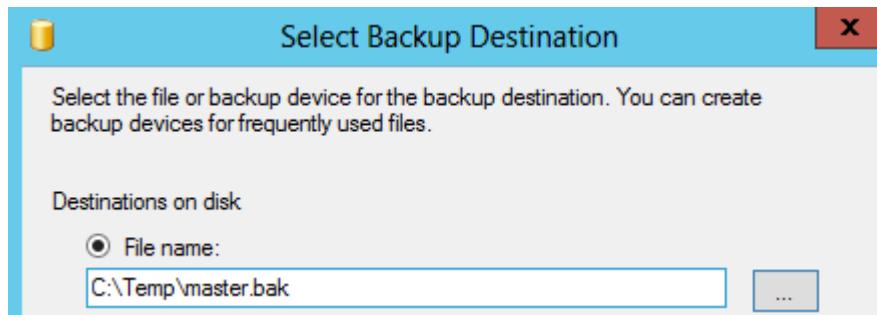
# ACTIVITY 9–2

## Backing Up System Databases

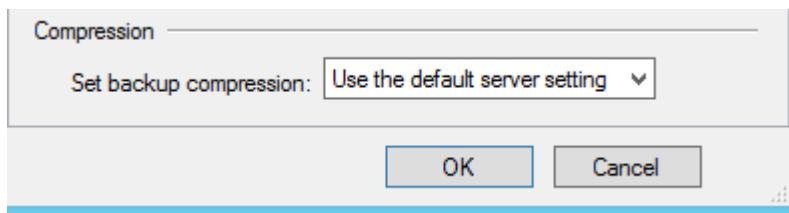
### Scenario

Management of Adventure Works Cycles wants a copy of the master database whenever a database is created, modified, or dropped. You decide to perform a full backup of the master database.

1. Set the database properties to perform a backup.
    - a) In the **Object Explorer** pane, expand **SQL01→Databases→System Databases**.
    - b) Right-click **master** and select **Properties**.
    - c) In the Database Properties window, in the **Select a page** section, select the **Options** page.
    - d) From the **Recovery Model** drop-down list, ensure that **Simple** is selected and select **OK**.
-  **Note:** The default value for the **Recovery Model** option is **Simple**.
2. Specify the backup destination for the backup.
    - a) In the **Object Explorer** pane, right-click **master** and select **Tasks→Backup**.
    - b) In the Back Up Database window, in the **Destination** section, select **Remove** to delete the current backup location.
    - c) In the Back Up Database window, from the **Database** drop-down list, ensure that **master** is selected and from the **Backup type** drop-down list, **Full** is selected.
    - d) In the **Backup component** section, ensure that the **Database** option is selected
    - e) In the Back Up Database window, in the **Destination** section, from the **Back up to** section, ensure that the **Disk** option is selected and select the **Add** button to add a new destination.
    - f) In the **Select Backup Destination** dialog box, in the **File name** text box, select the ellipsis button.
    - g) In the Locate Database Files window, navigate to **C:\Temp** and in the **File name** text box, type **master.bak** and select **OK**.
    - h) In the **Select Backup Destination** dialog box, in the **File name** text box, ensure that the file path **C:\Temp\master.bak** is displayed and select **OK**.



3. Set the reliability and compression settings the backup.
  - a) In the Back Up Database window, in the **Select a page** section, select **Options**.
  - b) In the Back Up Database window, in the **Reliability** section, check the **Verify backup when finished** and **Perform checksum before writing to media** check boxes.
  - c) In the **Compression** section, from the **Set backup compression** drop-down list, ensure that **Use the default server setting** is selected and select **OK** to start the back up.



- d) Wait for the backup to complete and in the message box that displays the success message for the backup, select **OK**.
4. Verify that the backup file is created in the specified location.
- Open Windows Explorer and navigate to **C:\Temp**.
  - In the **C:\Temp** folder, verify that the backup file **master.bak** is created.
  - Close the Windows Explorer window.

# TOPIC B

## Restore a Database

You have performed a database backup and identified the guidelines and strategies to prevent data loss. In case of server failures, you need to restore a database from a backup file. Familiarizing yourself with the restore feature of SQL Server will enable you to recover your database from the latest backed up file. In this topic, you will restore a database.

### Restore

*Restore* is the process of using a database backup to recover a failed database. It enables you bring a failed database into a running state with no or minimal loss of data. You can recover the data available from the last backup that you have taken. When you restore a backup of a database, the files and file groups as well as the transaction log of the backed up database are “recreated” in the destination server. If the files already exist, you need to either remove them or specify that you wish to overwrite the existing files at the time of restoration.

When you have a backup set, by using full and differential backups, you must first restore the full backup and then apply the differential backups in the same order that they were taken. If you have taken transaction log backups during this period, you can start restoring them in the correct order as well.

### Tail-Log Backup



Tail-Log Backup

*Tail-Log backup* is taken when the database is non-functional and there are transactions after the last backup (full, differential, or log). You need to take the Tail-Log backup before starting the restore process. This backup can be taken even on offline databases by selecting the **Backup the tail of the log** option in the **Back Up Database**→**Options** page when attempting a transaction log backup. This will ensure that even incomplete transactions will get backed up and will be rolled back when the restore process is done on this backup.

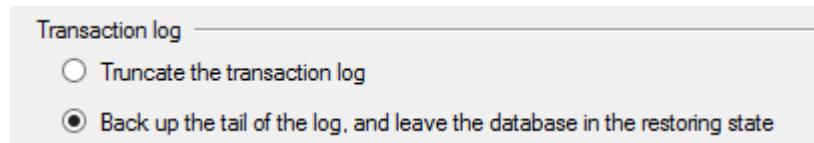


Figure 9–6: The option to activate the Tail-Log Backup in the Back Up Database window.

### Attach and Detach

Attach and Detach enables you to copy or move your databases to a new server. You can detach a database from a server by copying its .MDF and .LDF files offline and attaching them to a different server. This moves or copies the entire data and structure of the database. Attach and Detach can be used as an alternative way for backing up databases. However, it is recommended you do not back up databases using this method.

### Comparison of Attach and Restore

A comparison of Detach-Attach versus Backup-Restore is as follows.

| <b>Detach-Attach</b>                                                                       | <b>Backup-Restore</b>                                                                  |
|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Meant for moving databases.                                                                | Meant for database backups.                                                            |
| Takes database offline on source server.                                                   | Database remains online and transactions can be performed while backup is being taken. |
| Size to copy is the combined size of all .MDF and .LDF files of the database and is large. | Size to copy is smaller because only used pages are backed up.                         |
| Internal compression cannot be done.                                                       | Backups allow internal compression.                                                    |
| Always uses full size of database files.                                                   | Backups can be full, differential, or log only.                                        |
| No proper backup strategy can be maintained.                                               | Different types of strategies based on requirements can be setup.                      |
| Backups cannot be scheduled.                                                               | Backups can be scheduled for automatic processing.                                     |

The Backup-Restore component has much more features for the specified job. However, you can use Detach-Attach in special cases or for quick backups.

## Point in Time Snapshots

*Point in Time Snapshots* or database snapshots are mirror copies of the database for a particular point in time. The snapshot is created for a particular point in time and can be queried independently. You can also use the snapshot to restore the database to that point in time. You can use snapshots to quickly rollback in time for instance, when a bulk load operation was done incorrectly.

### Snapshot Restrictions

Snapshots are not meant to replace the traditional database backups and cannot be used as alternative to standard backup and restore. Snapshots cannot be used to restore a failed database.

## Filegroup Restores

SQL Server allows you to restore a single or multiple file or filegroups from a backup as well. In case a media containing one of the files/groups of the database fails, you might want to go ahead and only restore that failed file or group to a new location, rather than restoring the entire database. You can specify the file or group to be restored in the Restore Files and Filegroups window.

## Page Level Restores

Sometimes a small number of data pages might get corrupted in the database. Instead of having to perform a full restore of the database, SQL Server allows you to only restore those pages. This restore process is faster and therefore the database need not be kept offline for a long time.

You can check whether any page is currently corrupted or not. If it is, you can replace the corrupt pages with non-corrupted pages from the backups.



Ask the students about situations in which they would prefer file group restore.



Page Level Restores



Ask the students to specify the restore method that they would prefer when implementing a database backup.

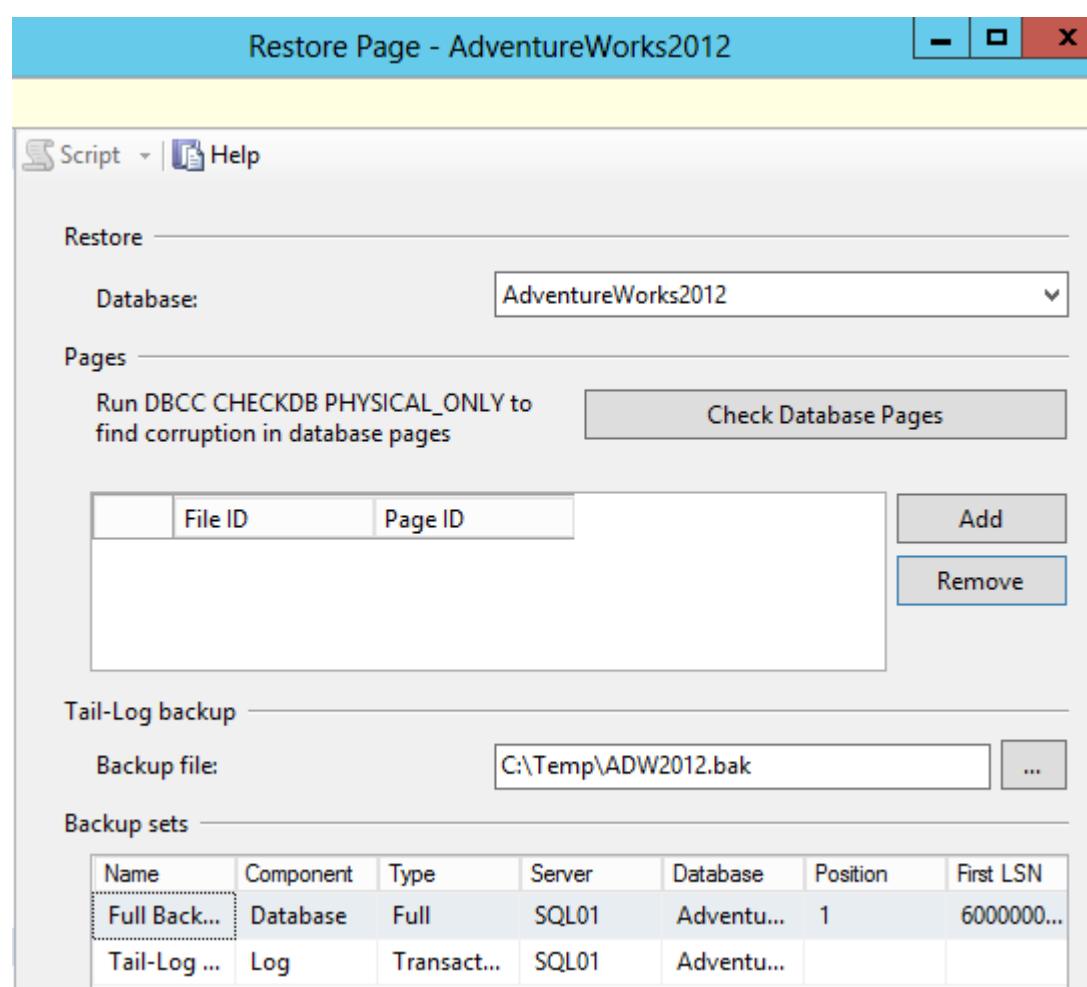


Figure 9-7: A Restore Page window in SSMS.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Restore a Database

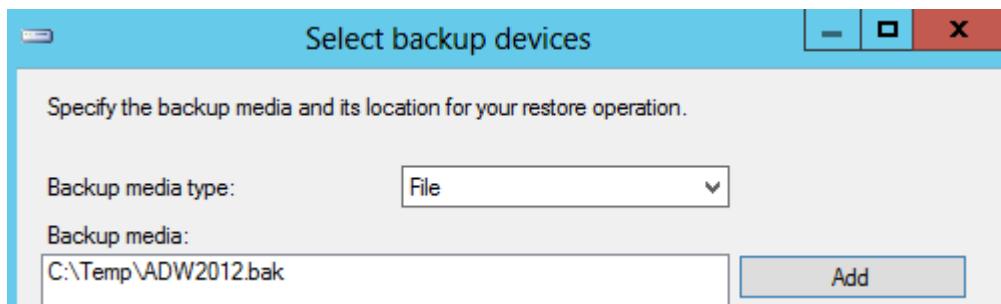
# ACTIVITY 9–3

## Restoring a Backed Up Database

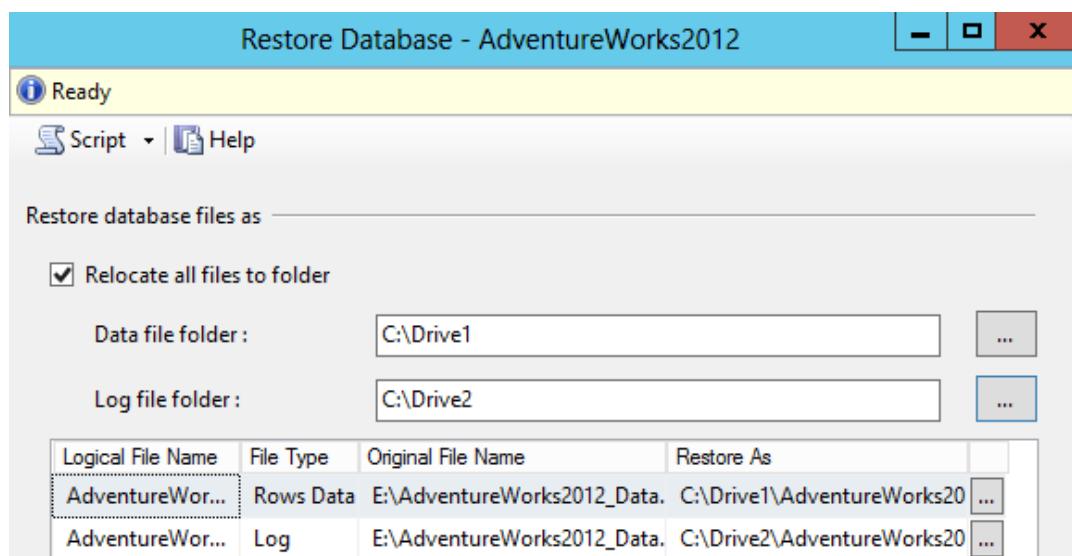
### Scenario

The Adventure Works Cycles Company witnessed an unexpected server failure that lead to loss of data in the current server. You decide to restore the backup of the AdventureWorks2012 database from another server.

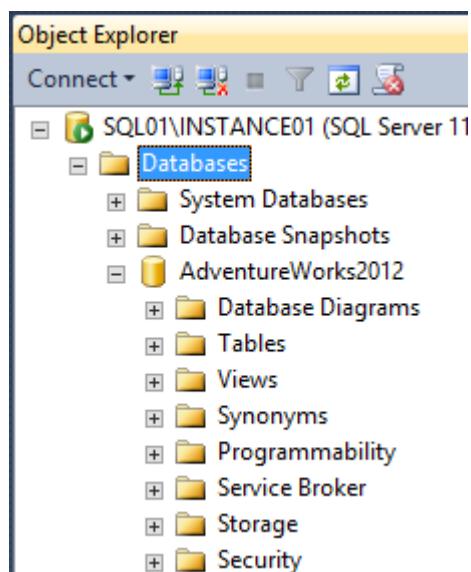
1. Connect to the server in which the AdventureWorks2012 database should be restored.
  - a) In the **Object Explorer** pane, right-click **SQL01**→**Disconnect**.
  - b) In the **Object Explorer** pane, select **Connect**→**Database Engine**.
  - c) In the **Connect to Server** dialog box, from the **Server name** drop-down list, select **SQL01\INSTANCE01** and from the **Authentication** drop-down list, select **Windows Authentication**.
  - d) In the **Connect to server** dialog box, select **Connect**.
2. Select the backup file to be restored.
  - a) In the **Object Explorer** pane, right-click **Databases** and select **Restore Database**.
  - b) In the **Restore Database** window, in the **Source** section, select the **Device** option and select the ellipsis button.
  - c) In the **Select backup devices** dialog box, select the **Add** button.
  - d) In the **Locate Backup File** window, navigate to **C:\Temp**, select **ADW2012.bak**, and select **OK**.



- e) In the **Select backup devices** window, select **OK**.
- f) In the **Restore Database** window, observe that the name of the database is automatically filled.
3. Relocate the files to be restored in the **C:\Temp** folder.
  - a) In the **Restore Database** window, in the **Select a page** section, select **Files**.
  - b) In the **Files** page, check the **Relocate all files to folder** check box.
  - c) In the **Data file folder** text box, select the ellipsis button and navigate to **c:\Drive1**.
  - d) In the **Log file folder** text box, select the ellipsis button and navigate to **c:\Drive1** and select **OK**.



- e) In the Restore Database window, select **OK** to start the restore.
  - f) Wait for the restore to complete and in the message box that displays the success message, select **OK**.
4. Verify that the database is restored in SSMS.
- a) In the **Object Explorer** pane, expand **SQL01\INSTANCE01→Databases→AdventureWorks2012** and ensure that all the tables and data are restored in the AdventureWorks2012 database.



- b) In the **Object Explorer** pane, right-click **SQL01\INSTANCE01→Disconnect**.

# ACTIVITY 9–4

## Restoring to a Point in Time

### Scenario

The database manager of Adventure Works Cycles wants to restore a recent backup of the AdventureWorks2012 database that was taken before the last backup. You are asked to restore the database to the specified point in time. You decide to perform a point in time restore.

1. Backup the AdventureWorks2012 database.
  - a) In the **Object Explorer** pane, select **Connect→Database Engine**.
  - b) In the **Connect to Server** dialog box, from the **Server name** drop-down list, select **SQL01**.
  - c) In the **Connect to Server** dialog box, select **Connect**.
  - d) In the **Object Explorer** pane, expand **SQL01→Databases**.
  - e) Right-click **AdventureWorks2012** and select **Tasks→Back Up**.
  - f) In the **Back Up Database** window, in the **Destination** section, ensure that the file path **C:\Temp\ADW2012.bak** is displayed and select **OK**.
  - g) In the message box that displays the success message, select **OK**.
  - h) In the **Object Explorer** pane, right-click **SQL01→Disconnect**.
  
2. Restore the previous backup.
  - a) In the **Object Explorer** pane, select the **Connect→Database Engine**.
  - b) In the **Connect to Server** dialog box, from the **Server name** drop-down list, select **SQL01\INSTANCE01** and from the **Authentication** drop-down list, verify that **Windows Authentication** is selected.
  - c) In the **Connect to Server** dialog box, select **Connect**.
  - d) In the **Object Explorer** pane, expand **SQL01\INSTANCE01→Databases**.
  - e) Right-click **AdventureWorks2012** and select **Tasks→Restore→Database**.
  - f) In the **Restore Database** window, in the **Source** section, select the **Device** option and select the **ellipsis (...)** button.
  - g) In the **Select backup devices** dialog box, select the **Add** button.
  - h) In the **Locate Backup File** window, navigate to **C:\Temp**, select **ADW2012.bak**, and select **OK**.
  - i) In the **Select backup devices** window, select **OK**.
  - j) In the **Restore Database** window, in the **Backup sets to restore** section, observe that the name of the database is automatically populated.

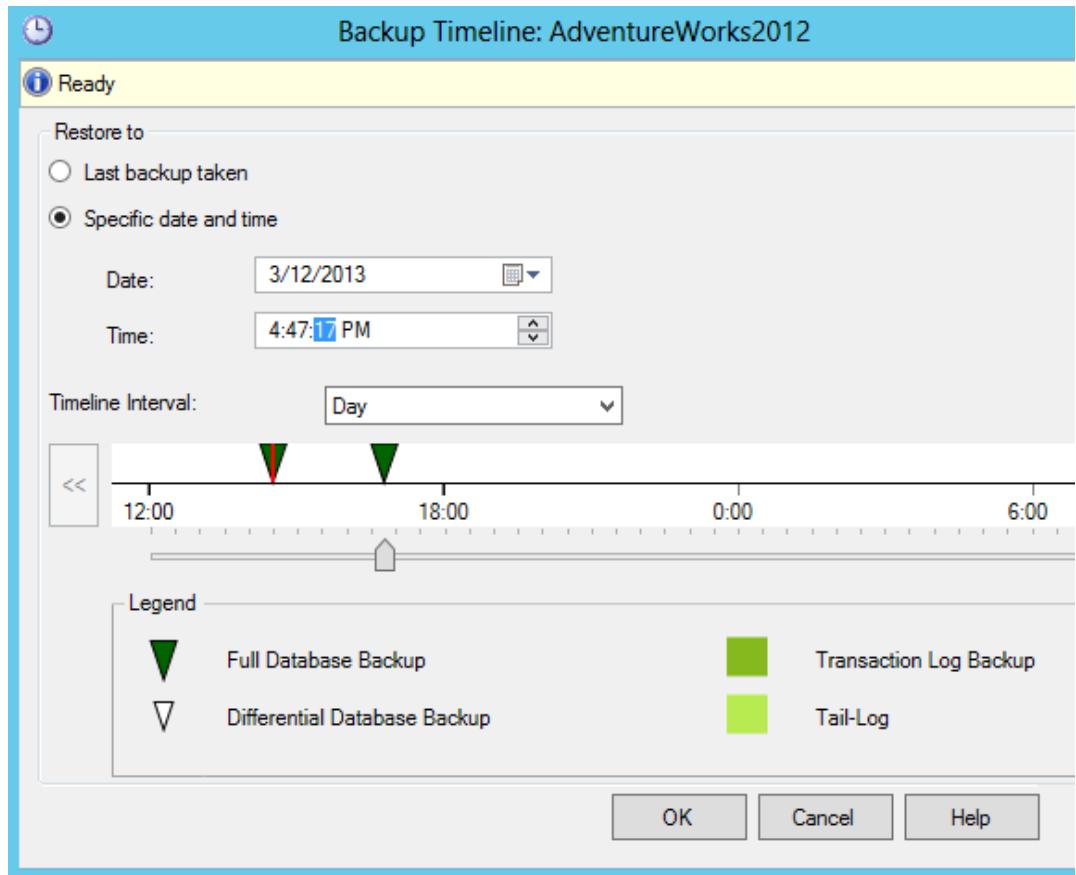
| Backup sets to restore:             |                                         |           |      |        |                    |
|-------------------------------------|-----------------------------------------|-----------|------|--------|--------------------|
| Restore                             | Name                                    | Component | Type | Server | Database           |
| <input checked="" type="checkbox"/> | AdventureWorks2012-Full Database Backup | Database  | Full | SQL01  | AdventureWorks2012 |

3. Specify the date and time for restoring the database to a point in time.
  - a) In the **Restore Database** window, in the **Destination** section, select the **Timeline** button.
  - b) In the **Backup Timeline** window, observe that the **Last backup taken** option is selected.
  - c) Select the **Specific date and time** option and observe that the **Date** and **Time** boxes show the date and time of the last backup taken.



Remind students that the backup taken in the previous step is shown as the last backup taken.

- d) In the Backup Timeline window, in the **Time** spin box, select the seconds value and select the down arrow to decrease the time value by one second and select **OK**.



- e) In the Restore Database window, in the **Backup sets to restore** section, ensure that only the backups to the point in time specified are selected and select **OK**.
- f) In the message box displaying the restore success message, select **OK**.
4. Verify the database restored in SSMS and close the connection.
- In the **Object Explorer** pane, right-click **SQL01\INSTANCE01**→**Refresh**.
  - In the **Object Explorer** pane, expand **SQL01\INSTANCE01**→**Databases**→**AdventureWorks2012** and ensure that all the tables and data are restored in the AdventureWorks2012 database.
  - Right-click **SQL01\INSTANCE01** and select **Disconnect** to close the connection.

# TOPIC C

## Import and Export Data

You have restored the required database to ensure that the backups are always available. When working with large volume of organizational data, there may be circumstances where you may need to transfer the data to different locations. SQL Server 2012 provides you various features such as Import and export data wizard, BULK COPY and BULK INSERT commands to transfer your data to the required location. Better understanding on these data transfer techniques will enable you to effectively manage your organizational data. In this topic, you will import and export data.

### Import and Export Wizard

In SQL Server, data transfer can be performed by three techniques. They are:

- Import and Export Wizard
- BULK COPY
- BULK INSERT



Import and Export Wizard

The Import and Export Wizard allows you to copy data to and from any data source in SQL Server. You can use the Import and Export Wizard to transfer the data to a required location. You can access this wizard through SSMS using **Tasks→Export Data** or **Tasks→Import Data**. It is important to provide a valid destination and source location in the Import and Export Wizard to ensure that the data is transferred to the correct location.

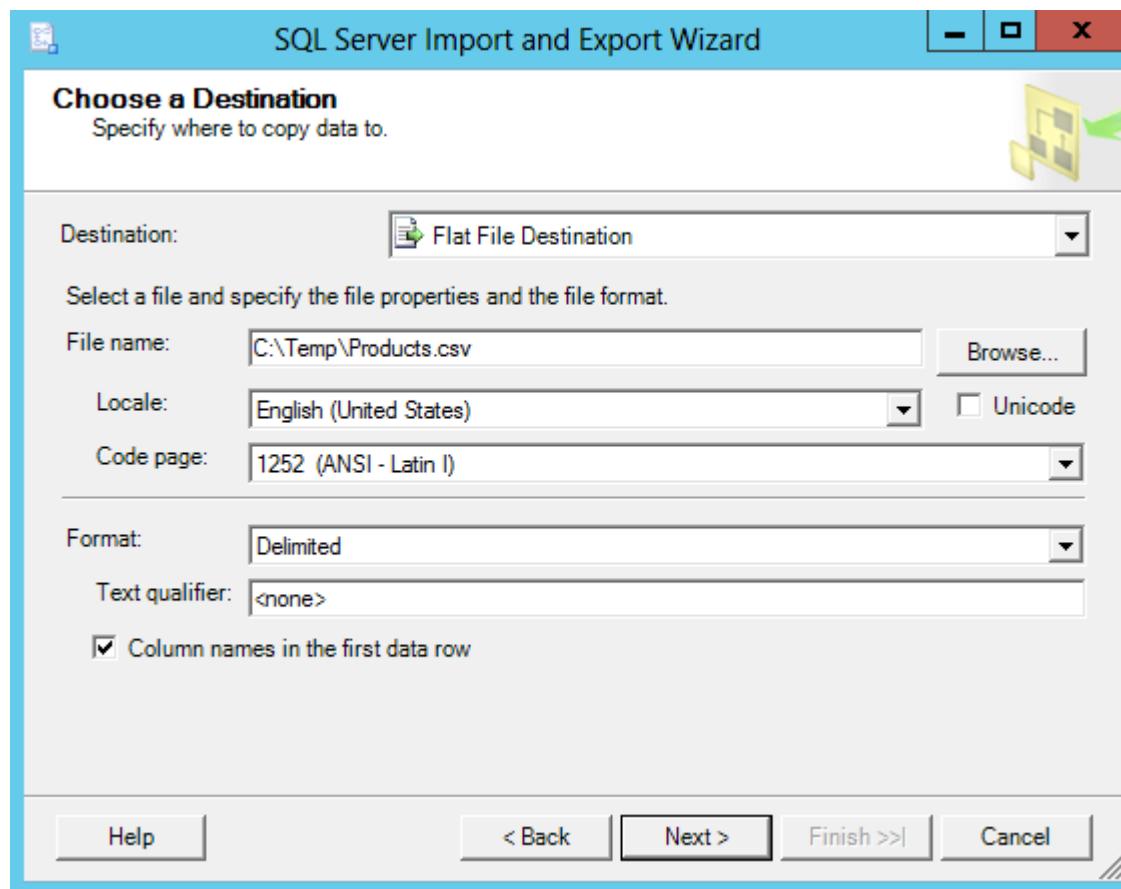


Figure 9-8: The SQL Server Import and Export Wizard is used to export and import data.

## The BULK COPY Command

The *BULK COPY* in SQL Server is performed using a utility tool called BULK COPY Program (BCP). The BCP is a command line tool that allows you to export and import data.

There are situations when you might need to copy a lot of data at once to an SQL Server database. For instance, let's say that you have setup replication between databases hosted in your central HQ and a branch office. For some reason, the data entered in the branch office database did not get replicated across to the HQ database. You can obtain the data from the branch office using a BULK COPY export and then import it back.

The syntax to declare a BCP command is as follows.

```
bcp [<database.schema.table> | <SQL
 query>] [in | out] <filename><options>
```

In this syntax, you can specify a database table or query to export or the table to import into. The “in” or “out” specify whether an import or export is to be performed. The filename is the file being imported from or exported to. And the options specify a number of different options you can use for formatting the information.

### Example

A bcp command to export the data from the AdventureWorks2012 database is as follows.

```
bcp AdventureWorks2012.Production.ProductCategory out c:\Temp
\ADW.Production.ProductCategory.dat
-n -T
```

This bcp command will export the data from the AdventureWorks2012.Production.ProductCategory table into a .dat file named ADW.Production.ProductCategory and save the exported backup into the Temp folder of the C:\ drive. The option “-n” specifies a native format (binary) and the “-T” specifies a trusted connection to the server.

## The BULK INSERT Command

The *BULK INSERT* command is a T-SQL command that allows you to insert a large number of rows at a time into a table. This is functionally equivalent to using the “in” option of the BCP utility – except that in this case the SQL Server process itself manages the connection and data import process.

In cases where you need to import large amounts of data, you might find it beneficial using this rather than the BCP “in” command. Also, you can use this in a T-SQL script quite easily. For instance, you might want to create a script that changes the Recovery Model of the database to “Bulk Logged”, you can perform the import using the BULK INSERT and change the recovery model back to full for the database.

The syntax to use the BULK INSERT command is as follows.

```
BULK INSERT<database.schema.table>
 FROM <data file path> WITH <options>;
```

In this syntax, you have to specify the database schema or table name along with the data file path from which the data need to be imported.

The DATAFILETYPE and FIELDTERMINATOR are the most commonly used keywords in a BULK INSERT command. The DATAFILETYPE keyword is used to define the kind of file you are importing from and the FIELDTERMINATOR keyword is used to define what the field terminator character is.

### Example

A BULK INSERT command to import data into AdventureWorks database is as follows.

```
BULK INSERT
AdventureWorks2012.Production.ProductCategory FROM
'c:\Temp\ADW.Production.ProductCategory.dat' WITH DATAFILETYPE=
'native';
```

This code snippet will import the data stored in the .dat file named ADW.Production.ProductCategory to the AdventureWorks2012.Production.ProductCategory table. The DATAFILETYPE option should be declared as native to import data from a data file in native format.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Import and Export Data

# ACTIVITY 9–5

## Transferring Data Using BULK COPY

### Before You Begin

1. In Object Explorer pane, select Connect→Database Engine.
2. In the Connect to Server dialog box, in the Server Name text box, select **SQL01** and select Connect.
3. Open a new Query Editor pane and ensure that the **SalesDB** database is selected.

### Scenario

The UK branch office of Adventure Work Cycles is having some network issues and is not able to replicate data entered in their local database back to the headquarters. You are asked to send the updates in the UK database back to the headquarters so that the HQ can import it in. You decide to bulk export all transactions that have been done recently since the last replication.

1. Execute the code to insert 14 new records into the SalesOrders table.
  - a) In the **Query Editor** pane, enter the INSERT INTO statement followed by the table name **dbo.SalesOrders**.

```
INSERT INTO dbo.SalesOrders
```

  - b) Type the field names **SalesDate**, **ProductID**, **ListPrice**, **Quantity**, and **TotalSalesAmount** to which you want to insert values.

```
INSERT INTO dbo.SalesOrders (SalesDate, ProductID, ListPrice, Quantity,
TotalSalesAmount)
```

  - c) Enter the **Values** keyword followed by the values that need to be inserted into the SalesOrder table.



**Note:** You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in **C:\093108Data\Code.txt**.

```
INSERT INTO dbo.SalesOrders (SalesDate, ProductID, ListPrice, Quantity,
TotalSalesAmount)
VALUES
('2013-01-15',1, 10.00, 3, 30.00),
('2013-01-16',2, 15.00, 2, 30.00),
('2013-01-16',1, 10.00, 2, 20.00),
('2013-01-17',1, 10.00, 5, 50.00),
('2013-01-18',2, 15.00, 3, 45.00),
('2013-01-21',2, 15.00, 1, 15.00),
('2013-01-22',2, 15.00, 3, 45.00),
('2013-01-23',3, 20.00, 3, 60.00),
('2013-01-28',3, 20.00, 5, 100.00),
('2013-02-02',1, 10.00, 2, 20.00),
('2013-01-03',2, 15.00, 2, 30.00),
('2013-01-04',3, 20.00, 3, 60.00),
('2013-01-05',1, 10.00, 2, 20.00),
('2013-01-10',1, 10.00, 4, 40.00);
```

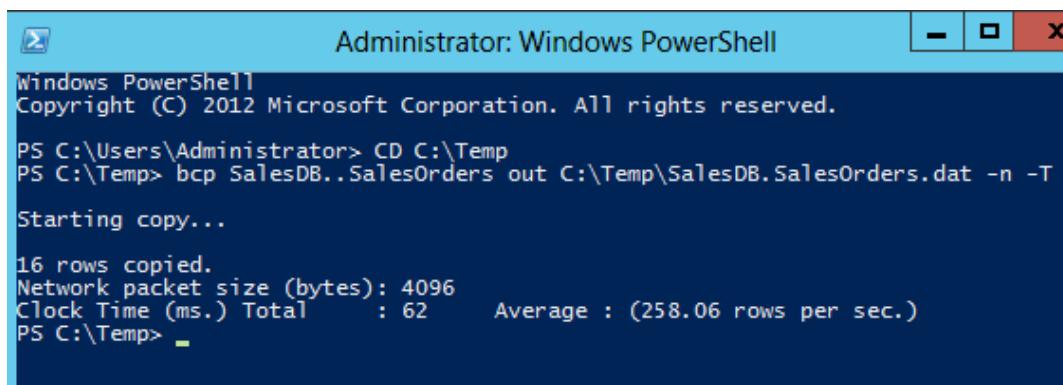
- d) Execute the statement to insert values.
  - e) On the **Messages** tab, observe that a message stating '(14 row(s) affected)' will be displayed.
2. Transfer data using BULK COPY.

- a) From the Start Screen of Windows Server 2012, select **Windows PowerShell**  to open the Command Prompt window.
- b) In the Administrator:Windows PowerShell command prompt, type **CD** followed by the folder path C:\Temp to change the directory to the C:\ Temp folder, and then press **Enter**.

```
CD C:\Temp
```

- c) Type the bcp command followed by the database and table name SalesDB and SalesOrders.
- ```
bcp SalesDB..SalesOrders
```
- d) Type the 'out' keyword followed by the path c:\Temp\SalesDB.SalesOrders.dat to which the data need to get transferred along with "-n" and "-T" representing native format and trusted connection to the server, and then press **Enter**.

```
bcp SalesDB..SalesOrders out c:\Temp\SalesDB.SalesOrders.dat -n -T
```



The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command PS C:\Users\Administrator> bcp SalesDB..SalesOrders out c:\Temp\SalesDB.SalesOrders.dat -n -T is entered and executed. The output shows the progress of the copy operation, stating "Starting copy...", "16 rows copied.", and providing network statistics: "Network packet size (bytes): 4096", "Clock Time (ms.) Total : 62", and "Average : (258.06 rows per sec.)".

- e) Enter the Notepad statement followed by the file named 'SalesDB.SalesOrders.Dat', and then press **Enter**. In the notepad file, observe that the exported file is in a binary (native) format and you will need to re-export the data in character format so that you can view it.

```
Notepad SalesDB.SalesOrders.Dat
```



The screenshot shows a Notepad window titled "SalesDB.SalesOrders.dat - Notepad". The content of the file is displayed in binary format, showing various characters and symbols.

- f) Close the Notepad file. In the Administrator:Windows PowerShell command prompt, type the bcp command followed by the database and table name SalesDB and SalesOrders.

```
bcp SalesDB..SalesOrders
```

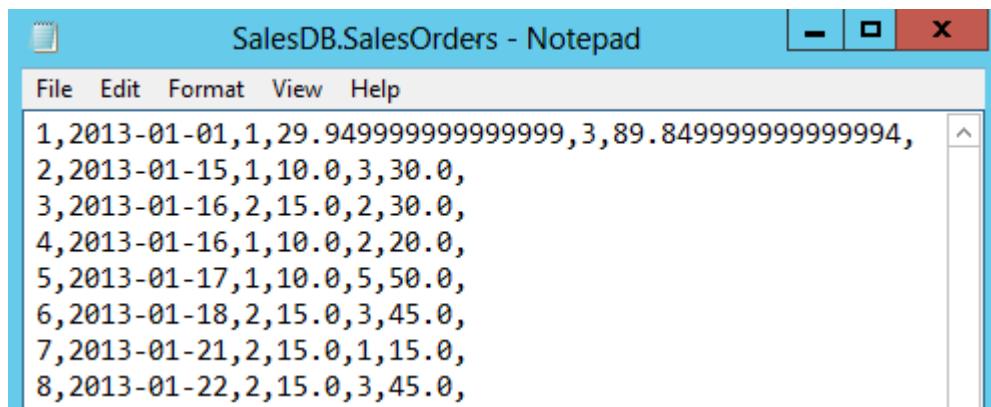
- g) Type the out keyword followed by the pathc:\Temp\SalesDB.SalesOrders.csv, along with the keyword -c to convert as character, -t and field terminator keyword as comma (-t ",") to re-export the data, and then press **Enter**.

```
bcp SalesDB..SalesOrders out c:\Temp\SalesDB.SalesOrders.csv -c -t "," -T
```

- h) Enter the Notepad statement followed by the file named 'SalesDB.SalesOrders.csv', and then press **Enter**. This will open a notepad file with data in a readable format.

- i) View the result and close the Notepad file.

```
Notepad SalesDB.SalesOrders.csv
```



- j) Close the Windows PowerShell window.
-

ACTIVITY 9–6

Transferring Data Using BULK INSERT

Before You Begin

1. Ensure that **SQL01** is connected.
2. Open a **Query Editor** pane and ensure that the **SalesDB** database is selected.

Scenario

The branch office of Adventure Works Cycles in UK had sent you the exported data in CSV format. You are asked to update this data into your table using BULK INSERT. Before updating the data, you decide to validate the integrity of the data by creating a copy of the table and performing a BULK INSERT operation in it.

1. Execute the code to create a table named **dbo.SalesOrdersCopy**.

- a) In the Query Editor window, enter the CREATE TABLE statement followed by the table name **dbo.SalesOrdersCopy**.

```
CREATE TABLE dbo.SalesOrdersCopy (
```

- b) Enter the following columns or fields that need to be created along with the table with the specified data types.

- SalesOrderID column with integer data type, IDENTITY value of 1,1, and NOT NULL constraint
- SalesDate column with date data type and NOT NULL constraint
- ProductID column with integer data type and NOT NULL constraint
- ListPrice column with float data type and NOT NULL constraint
- Quantity column with integer data type and NOT NULL constraint
- TotalSalesAmount column with float data type and NOT NULL constraint
- CustomerID column with int data type



Note: You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```
CREATE TABLE dbo.SalesOrdersCopy (
    SalesOrderID int IDENTITY(1,1) NOT NULL,
    SalesDate date NOT NULL,
    ProductID int NOT NULL,
    ListPrice float NOT NULL,
    Quantity int NOT NULL,
    TotalSalesAmount float NOT NULL,
    CustomerID int
```

- c) Enter the CONSTRAINT keyword followed by the constraint name **PK_SalesOrdersCopy** along with the PRIMARY KEY NONCLUSTERED keyword.

```
"...
ListPrice float NOT NULL,
    Quantity int NOT NULL,
    TotalSalesAmount float NOT NULL,
CustomerID int,
    CONSTRAINT PK_SalesOrdersCopy PRIMARY KEY NONCLUSTERED
```

- d) Enter the SalesOrderID and SalesDate columns with ASC keyword to arrange the data in these columns in ascending order.

```
"...
TotalSalesAmount float NOT NULL,
CONSTRAINT PK_SalesOrdersCopy PRIMARY KEY NONCLUSTERED
(
    SalesOrderID ASC,
    SalesDate ASC
) );
```

- e) Execute the statement to create a new table.
- f) On the **Messages** tab, observe that the message "Command(s) completed successfully" is displayed.

2. Execute the code to perform BULK INSERT into the SalesOrdersCopy table.

- a) Open a **Query Editor** pane and enter the BULK INSERT statement followed by the table name SalesDB..SalesOrdersCopy.

```
BULK INSERT SalesDB..SalesOrdersCopy
```

- b) Enter the FROM clause followed by the path 'c:\Temp\SalesDB.SalesOrders.csv' in which the bulk insert data is stored.

```
BULK INSERT SalesDB..SalesOrdersCopy
FROM 'c:\Temp\SalesDB.SalesOrders.csv'
```

- c) Enter the WITH keyword followed by the DATAFILETYPE as character and FIELDTERMINATOR keyword as comma.

```
BULK INSERT SalesDB..SalesOrdersCopy
FROM 'c:\Temp\SalesDB.SalesOrders.csv'
WITH (
    DATAFILETYPE='char',
    FIELDTERMINATOR=','
);
```

- d) Execute the BULK INSERT statement.
- e) On the **Messages** tab, observe that a message stating the number of rows affected is displayed.

3. Verify the bulk insert in the SalesOrdersCopy table.

- a) In the **Object Explorer** pane, expand **Databases**, right-click **SalesDB** and from the fly-out pane, select **Refresh**.
- b) Expand the **Tables** folder, and then right-click **dbo.SalesOrdersCopy** table and from the fly-out pane, select **Select Top 1000** rows. This will display the inserted records in the **Results** tab of the **Query Editor** pane.

SQLQuery16.sql - S...Administrator (58))* X SQLQuery14.sql - S...Administrator (55))*

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [SalesOrderID]
    ,[SalesDate]
    ,[ProductID]
    ,[ListPrice]
    ,[Quantity]
    ,[TotalSalesAmount]
    ,[CustomerID]
FROM [SalesDB].[dbo].[SalesOrdersCopy]
```

100 % < |

	SalesOrderID	SalesDate	ProductID	ListPrice	Quantity	TotalSalesAmount
1	1	2013-01-01	1	29.95	3	89.85
2	2	2013-01-15	1	10	3	30
3	3	2013-01-16	2	15	2	30
4	4	2013-01-16	1	10	2	20
5	5	2013-01-17	1	10	5	50
6	6	2013-01-18	2	15	3	45

- c) Close the **Query Editor** pane without saving the query.

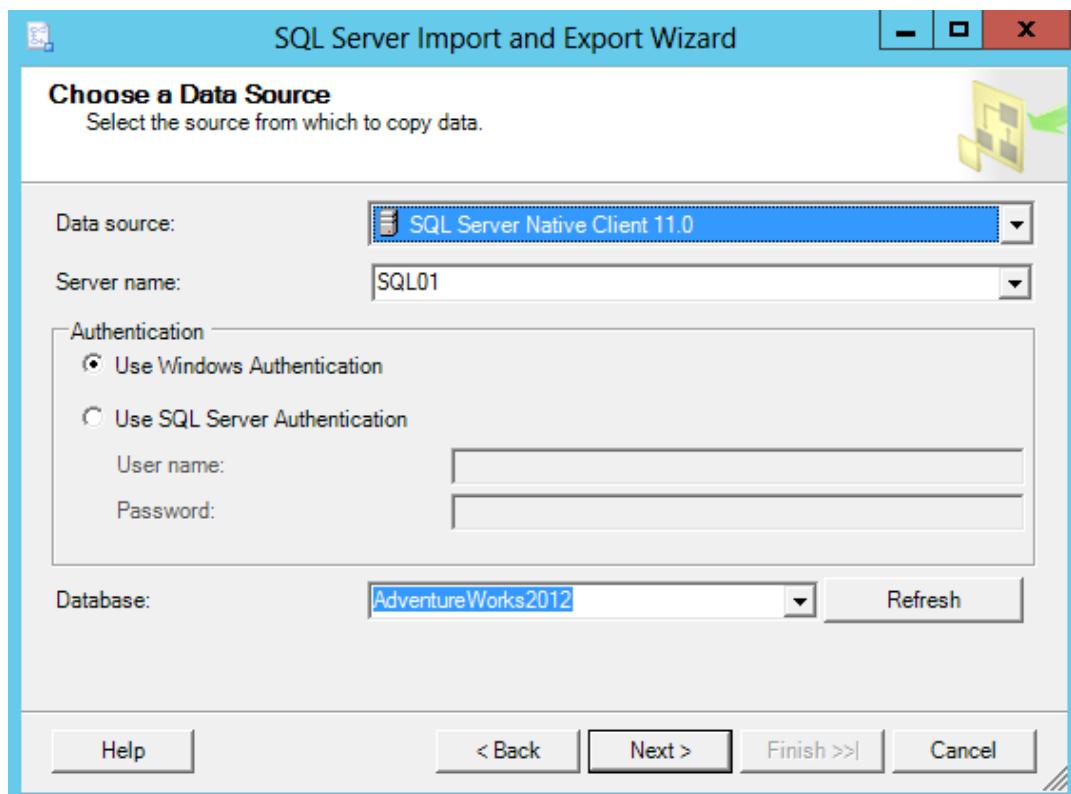
ACTIVITY 9–7

Exporting Data

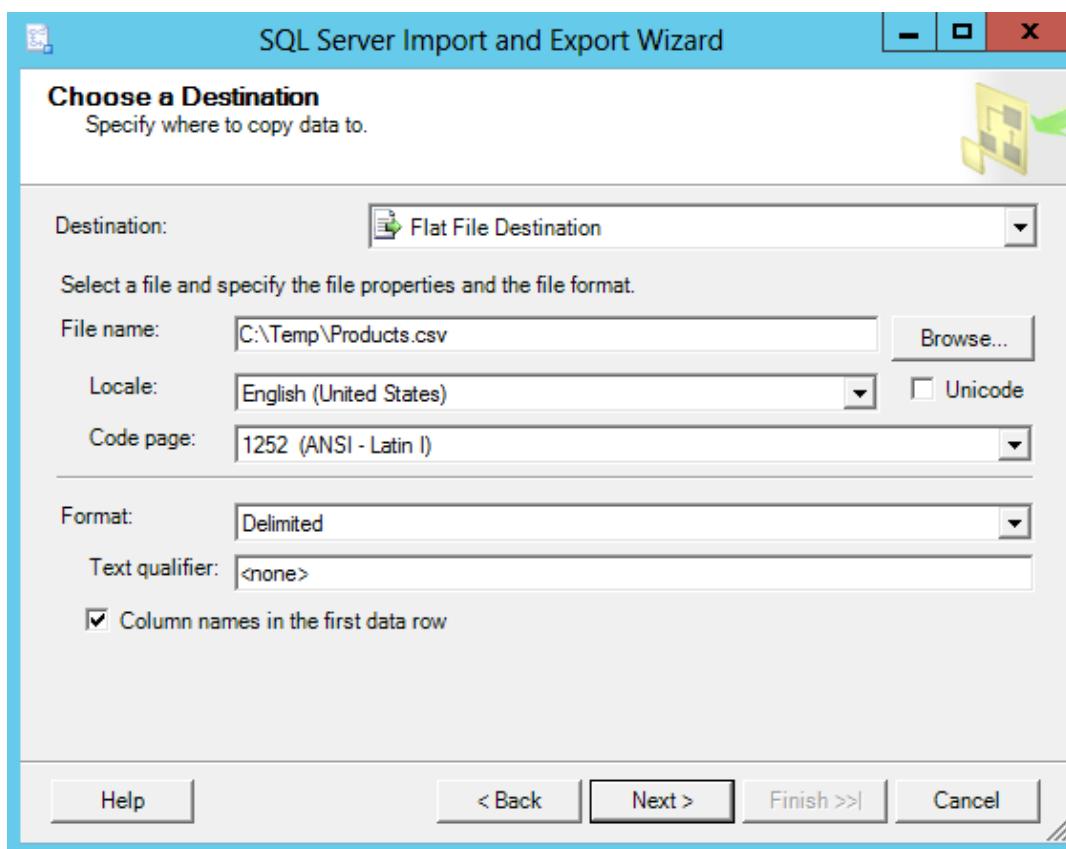
Scenario

The sales team of Adventure Works Cycles wants to export the products that are in production to a new location. You are asked to support the sales team in exporting the required data. You decide to use the Import and Export Wizard to perform this task.

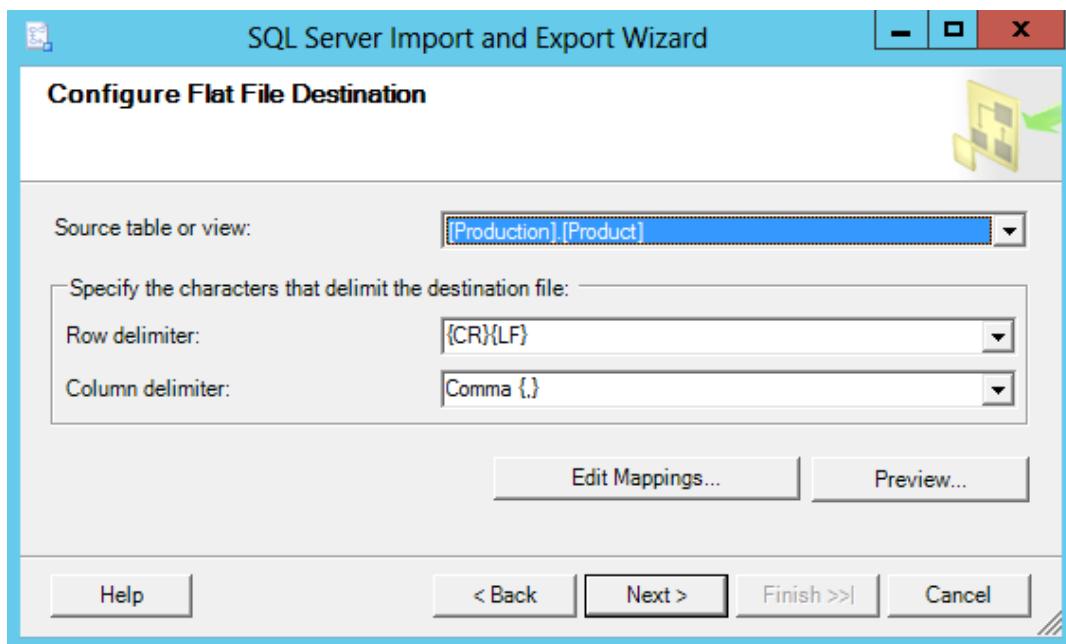
1. Launch the SQL Server Import and Export Wizard.
 - a) In the **Object Explorer** pane, right-click the **AdventureWorks2012** database.
 - b) Select **Tasks→Export Data** and select **Next**.
 - c) In **SQL Server Import and Export Wizard**, in the **Choose a Data Source** page, in the Database drop-down list, ensure that the **AdventureWorks2012** is selected and select **Next**.



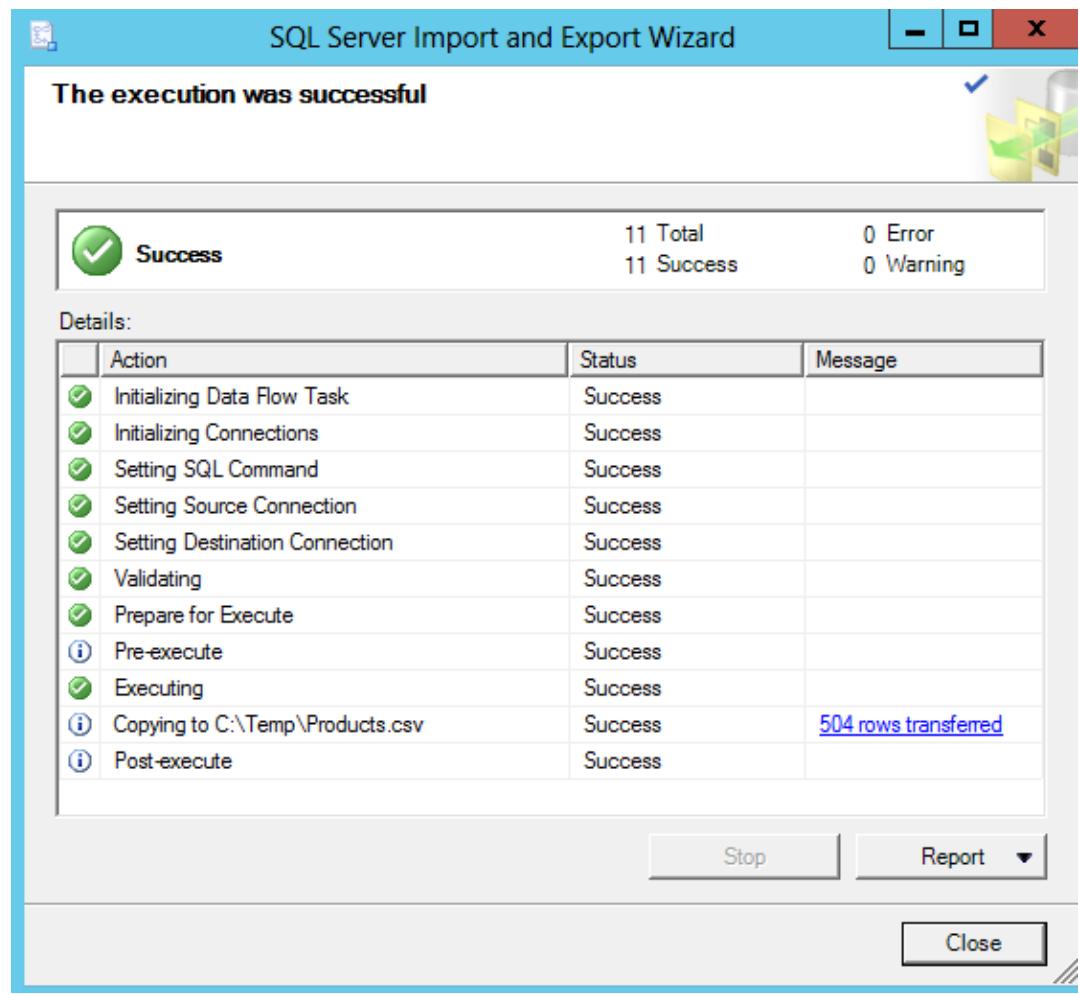
2. Choose the destination for the exported data to be stored.
 - a) In the **Choose a Destination** page, from the **Destination** drop-down list, select **Flat File Destination**.
 - b) In the **Filename** text box, type **C:\Temp\Products.csv**
 - c) In the **Format** drop-down list, ensure that **Delimited** is selected.
 - d) Ensure that the **Column names in the first data row** check box is checked and select **Next**.



3. Export the Product.Product table.
 - a) On the **Specify Table Copy or Query** page, ensure that the **Copy data from one or more tables or views** option is selected and select **Next**.
 - b) On the **Configure Flat File Destination** page, from the **Source table or view** drop-down list, select **[Production].[Product]** and select **Next**.



- c) On the **Save and run package** page, ensure that the **Run immediately** check box is checked and select **Next**.
- d) On the **Complete the Wizard** page, select **Finish**.
- e) Select **Close** when the execution is completed.



TOPIC D

Configure the SQL Server Agent

You have restored the required database to ensure that the backups are always available. However, to ensure that the data loss is minimal or non-existent, you need to perform backups regularly and frequently. Performing regular backups manually is extremely time consuming and waste of resources. This is where a SQL Server component called the SQL Server Agent comes in. Familiarizing yourself with configuring the SQL Server Agent will enable you to automate and secure your backups. In this topic, you will configure SQL Server agent.

SQL Server Agent

SQL Server Agent is a SQL Server component that allows you to setup administrative tasks, called jobs, and run them on a schedule. SQL Server Agent allows you to automate a number of regular maintenance functions rather than performing them manually.

The following components are available in the SQL Server Agent:

SQL Server Agent Component	Description
Jobs	A job is the actual task that needs to be executed on the server agent.
Schedules	A schedule is a time at which a job can run. A schedule can contain more than one job. A job needs to be a part of at least one schedule.
Alerts	An alert is the notification generated based on a particular condition of a job or schedule. When a particular event occurs, the alert is triggered and can do one of two things – inform an operator or start another job.
Operators	An operator is a person responsible for managing a job and its schedules. The contacts defined in Agents contain the contact information of the operators. The Alerts component uses these contacts to send status information to the operators when a particular event occurs. The operators are not security principals and cannot be used to perform the login operation on the server.



Note: For more information about SQL Server Agent, check out the LearnTO **Configure SQL Server Agents** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.



You may want to show LearnTO **Configure SQL Server Agents** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



Agent Jobs

Agent Jobs

An *Agent job* is defined as a series of steps each executing one task. The job runs based on a schedule, event (Alert), or manually.

For instance, you can create a job that takes a backup of the database, truncates the log, and exports a data into a report. You can set the job to run daily at 6AM using a schedule and once it is complete, an alert with the report attached can be sent to an operator. The job is the most important aspect of the SQL Server Agent component.

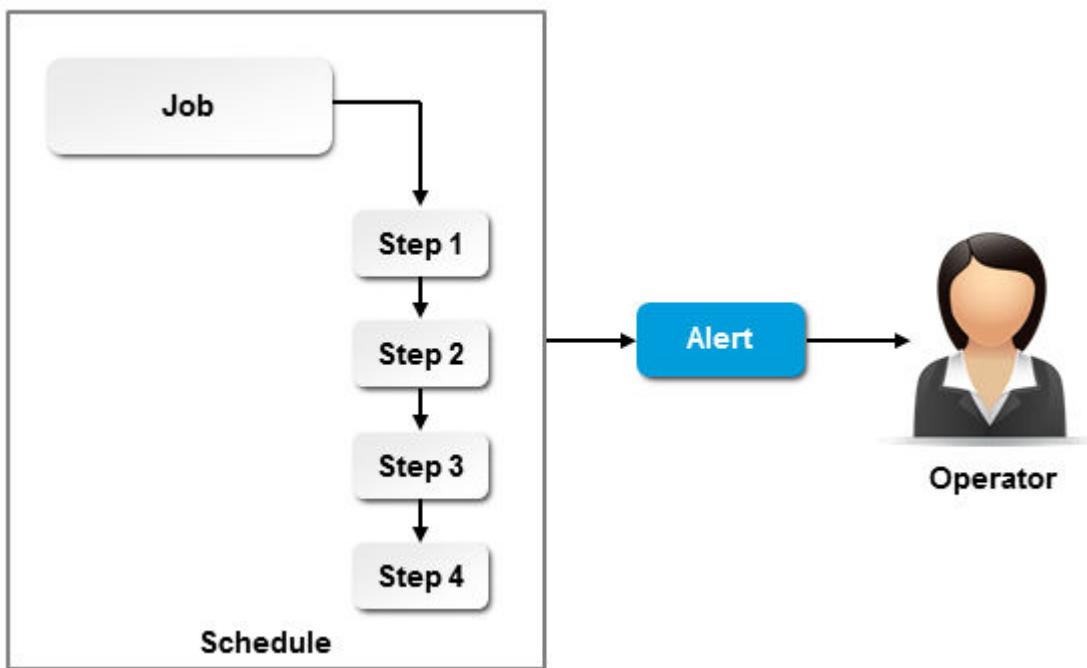


Figure 9–9: An alert sent to an operator after a scheduled job is completed.



Ask the students the type of tasks that they would automate using server agent jobs.



Job History

Job History

Job history is a log maintained for each job defined in the system. The job history feature enables you to check the status of a job you configured on the server.

A job history displays the job, status, date and times it was run. You can even drill-down to each job step in the defined job and see the status of each step in the job.

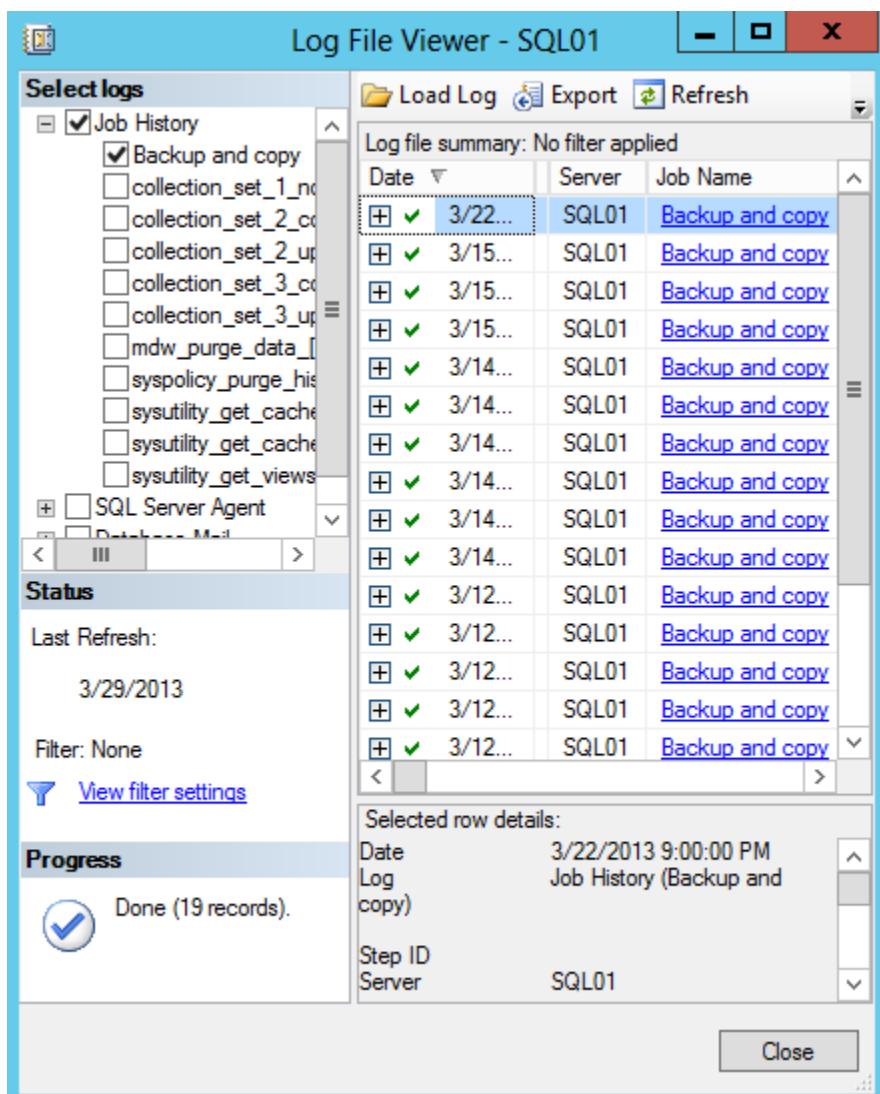


Figure 9-10: The Log File Viewer displaying a job history.

You can configure job history to limit the number of total history items to be stored for all jobs defined on the server. You can also limit the number of history items to be stored for each job. You can purge the old history items by specifying the duration for the history items to be stored. For instance, you can purge all job histories that are older than 15 days or 3 weeks. You can configure job history through SSMS using the **SQL Server Agent Properties→History** page.

SQL Server Agent Security

SQL Server Agent is implicitly secured using database roles. To be able to view or configure the jobs and components of the Agent, a user has to be a member of one of the following roles:

- SQLAgentRole
- SQLAgentReaderRole
- SQLAgentOperatorRole
- sysadmin



SQL Server Agent Security

SQL Server also uses a concept of *subsystem*, which is a listing of different types of job steps. Only the users of the subsystem can access the job steps in the subsystem.

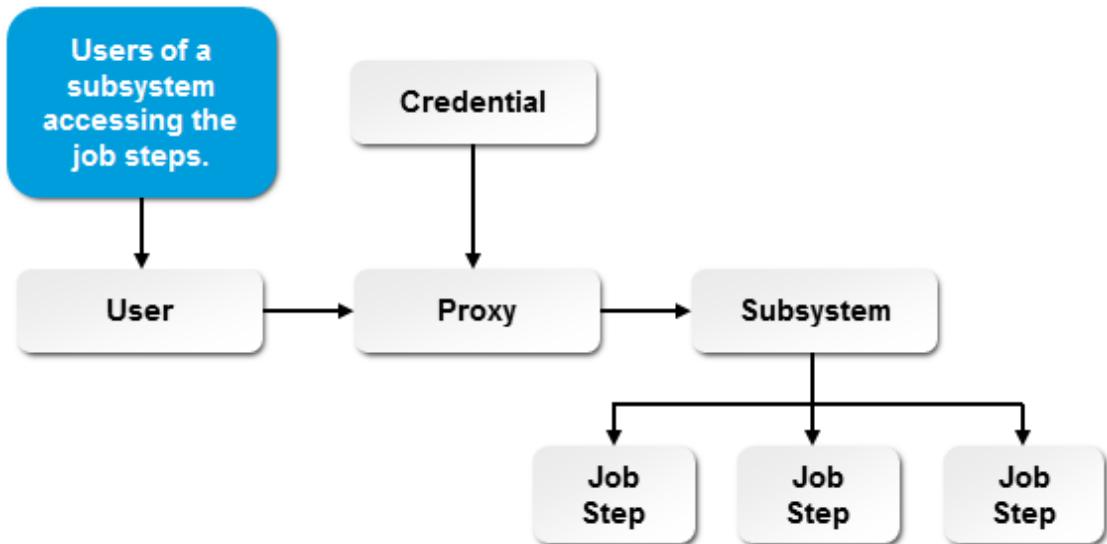


Figure 9–11: Users of a subsystem accessing the job steps using proxy and credential.

For instance, a subsystem called “Operating System” allows creating a job step that uses OS level processes such as moving, copying, or deleting a file. A user should be a part of the subsystem to create a job step in the subsystem.

Credentials



A *credential* is a record in SQL Server that stores authenticated information for an external resource. For instance, if you need to work with external entities, such as Windows Active Directory or a Linux based Lightweight Directory Access Protocol (LDAP) system, you can store the authentication information within SQL Server as a credential.

You can use a credential for the following purposes:

- Creating a normal SQL Server login against the credential.
- Using it as a proxy for access.

You can create a new credential window by using the **Server→Security→Credentials→New Credential** window.

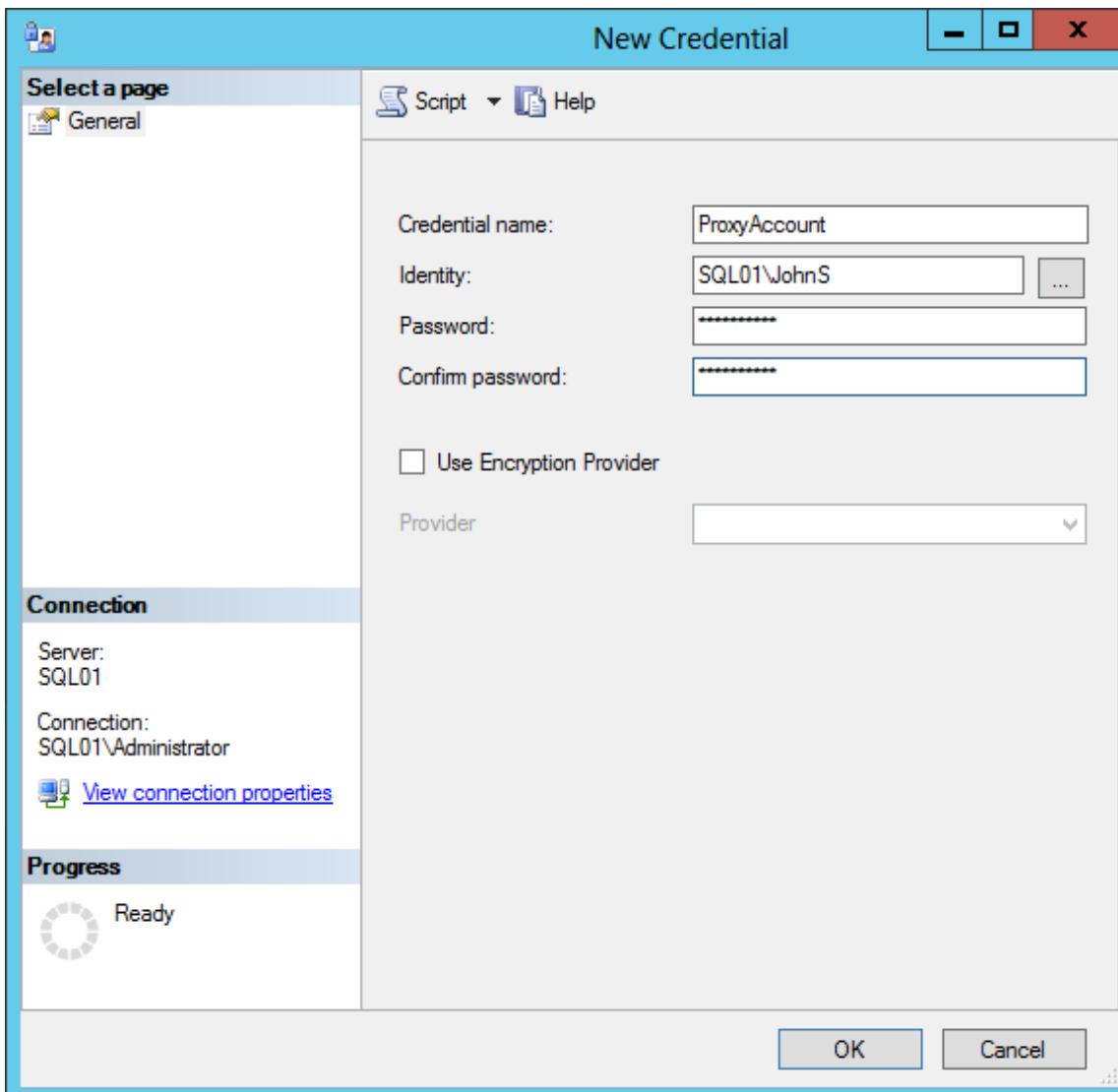


Figure 9–12: The New Credential window is used to create a new credential.



Note: You can use credentials in various ways.

Proxy Accounts

Proxy accounts are accounts that have permissions to perform job steps in a subsystem. To become part of the subsystem, you need to create a proxy account. If a proxy account called A is assigned to Subsystem S1 but not to Subsystem S2, it will not be able to run any process of S2. Each proxy account is mapped to a SQL Server Credential. SQL Server Agent uses the credential in the proxy to run a job step.



Proxy Accounts

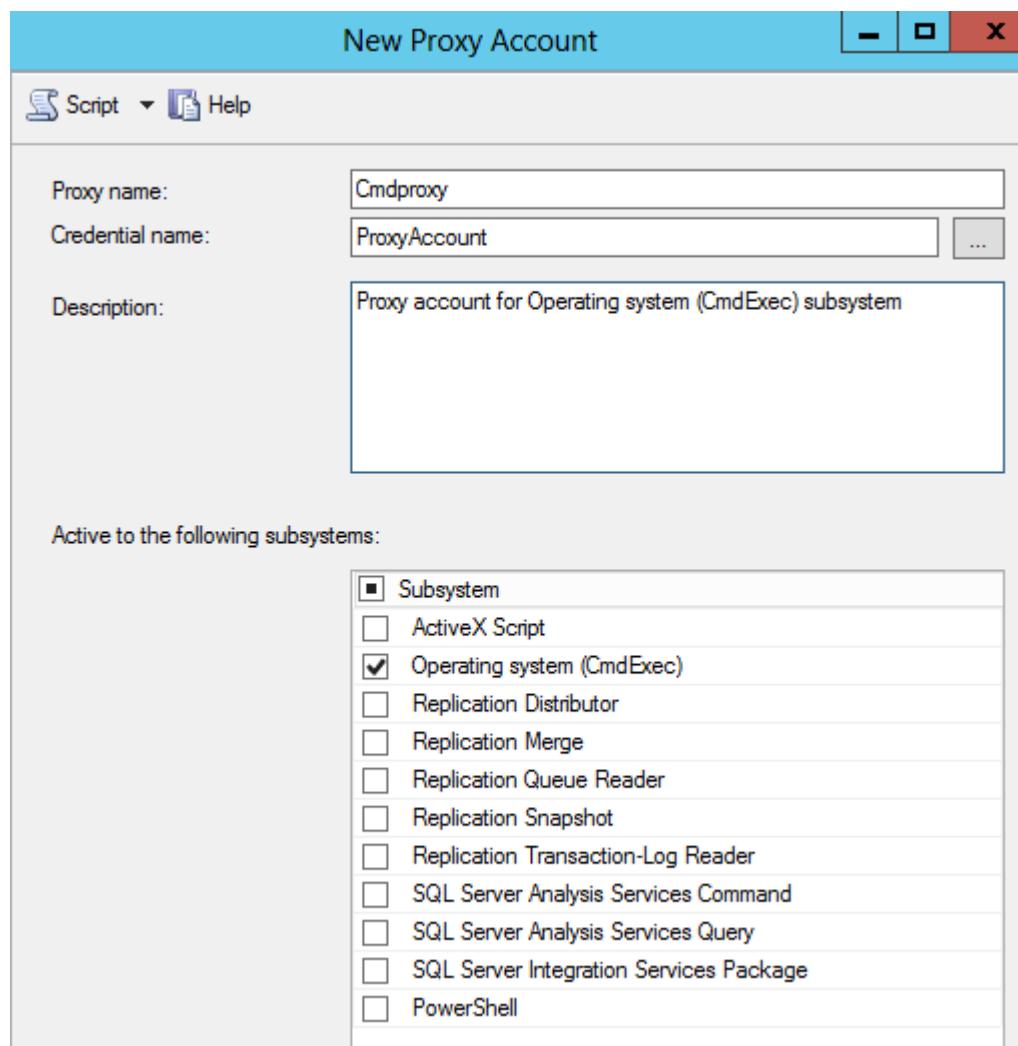
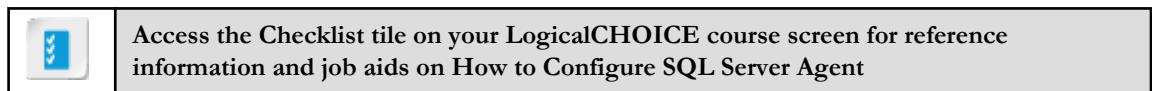


Figure 9–13: The New Proxy Account window is used to create a proxy account for a subsystem.



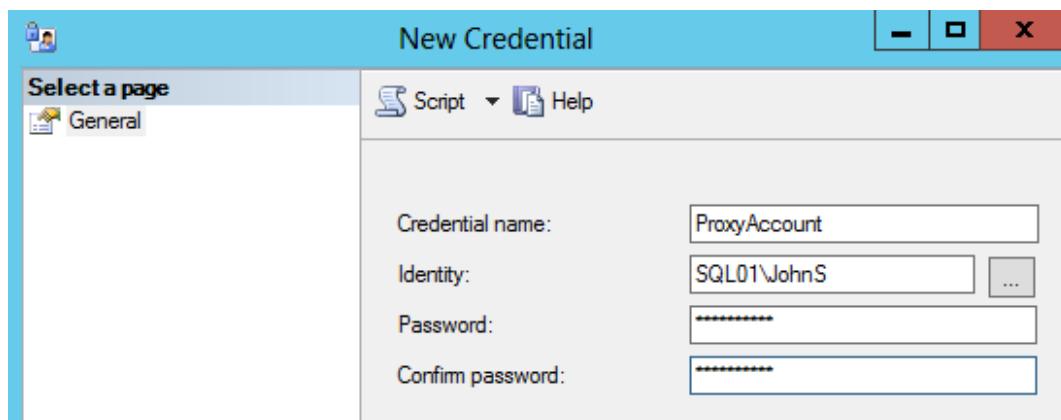
ACTIVITY 9–8

Implementing SQL Server Agent Security

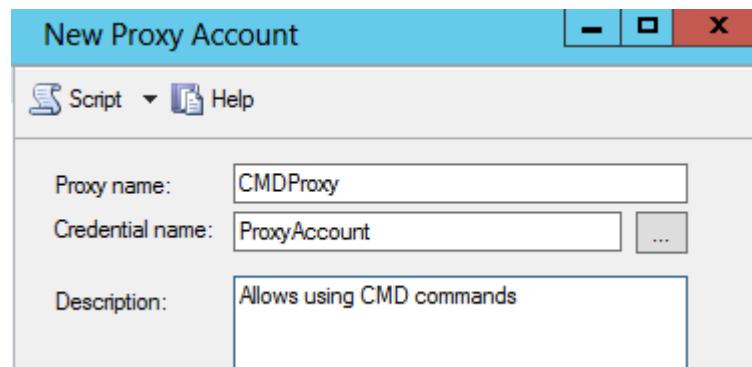
Scenario

The management of Adventure Works Cycles wants to implement security for jobs created in the Operating System sub-system. You are asked to allow a particular user to access the Operating System processes. You decide to create a credential and proxy account for the selected user.

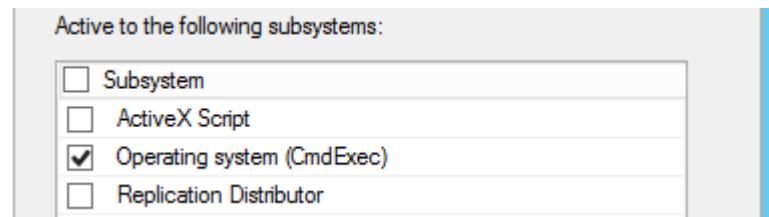
1. Create a new credential named *ProxyAccount*.
 - a) In the **Object Explorer** pane, expand **SQL01→Security**.
 - b) Right-click **Credentials** and select **New Credential**.
 - c) In the New Credential window, in the **Credential name** text box, type *ProxyAccount*.
 - d) In the **Identity** text box, select the ellipses button.
 - e) In the **Select User or Group** dialog box, in the **Enter the object name to select** text area, type *JohnS* and select the **Check Names** button.
 - f) Observe that the name **SQL01\JohnS** is displayed and select **OK** to add the identity.
 - g) In the **Password** and **Confirm password** text boxes, type *pass@word1*.



- h) In the New Credential window, select **OK** to save the credential.
2. Start the SQL Server Agent and create a proxy for the *ProxyAccount* credential.
 - a) In the **Object Explorer** pane, right-click **SQL Server Agent** and select **Start**.
 - b) In the **Microsoft SQL Server Management** message box, select **Yes** to confirm starting the SQL Server Agent.
 - c) In the **Object Explorer** pane, expand **SQL Server Agent →Proxies**.
 - d) Right-click **Operating System (CmdExec)** and select **New Proxy**.
 - e) In the New Proxy Account window, in the **Proxy name** text box, type **CMDProxy**.
 - f) In the **Credential name** text box, select the ellipses button.
 - g) In the **Select Credential** dialog box, in the **Enter the object names to select** section, type *ProxyAccount* and select **Check Names**.
 - h) Observe that the name **ProxyAccount** is displayed and select **OK** to add the credential.
 - i) In the New Proxy Account window, in the **Description** text box, type *Allows using CMD commands*.



- j) In the **Active to the following subsystems** section, ensure that the **Operating system (CmdExec)** check box is checked and select OK to add the Proxy account to the operating system subsystem.



ACTIVITY 9–9

Creating a New Agent Job

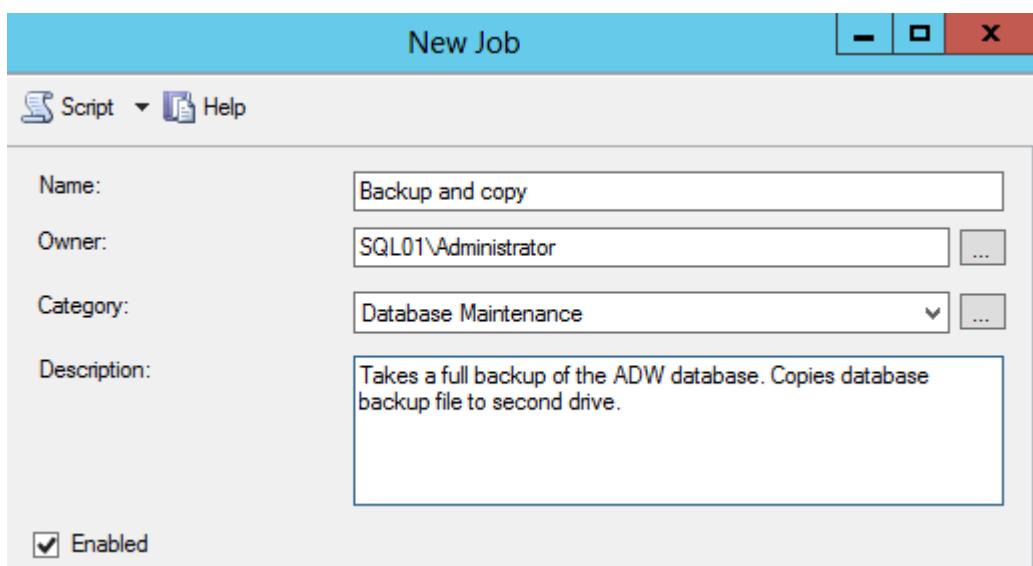
Before You Begin

Create a new folder named Backup in C:\Temp to store the backup of data.

Scenario

Your manager feels that a backup may be lost if the drive storing the backup fails. You are asked to take a copy of a backup file and store it in another drive as soon as the backup is taken. You decide to create a new agent job to perform this task.

1. Create a new job.
 - a) In the **Object Explorer** pane, right-click **Jobs** and select **New Job**.
 - a) In the New Job window, in the **Name** text box, type **Backup and copy** and in the **Owner** text box, ensure that **SQL01\Administrator** is selected.
 - a) From the **Category** drop-down list, select **Database Maintenance**.
 - a) In the **Description** text area, type **Takes a full backup of the ADW database. Copies database backup file to second drive** and ensure that the **Enabled** check box is checked.



2. Add a step to the new job to take a backup of the AdventureWorks2012 database.
 - a) In the New Job window, in the **Select a page** section, select **Steps**.
 - b) On the **Steps** page, select **New** to add a new job step.
 - c) In the New Job Step window, add the following options:
 - **Step Name:** Take Backup
 - **Type:** Transact-SQL script (T-SQL)
 - **Database:** AdventureWorks2012
 - d) In the **Command** text area, enter a BACKUP DATABASE statement to back up the AdventureWorks2012 database.

```
BACKUP DATABASE AdventureWorks2012
```

- e) Enter the TO DISK clause followed by the location of the backup file C:\Temp\Backup\ADW2012.bak.

```
BACKUP DATABASE AdventureWorks2012
TO DISK = 'C:\Temp\Backup\ADW2012.bak'
```

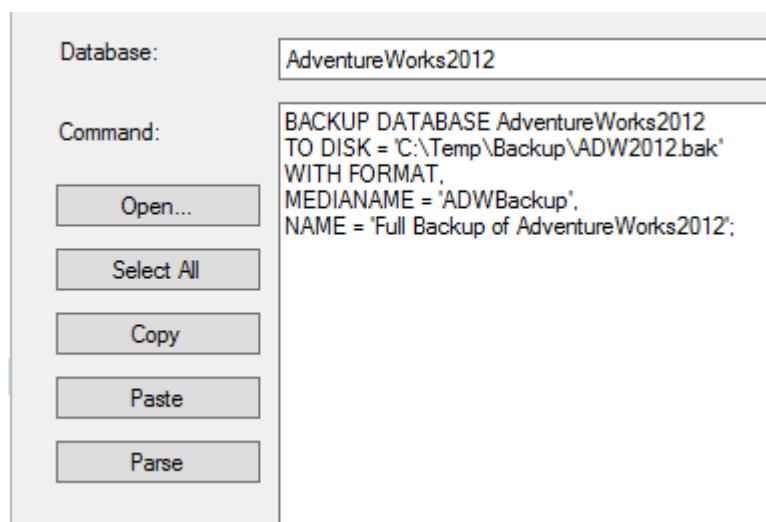
- f) Enter the WITH FORMAT clause and enter the MEDIANAME keyword followed by name of media as ADWBackup.

```
"...
TO DISK = 'C:\Temp\Backup\ADW2012.bak'
WITH FORMAT,
MEDIANAME = 'ADWBackup',
```

- g) Enter the NAME keyword followed by the name of backup Full Backup of AdventureWorks2012.

```
"...
MEDIANAME = 'ADWBackup',
NAME = 'Full Backup of AdventureWorks2012';
```

- h) In the New Job Step window, select **Parse** to ensure that there are no syntax errors in the statement and select **OK** to add the job step.



- i) In the **Parse Command Text** message box that displays the parsing details, select **OK**.
j) In the New Job Step window, select **OK** to add a new job step.

3. Add a step to the job to copy the backup file to another drive.

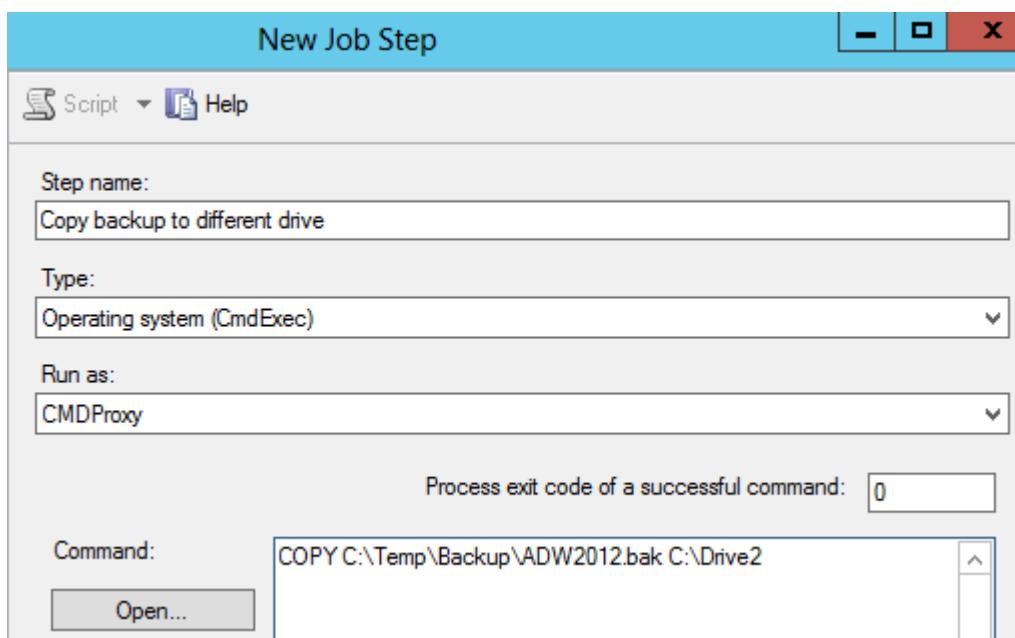
- a) On the **Steps** page, select **New** again to add another step.

- b) In the New Job Step window, add the following options:

- **Step Name:** Copy backup to different drive
- **Type:** Operating system (CmdExec)
- **Run As:** CMDProxy

- c) In the **Command** text area, type a COPY statement with the location C:\Temp\Backup\ADW2012 to copy the backup file to C:\Drive2.

```
COPY C:\Temp\Backup\ADW2012.bak C:\Drive2
```



- d) In the New Job Step window, select **OK** to add the new job step.
- e) In the New Job window, ensure that the two job steps are added and select **OK** to create a new job.

Job step list:				
St...	Name	Type	On Success	On Failure
1	Take Backup	Transact-...	Go to the ...	Quit the jo...
2	Copy backup to different drive	Operating...	Go to the ...	Quit the jo...

- f) If necessary, in the **Microsoft SQL Server Management Studio** message box, select **Yes** to change the **On Success** action of the last step.

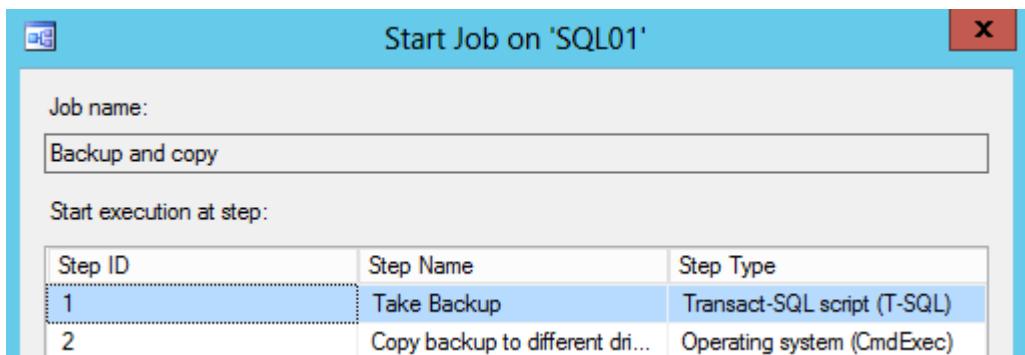
ACTIVITY 9–10

Monitoring Jobs

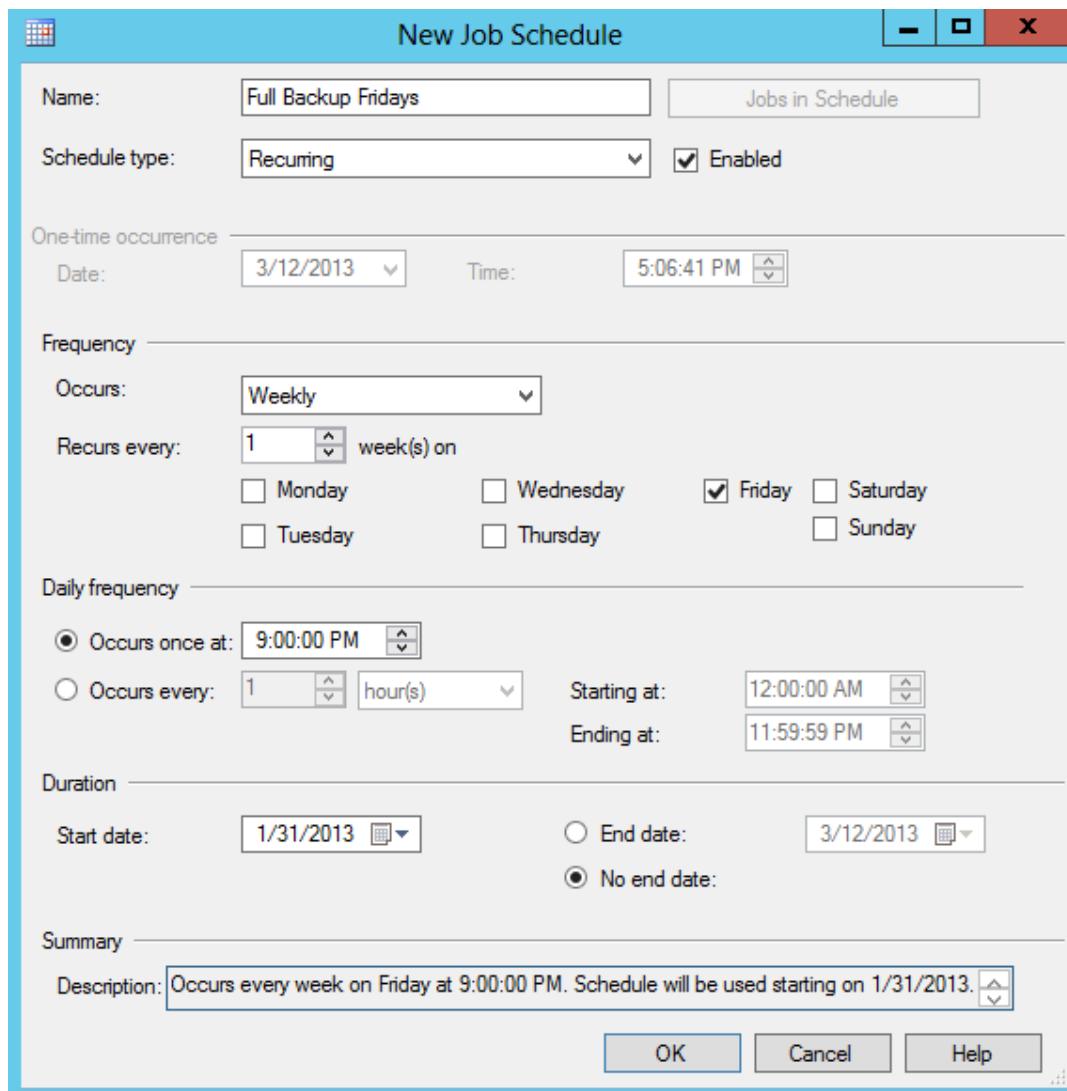
Scenario

The management of the Adventure Works Cycles wants to perform a backup of the AdventureWorks2012 database every Friday at 9 P.M. You decide to create a schedule to perform a backup.

1. Ensure that the job runs successfully.
 - a) In the **Object Explorer** pane, right-click **SQL01**→**Refresh**.
 - b) In the **Object Explorer** pane, expand **SQL Server Agent**→**Jobs**.
 - c) Right-click **Backup and copy** and select **Start Job at Step**.
 - d) In the Start Job window, ensure that step with **Step ID 1** is selected and select **Start**.



- e) Wait for the job to complete and select **Close**.
2. Create a schedule for the backup and copy job.
 - a) In the **Object Explorer** pane, right-click **Backup and copy** and select **Properties**.
 - b) In the **Select a page** section, select **Schedules**.
 - c) On the **Schedules** page, select the **New** button to create a new job schedule.
 - d) In the New Job Schedule window, in the **Name** text box, type **Full Backup Fridays**.
 - e) In the **Schedule type** drop-down list, ensure that **Recurring** is selected.
 - f) In the **Frequency** section, from the **Occurs** drop-down list, ensure that **Weekly** is selected.
 - g) In the **Recurs every** spin box, ensure that **1** is selected.
 - h) Uncheck the **Sunday** check box and check the **Friday** check box.
 - i) In the **Daily frequency** section, in the **Occurs once at** spin box, select **9:00:00 PM**.
 - j) In the **Duration** section, in the **Start date** spin box, verify that the current date is already displayed.
 - k) In the **Description** text box, ensure that a description for the schedule is displayed and select **OK** to save the schedule.



- I) In the Job Properties window, select **OK** to save the job with the schedule.

ACTIVITY 9-11

Administering Jobs and Alerts

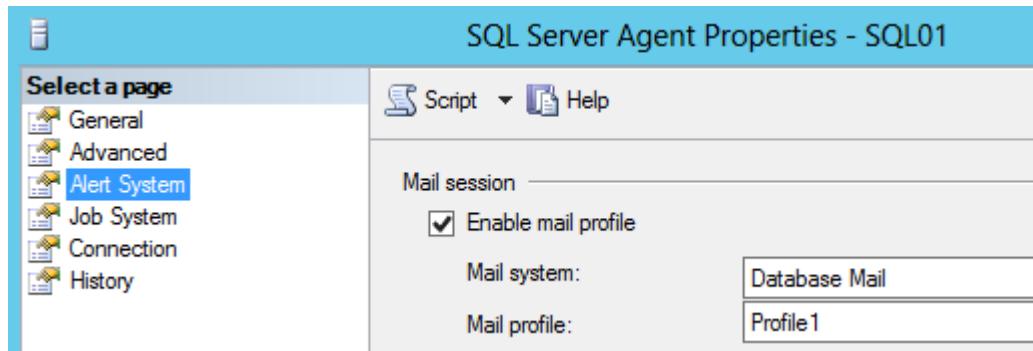
Scenario

Your manager wants a notification update of a scheduled job when the job is completed. You are asked to send the job status to the manager through email. You decide to create an alert to perform this task.

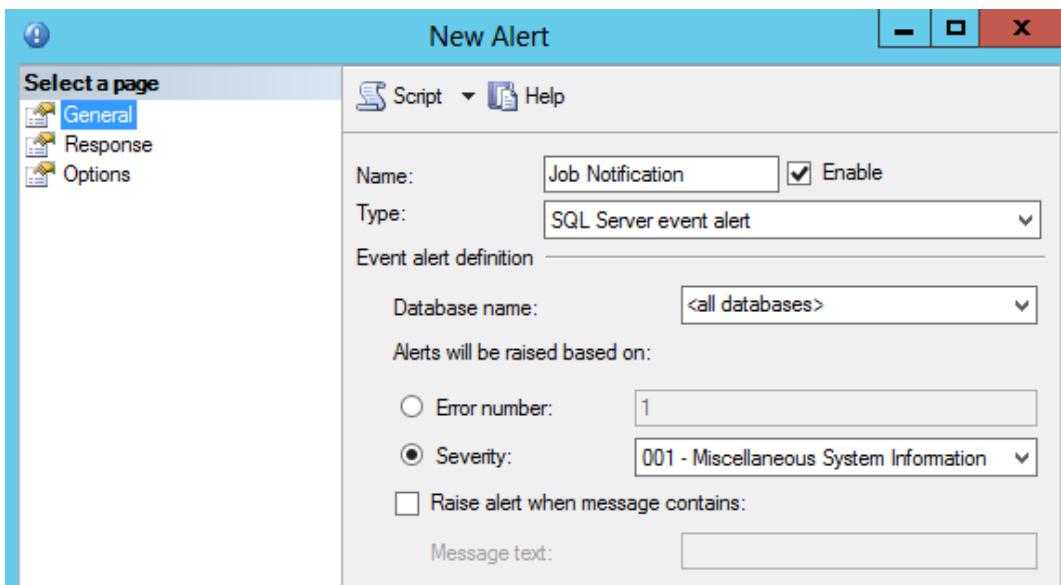
1. Configure SQL Server Agent alert system.
 - a) In the **Object Explorer** pane, right-click **Server Agent** and select **Properties**.
 - b) In the SQL Server Agent Properties window, in the **Select a page** section, select the **Alert System** page.
 - c) In the **Mail session** section, check the **Enable mail profile** check box.
 - d) From the **Mail system** drop-down list, ensure that **Database Mail** is selected and from the **Mail profile** drop-down list, select **Profile1**.



Remind participants that they have already configured Database Mail on the server.

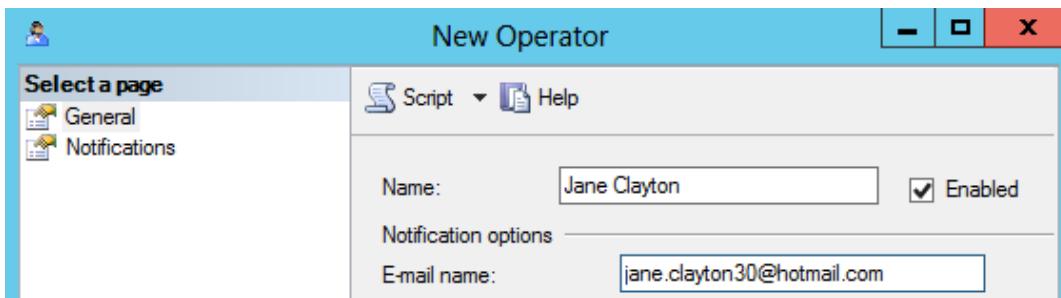


- e) In the SQL Server Agent Properties window, select **OK** to configure the alert system.
2. Create an alert for the Backup and copy job.
 - a) In the **Object Explorer** pane, right-click **Backup and copy** and select **Properties**.
 - b) In the Job Properties window, in the **Select a page** section, select the **Alerts** page.
 - c) In the **Alerts** page, select **Add** to add an alert.
 - d) In the New Alert window, in the **Name** text box, type **Job Notification**.
 - e) From the **Type** drop-down list, ensure that **SQL Server event alert** is selected.
 - f) In the **Event alert definition** section, from the **Database name** drop-down list, ensure that **<all databases>** is selected.
 - g) In the **Alerts will be raised based on** section, ensure that the **Severity** option is selected and from the drop-down list, ensure that **001 - Miscellaneous System Information** is selected.

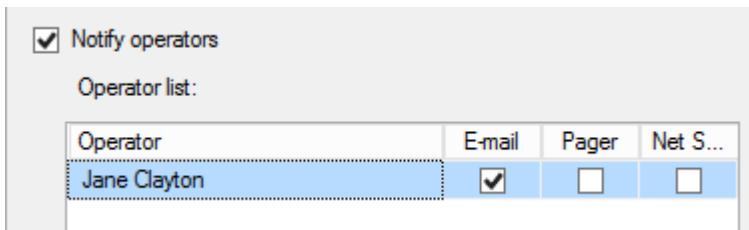


3. Specify the operator for the alert.

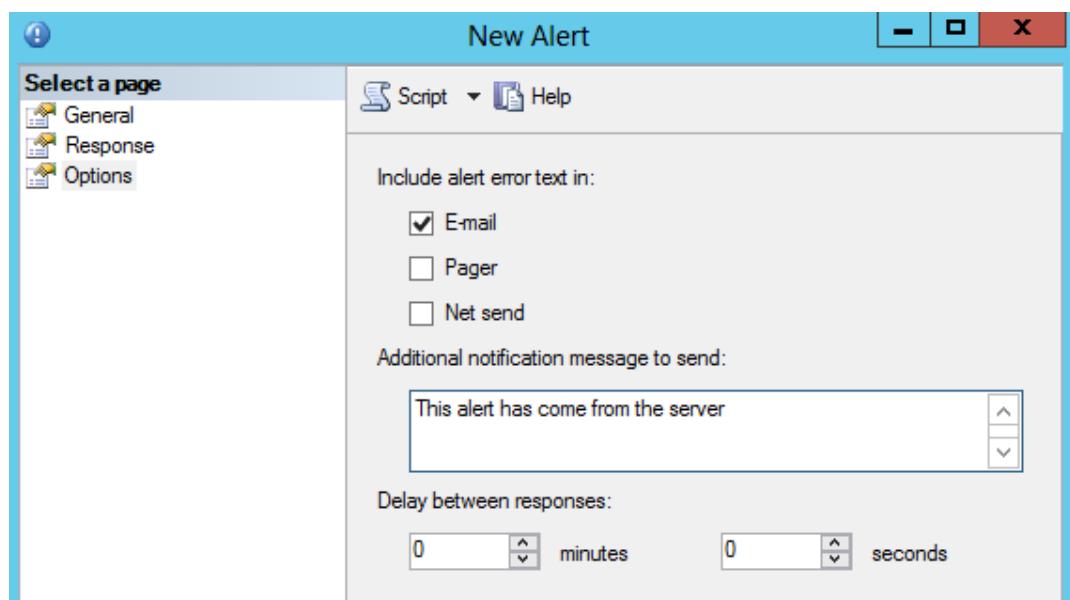
- In the **Select a page** section, select **Response**.
- In the **Response** page, check the **Notify operators** check box and select **New Operator**.
- In the New Operator window, in the **Name** text box, type **Jane Clayton**.
- In the New Operator window, in the **Email name** text box, type your email id and select **OK** to create a new operator.



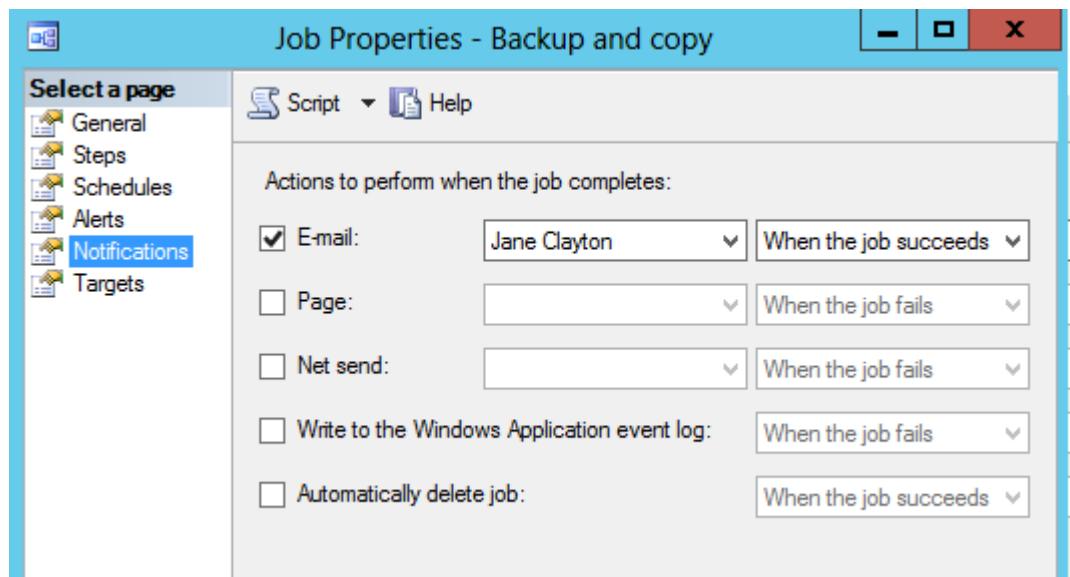
- In the New Alert window, in the **Operator list** section, observe that the operator **Jane Clayton** is listed and check the **Email** check box for **Jane Clayton**.



- In the **Select a page** section, select the **Options** page.
- On the **Options** page, in the **Include alert error text in** section, check the **E-mail** check box.
- In the **Additional notification message to send** text area, type **This alert has come from the server** and select **OK** to save the alert.



- i) In the Job Properties window, in the **Select a page** section, select the **Notifications** page.
- j) On the **Notifications** page, in the **Actions to perform when the job completes** section, check the **E-mail** check box, ensure that **Jane Clayton** is selected and from the drop-down list, select **When the job succeeds**.



- k) In the Job Properties window, select **OK** to save the job.

 Verify that all the participants were able to receive an alert in their email id.

 Remind participants that they will also receive notifications on all Fridays at 9:00 P.M. Invoked by the **Full Backup Friday** schedule.

4. Execute the job and verify the email alert.
 - a) In the **Object Explorer** pane, right-click **Backup and copy** and select **Start Job at Step**.
 - b) Select **Start** to execute the job and send an email alert once the job is completed.
 - c) In the Start Jobs window, select **Close** once the job is complete.
 - d) Login to the mail id of the operator.
 - e) Verify that the alert mail for the job is received.

Summary

You performed backup, restore, data transfer techniques, and configured SQL Server Agent on your server. The different backup, restore strategies, and transfer techniques will enable you to effectively plan and schedule your server maintenance and help recover your server from unexpected failures with minimal or no loss of data.

What are the factors that need be considered for implementing a backup and restore strategy?

A: Answers will vary, but may include: nature of the databases, how frequently the data in the databases are modified, and hardware constraints such as disk space.



Use the review questions provided to generate discussion among the participants.

What are the benefits of SQL Server Agent?

A: Answers will vary, but may include: to automate tasks using jobs, scheduling jobs, and sending alerts.



Note: Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

10

Implementing High Availability

Lesson Time: 6 hours

Lesson Objectives

In this lesson, you will:

- Work with clustering.
- Work with replication.
- Work with database mirroring and AlwaysOn.
- Work with Central Management Servers.

Lesson Introduction

You have performed backup and restore operations on databases. At an enterprise level, it is critical to ensure that data is available to clients even when a maintenance operation is in progress. By using the built-in features of SQL Server, you can build an environment that ensures high data availability. In this lesson, you will implement high availability.

TOPIC A

Work with Clustering

SQL Server provides several techniques that ensure high availability of data and provide disaster recovery capability. Clustering is one such feature. Clustering enables a network to service clients even if there is a failure of certain physical components in the network environment. In this topic, you will work with clustering.

WSFC

Windows Server Failover Clustering (WSFC) is a Windows Server technology that enables multiple servers to work together. This technology provides features that support high availability and disaster recovery capabilities. By deploying SQL Server on Windows Server failover clusters, you can take advantage of the features that WSFC supports. In a failover cluster, multiple SQL Servers work together as if they were a single server. If one of the servers fails, another server takes over to ensure that services to the end user are not affected. Failover clustering ensures that there is no point of failure.

Clusters

A *cluster* is a group of independent servers that provide a set of services as a single unit. Each server in a cluster is called a *node*. The nodes in a cluster access a shared storage that contains databases and log files. They work together to ensure availability of SQL Server services and applications at all times. SQL Server supports a number of different configuration schemes for clustering.



Note: For more information about clusters, check out the **LearnTO Ensure Availability Using SQL Server Clusters** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.



You may want to show **LearnTO Ensure Availability Using SQL Server Clusters** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOS for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.



Resource and Availability Groups

Resource and Availability Groups

Groups are logical sets of components that make up a cluster and act as a single unit within the cluster. There are two types of groups—resource groups and availability groups.

Group	Description
<i>Resource group</i>	Contains resources that can be shared. For example, a resource group can contain a Network Attached Storage (NAS) device and a Storage Area Network (SAN) share. The resources in the group work together in tandem to ensure that the storage system never fails.
<i>Availability group</i>	Contains SQL Server nodes that work together. For example, you can create one availability group of all the SQL Server nodes at the data center and another availability group of all the SQL Server nodes at your branch office. If there is an outage at the data center, the entire cluster switches roles to use the availability group at the branch office.

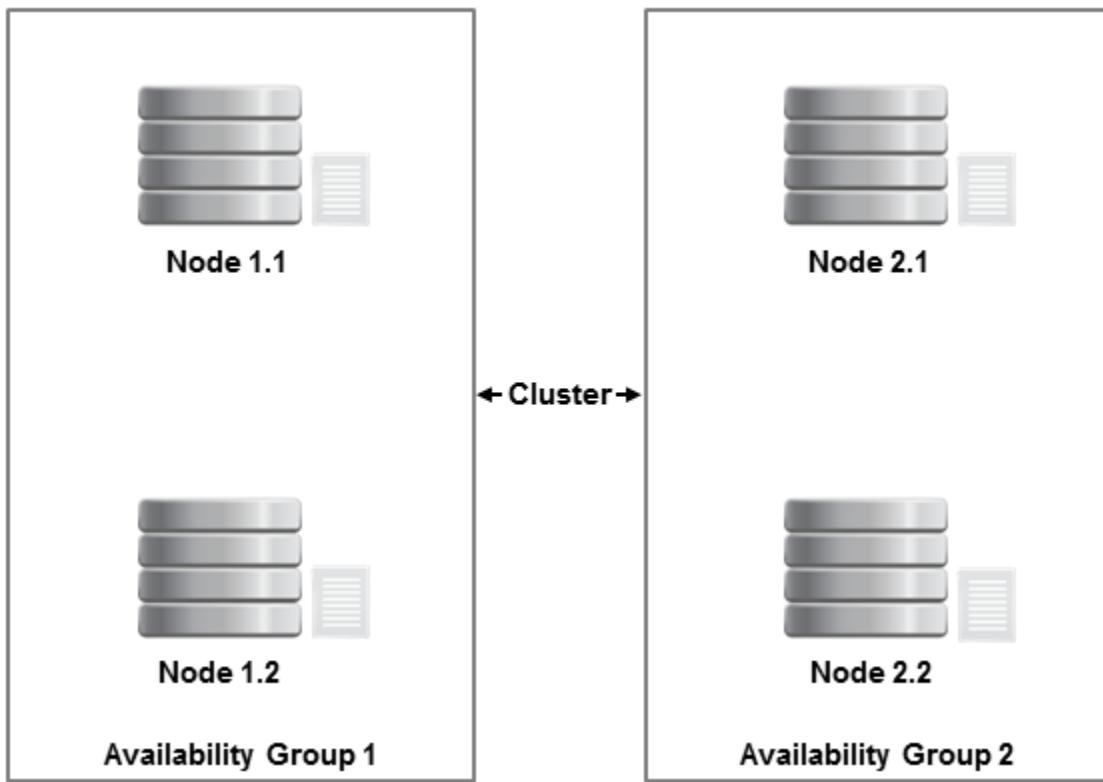


Figure 10–1: A cluster with two availability groups each containing two nodes.

Failover

Failover is the process in which when a node in a cluster fails, the services provided by that node are switched to another available node or availability group. The current availability group that services requests is called the *primary availability group*. The other availability groups in the cluster are called *secondary availability groups*.

When a primary availability group fails, the cluster switches to a secondary availability group. Once the switch occurs, the secondary availability group is active and it becomes the primary availability group. When the failed primary availability group comes back online and rejoins the cluster, it becomes a secondary availability group until the time another switch occurs.



Failover

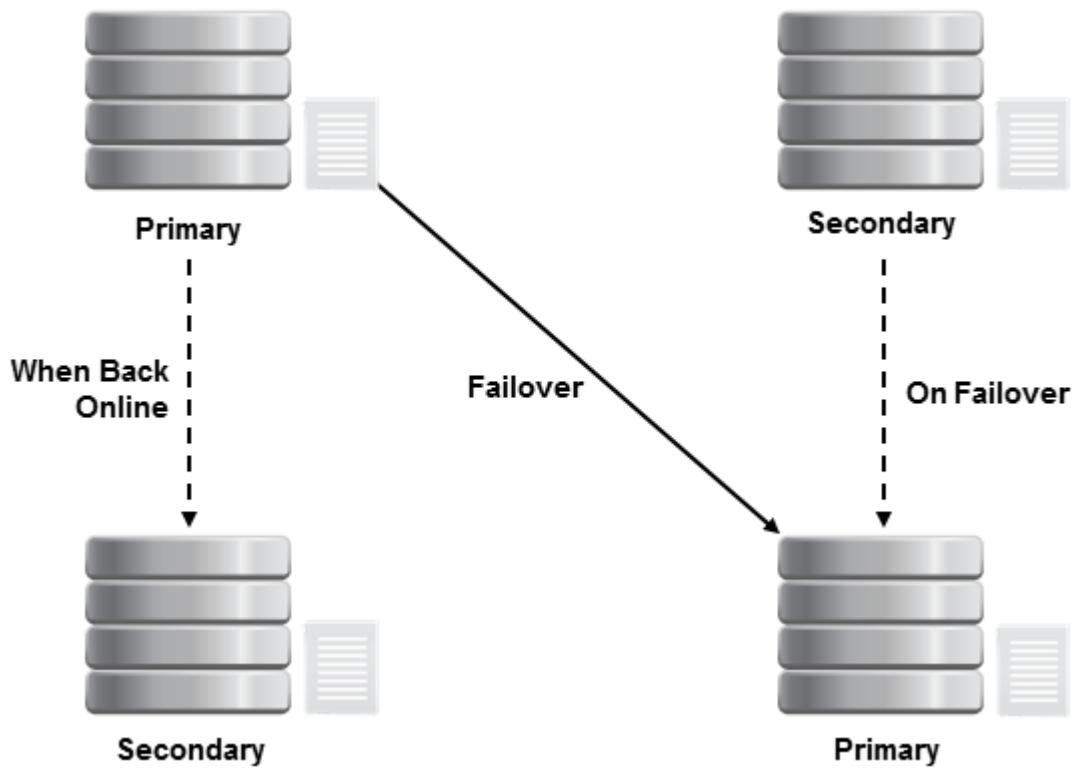


Figure 10-2: Switching between secondary and primary availability groups during the failover process.



Ask the students to identify the critical servers in their enterprise networks that require high availability and disaster recovery capabilities. Ask them to identify how the networks would benefit by implementing a failover cluster.

Types of Failover

There are three ways in which switching occurs in a cluster—Automatic Failover, Planned Manual Failover, and Unplanned Forced Manual Failover.

Type of Failover	Occurs When
Automatic Failover	The cluster detects that the primary availability group is unavailable and switches to the secondary availability group. No data loss is incurred in this mode.
Planned Manual Failover	Database administrators decide to perform a planned failover so that the secondary availability group becomes the active availability group. A planned manual failover is carried out when a maintenance operation has to be performed on the nodes in the primary availability group. No data loss is incurred in this mode.
Unplanned Forced Manual Failover	There is a catastrophic failure on the network or nodes and database administrators need to perform an emergency failover operation. This scenario has the potential for data loss if the failure remains undetected for a while, or the failover takes a long time to complete, or the primary availability group failed to commit the day before it crashed.

Failover Mode

Failover mode is a property of an availability group that you need to set while creating the availability group. This property determines whether failover occurs automatically or manually. The value you set for this property depends on the availability mode of the availability group. You can set two values—automatic and manual.

Value	Description
Automatic	The cluster switches from the primary availability group to a secondary availability group automatically.
Manual	Switching does not occur unless the database administrator manually switches the primary and secondary availability groups.

Availability Modes

Availability mode is a property of an availability group that determines whether the primary availability group waits for a secondary availability group to commit its transactions on the disk. There are two availability modes—synchronous commit and asynchronous commit.

Availability Mode	Description
Synchronous commit	Data in the primary and the secondary availability groups is in sync at all times. In this mode, the primary availability group commits its transactions only after a secondary availability group has committed its transactions. Because the primary and secondary availability groups are always synchronized, this mode supports all types of failover—automatic, planned manual, or unplanned forced manual failover. When a failover occurs, the cluster switches over without encountering any issues or data loss.
Asynchronous commit	Data in the primary and the secondary availability groups is not synchronized all the time. In this mode, the primary availability group commits its transactions irrespective of whether or not a secondary availability group has committed its transactions to the disk. Data loss is possible if the secondary availability group failed to commit its transactions ahead of the primary availability group. This mode supports only unplanned forced failover.

Clustered Instances

A *clustered instance* is an instance of SQL Server that is installed on a node in a cluster. As more nodes are added to the cluster, one of the nodes assumes leadership of the cluster. This node brings up its SQL Server instance online and takes charge of all the shared resources or groups. Some of these shared resources are:

- Cluster network name
- Cluster IP address
- Database engine
- SQL Server agent
- All file and disk resources, such as shares, disks, and FILESTREAMs



Ask the students to identify the differences between synchronous and asynchronous commit. Discuss scenarios that would require the use of each availability mode.

Failover in Clustered Instances

In a cluster, once a node assumes leadership and takes charge of all the resources, the SQL Server instance installed on that node becomes the only active instance running at any point of time. If either an automatic or manual failover occurs, this instance writes all uncommitted data to the disk. It then releases all the resources under its control and the cluster switches to another node. The new leader repeats the process by taking over the shared resources and starting up its SQL Server instance.

Benefits of Clustered Instances

Clustered instances provide high availability because of the presence of redundant nodes. In the case of a failure, the cluster switches from one node to the other. However, this failure is not visible to clients because they may not be aware of any change in the server status. Clients always use the network name and IP address of the cluster and they are automatically redirected to the new node handling their requests.

MSFC

Normally, all the nodes in a cluster are in the same IP address subnet. However, in SQL Server 2012, you can span the nodes in a cluster across multiple IP subnets using the Multi Subnet Failover Clustering (MSFC) feature. MSFC enables you to set up a cluster in which each node in the cluster is connected to a different IP subnet or some nodes are in a single subnet and the other nodes are in other subnets. The subnets can be present in the same physical location or in geographically dispersed locations.

Based on the IP addresses bound to the nodes in a cluster, SQL Server automatically detects multi subnet clustering. When multi subnet clustering is detected, SQL Server attempts to perform a failover within nodes in the same subnet as that of the cluster. If nodes are not found in the same subnet, then it attempts to connect to nodes in the other subnets.



Note: When nodes are in geographically dispersed locations, it is not possible to share disks and resources. In such cases, it is required to maintain a copy of data in each node in a cluster so that there is no data loss whenever a failover occurs.



Ask the students to list the benefits of using MSFC. Discuss the network scenarios that would benefit from an MSFC implementation.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Work with Clustering

ACTIVITY 10-1

Configuring Windows Server Clusters

Before You Begin

Start and connect to the Server3 and Server4 VMs.

1. In the Server Manager window, select **Tools**→**Hyper-V Manager**.
2. In the Hyper-V Manager window, in the right pane, right-click **Server2** and select **Start**.
3. Right-click **Server3** and select **Start**.
4. Right-click **Server3** and select **Connect**.
5. In the login screen in Server3, press **Ctrl+Alt+End**.
6. Verify that the username is **ADWCYCLES\Administrator**.



Note: If the domain is not displayed, select the right arrow icon and switch the user.

7. In the Password box, type **server@2012** and press **Enter**.
8. Switch to the physical system.
9. In the Hyper-V Manager window, in the right pane, right-click **Server4** and select **Start**.
10. Right-click **Server4** and select **Connect**.
11. In the login screen in Server4, press **Ctrl+Alt+End**.
12. Verify that the username is **ADWCYCLES\Administrator**.



Note: If the domain is not displayed, select the right arrow icon and switch the user.

13. In the Password box, type **server@2012** and press **Enter**.

Scenario

The management of Adventure Works Cycles has asked you to implement high availability to ensure that the databases in the organization's network are always available. Before doing this, you need to enable failover clustering on the servers in your network and connect together in one cluster.

1. Add the Failover Clustering feature in Server3 VM.
 - a) Switch to the Server3 VM.
 - b) In the Server Manager window, in the left pane, select **Dashboard**.
 - c) In the **WELCOME TO SERVER MANAGER** pane, select **Add roles and features**.
 - d) In **Add Roles and Features Wizard**, on the **Before you begin** page, select **Next**.
 - e) On the **Select installation type** page, verify that **Role-based or feature-based installation** is selected and select **Next**.

Role-based or feature-based installation

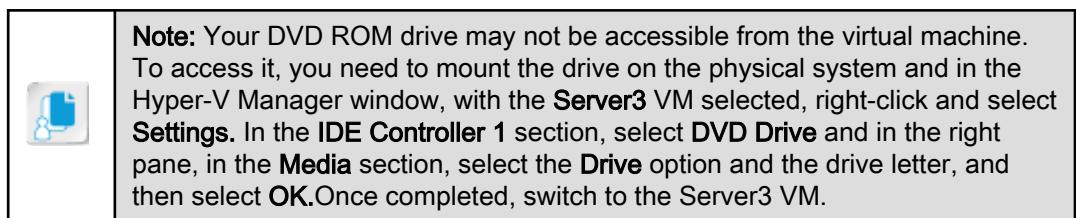
Configure a single server by adding roles, role services, and features.

Remote Desktop Services installation

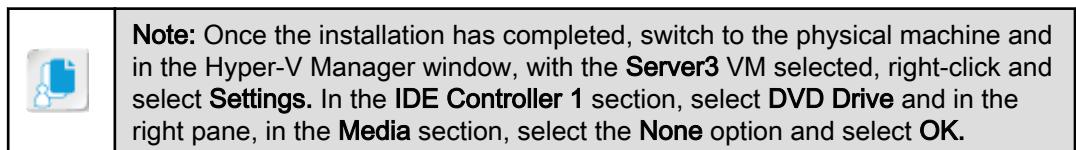
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.

- f) On the **Select destination server** page, select **Server3.adwcycles.local** and select **Next**.

- g) On the **Select server roles** page, accept the default settings and select **Next**.
- h) On the **Select features** page, in the **Features** list box, check the .NET Framework 3.5 (Features) check box.
- i) Check the **Failover Clustering** check box and select **Next**.
- j) In the **Add Role and Feature wizard** dialog box, select **Add Features** to add the features required for Failover Clustering. Select **Next**.
- k) On the **Confirm installation selections** page, select the **Specify an alternate source path** link.
- l) In the **Specify Alternate Source Path** dialog box, in the **Path** text box, specify the drive letter of your DVD ROM drive followed by :\\Sources\\SxS and select **OK**.



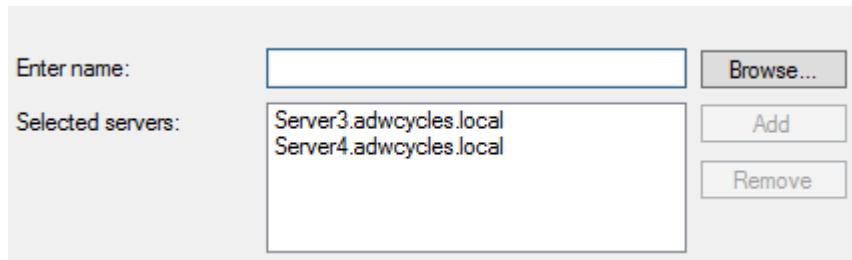
- m) On the **Confirmation installation** page, select **Install**.
- n) On the **Installation progress** page, view the progress of the installation and select **Close**.



- o) Similarly, perform steps "a" through "n" to add the Failover Clustering feature in Server4 VM.

2. Validate the configuration of the nodes.

- a) Switch to the Server3 VM.
- b) In the Server Manager window, select **Tools→Failover Cluster Manager**.
- c) In the **Actions** pane, in the **Failover Cluster Manager** section, select the **Validate Configuration** link.
- d) In the **Validate a Configuration** wizard, on the **Before You Begin** page, select **Next**.
- e) On the **Select Servers or a Cluster** page, in the **Enter name** text box, type **Server3** and select **Add**.
- f) Verify that in the **Selected servers** box, **Server3.adwcycles.local** is displayed.
- g) In the **Enter name** text box, type **Server4** and select **Add**.



- h) Verify that in the **Selected servers** box, **Server4.adwcycles.local** is displayed and select **Next**.
- i) On the **Testing Options** page, verify that the **Run all tests (recommended)** option is selected and select **Next**.
- j) On the **Confirmation** page, review the settings and select **Next** to start running the cluster validation tests. On the **Validating** page, you can view the progress of the validation.
- k) On the **Summary** page, view the summary of the cluster validation test result and select **Finish**.

Failover Cluster Validation Report

Node:	Server3.adwcycles.local	Validated
Node:	Server4.adwcycles.local	Validated

Create the cluster now using the validated nodes...

3. Create clusters and configure them.
 - a) In the **Create Cluster** wizard, on the **Before You Begin** page, select **Next**.
 - b) On the **Access Point for Administering the Cluster** page, in the **Cluster name** text box, type **SQLCluster** and in the **Address** text box, type **192.168.1.245** and select **Next**.

Type the name you want to use when administering the cluster.

Cluster Name: **SQLCluster**

The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

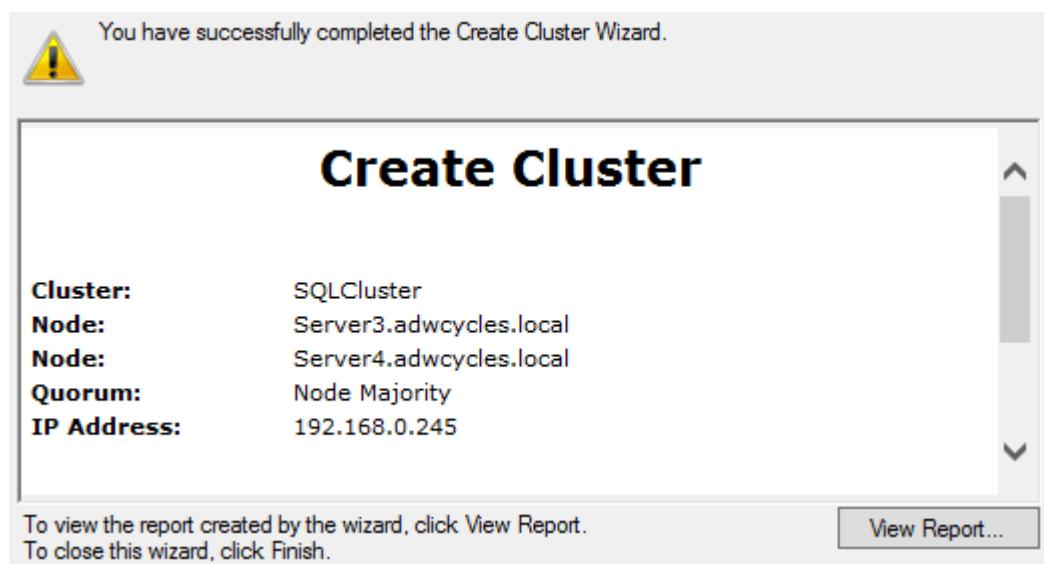
	Networks	Address
<input checked="" type="checkbox"/>	192.168.0.0/24	192 . 168 . 0 . 245

- c) On the **Confirmation** page, verify the settings to create a cluster, and select **Next**.

You are ready to create a cluster.
The wizard will create your cluster with the following settings:

Cluster:	SQLCluster
Node:	Server3.adwcycles.local
Node:	Server4.adwcycles.local
IP Address:	192.168.0.245

- d) On the **Creating New Cluster** page, observe the progress of cluster configuration.
- e) On the **Summary** page, select **Finish**.



- f) Close the Failover Cluster Manager window.
4. Create a folder for the quorum storage.
 - a) Switch to the physical system.
 - b) In the Hyper-V manager, right-click **Server2** and select **Connect**.
 - c) Open Windows Explorer and create a folder named **Shared** in C:\ drive.
 - d) In the **C:\Shared** folder, create a folder named **Quorum**.
 - e) Right-click the **Quorum** folder and select **Properties**.
 - f) In the **Quorum Properties** dialog box, select the **Sharing** tab.
 - g) Select **Advanced Sharing**.
 - h) Select the **Share this folder** check box.
 - i) Select the **Permissions** button. In the **Advanced Sharing** dialog box, in the **Group or user names** list, select **Everyone**.
 - j) In the **Permissions for Everyone** list, in the **Allow** column, check the **Full Control** check box and select **OK**.
 - k) In the **Advanced Sharing** dialog box, select **OK**.
 - l) In the **Quorum Properties** dialog box, select **Close**.
5. Create a folder for sharing data.
 - a) In Windows Explorer, in the **C:\Shared** folder, create a folder named **Data**.
 - b) Right-click the **Data** folder and select **Properties**.
 - c) In the **Data Properties** dialog box, select the **Sharing** tab.
 - d) Select **Advanced Sharing** and then select the **Share this folder** check box.
 - e) Select the **Permissions** button. In the **Advanced Sharing** dialog box, in the **Group or user names** list, select **Everyone**.
 - f) In the **Permissions for Everyone** list, in the **Allow** column, check the **Full Control** check box and select **OK**.
 - g) In the **Advanced Sharing** dialog box, select **OK**.
 - h) In the **Data Properties** dialog box, select **Close**.
6. Configure the cluster Quorum settings.
 - a) Switch to the Server3 VM.
 - b) In the Failover Cluster Manager window, in the left pane, select **SQLCluster.adwcycles.local**.
 - c) In the **Actions** pane, select **More Actions**, and then select **Configure Cluster Quorum Settings**.
 - d) In the **Configure Cluster Quorum** wizard, on the **Before You Begin** page, select **Next**.
 - e) On the **Select Quorum Configuration Option** page, select the **Add or change the quorum witness** option and select **Next**.

- f) On the **Select Quorum Witness** page, select the **Configure a file share witness** option and select **Next**.
 - g) On the **Configure File Share Witness** page, in the **File Share Path** text box, type **\server2\Quorum** and select **Next**.
 - h) On the **Confirmation** page, select **Next**.
 - i) On the **Summary** page, select **Finish**.
-

ACTIVITY 10-2

Setting Up an SQL Server Cluster

Before You Begin

You have logged in as **ADWCYCLES\Administrator** on Server3 VM.

Scenario

Adventure Works Cycles has various departments constantly accessing the databases for viewing and generating reports. It is essential to ensure that the databases are up and running all the time. You decide to implement high availability by installing SQL Server clusters.

1. Launch the Installation wizard.

- Open Windows Explorer and navigate to the **\SQL01\C\$\SQLServer2012 Installation** folder that contains the SQL Server installation components.
- Double-click the **Setup** file to launch the **SQL Server Installation Center**.
- In the SQL Server Installation Center window, in the left pane, select **Installation**.
- In the right pane, select the **New SQL Server failover cluster installation link**.



Note: In the **SQL Server 2012 Setup** dialog box, the **Setup Support Rules** operation identifies problems that might occur during the installation. This operation might take a few minutes to complete. Once the **Setup Support Rules** operation completes problem detection, a message indicating that the operation is complete appears.



[New SQL Server stand-alone installation or add features to an existing installation](#)

Launch a wizard to install SQL Server 2012 in a non-clustered environment or to add features to an existing SQL Server 2012 instance.



[New SQL Server failover cluster installation](#)

Launch a wizard to install a single-node SQL Server 2012 failover cluster.



[Add node to a SQL Server failover cluster](#)

Launch a wizard to add a node to an existing SQL Server 2012 failover cluster.



[Upgrade from SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2](#)

Launch a wizard to upgrade SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2 to SQL Server 2012.

- In the **Setup Support Rules** dialog box, select **OK**.
- In the **SQL Server 2012 Setup** dialog box, select **OK**.
- On the **Product Key** page, verify that the **Specify a free edition** option is selected and select **Next**.
- On the **License Terms** page, check the **I accept the license terms** check box and select **Next**.
- On the **Product Updates** page, uncheck the **Include SQL Server product updates** check box, and select **Next**.
- On the **Install Setup Files** page, select **Install**.



Note: Once you select **Install** on the **Install Setup File** page, the installation process starts. Wait for a few minutes until the installation of setup files is complete.

2. Specify the features to be installed in the cluster.
 - a) On the **Setup Support Rules** page, select **Next**.
 - b) On the **Setup Role** page, verify that the **SQL Server Feature Installation** option is selected and select **Next**.

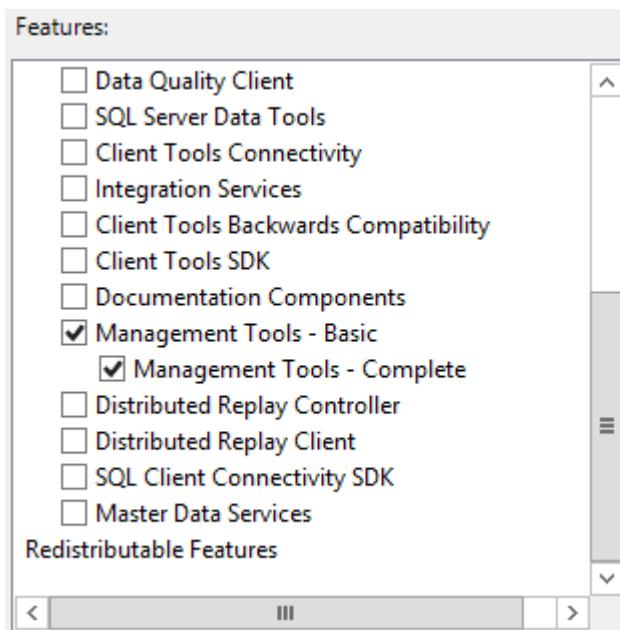


Note: Once you select **Next** on the **Setup Role** page, it takes a few minutes for the features to load on the **Feature Selection** page.

- c) On the **Feature Selection** page, in the list of features, check the **Database Engine Services** check box.
- d) Scroll down and in the **Shared Features** list, select the **Management Tools — Complete** and select **Next**.



Note: Once you select **Next** on the **Feature Selection** page, wait for the wizard to complete the task and return to the **Install a SQL Server Failover Cluster** dialog box.



- e) On the **Feature Rules** page, select **Next**.
3. Specify the cluster's name and IP address settings.
 - a) On the **Instance Configuration** page, in the **SQL Server Network Name** text box, type **SQL2012Cluster** and select **Next**.
 - b) On the **Disk Space Requirements** page, observe the summary that describes the disk usage and select **Next**.
 - c) On the **Cluster Resource Group** page, in the **SQL Server cluster resource group name** text box, type **SQLDefaultCluster** and select **Next**.

SQL Server cluster resource group name: **SQLDefaultCluster**

Qualified	Name	Message
■	Available Storage	The cluster group 'Available Storage' is reserved by Windows Failover Cluster.
■	Cluster Group	The cluster group 'Cluster Group' is reserved by Windows Failover Cluster.

- d) On the **Cluster Disk Selection** page, select **Next**.
- e) On the **Cluster Network Configuration** page, in the **IP Type** column, check the **IPv4** check box.
- f) In the **Address** column, type **192.168.1.240**

Specify the network settings for this failover cluster:

	IP Type	DHCP	Address	Subnet Mask	Subnet(s)	Network
<input checked="" type="checkbox"/>	IPv4	<input type="checkbox"/>	192.168.0.240	255.255.255.0	192.168.0.0/24	Cluster Network 1

- g) In the **DHCP** column, verify that the **DHCP** check box is not checked and select **Next**.
4. Configure the server and database engine.
- a) On the **Server Configuration** page, in the **SQL Server Agent** row, in the **Account Name** column, type **adwcycles\administrator**.
 - b) In the **Password** column, type **server@2012**.
 - c) In the **SQL Server Database Engine** row, in the **Account Name** column, type **adwcycles\administrator**.
 - d) In the **Password** column, type **server@2012** and select **Next**.

Microsoft recommends that you use a separate account for each SQL Server service.

Service	Account Name	Password	Startup Type
SQL Server Agent	adwcycles\administrator	*****	Manual
SQL Server Database Engine	adwcycles\administrator	*****	Manual
SQL Full-text Filter Daemon Launc...	NT Service\MSSQLFDFL...		Manual
SQL Server Browser	NT AUTHORITY\LOCAL ...		Automatic

- e) On the **Database Engine Configuration** page, in the **Authentication Mode** section, select the **Mixed Mode (SQL Server authentication and Windows authentication)** option.
- f) In the **Enter password** text box, type **pass@word1**.
- g) In the **Confirm password** text box, type **pass@word1**.
- h) Select **Add Current User**.

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

Windows authentication mode
 Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password:

Confirm password:

Specify SQL Server administrators

ADWCYCLES\Administrator (Administrator)	SQL Server administrators have unrestricted access to the Database Engine.
--	--

Add Current User **Add...** **Remove**

- i) Verify that **ADWCYCLES\Administrator** is added to the SQL Server administrators list and select the **Data Directories** tab.
- j) In the **Data root directory** **\server2\data**.
- k) Observe that the other text boxes are populated based on the path specified in the **Data root directory** text box and select **Next**.

Data root directory:	<input type="text" value="\\server2\data"/>	...
System database directory:	<input type="text" value="\\server2\data\MSSQL11.MSSQLSERVER\MSSQL\Data"/>	...
User database directory:	<input type="text" value="\\server2\data\MSSQL11.MSSQLSERVER\MSSQL\Data"/>	...
User database log directory:	<input type="text" value="\\server2\data\MSSQL11.MSSQLSERVER\MSSQL\Data"/>	...
Temp DB directory:	<input type="text" value="\\server2\data\MSSQL11.MSSQLSERVER\MSSQL\Data"/>	...
Temp DB log directory:	<input type="text" value="\\server2\data\MSSQL11.MSSQLSERVER\MSSQL\Data"/>	...
Backup directory:	<input type="text" value="\\server2\data\MSSQL11.MSSQLSERVER\MSSQL\Backup"/>	...

- l) In the **Install a SQL Server Failover Cluster** message box, select **Yes**.
- m) On the **Error Reporting** page, select **Next**.
- n) On the **Cluster Installation Rules** page, select **Next**.
- o) On the **Ready to Install** page, select **Install**.

	Note: Once you select Install , the installation process begins. This could take a few minutes to complete.
---	---

- p) On the **Complete** page, verify that the installation has completed successfully and select **Close**.
- q) Close the SQL Server Installation Center window.



Check with participants to ensure that they were able to progress to the **Ready to Install** page of the **Install a SQL Server Failover Cluster** dialog box without encountering any problems.

ACTIVITY 10-3

Adding a Node to a Cluster

Before You Begin

Switch to the Server4 VM.

Scenario

You have created a failover cluster with the Server3 node. However, to ensure that the applications accessing the network are always serviced, you want to add another node that takes over the operations whenever Server4 fails.

1. Launch the Installation wizard.
 - a) Open Windows Explorer and navigate to the C:\SQL Server 2012 Installation folder.
 - b) Double-click the **Setup** file to launch the **SQL Server Installation Center**.
 - c) In the SQL Server Installation Center window, in the left pane, select **Installation**.
 - d) In the right pane, select the **Add node to a SQL Server failover cluster** link.



Note: In the **SQL Server 2012 Setup** dialog box, the **Setup Support Rules** operation identifies problems that might occur during the installation. This operation might take a few minutes to complete. Once the **Setup Support Rules** operation completes the task of problem detection, a message indicating that the operation is complete appears.

 - e) On the **Product Key** page, verify that the **Specify a free edition** option is selected and select **Next**.
 - f) On the **License Terms** page, check the **I accept the license terms** check box and select **Next**.
 - g) In the **SQL Server 2012** dialog box, on the **Setup Support Rules** page, select **OK**.
 - h) On the **Product Updates** page, uncheck the **Include SQL Server product updates** check box and select **Next**.
 - i) On the **Install Setup Files** page, select **Install**.
2. Specify the settings to start the installation of the node.
 - a) On the **Setup Support Rules** page, verify that no problems are detected and select **Next**.
 - b) On the **Cluster Node Configuration** page, verify that the default instance is selected on the **SERVER4** node and select **Next**.
 - c) On the **Cluster Network Configuration** page, in the **IP Type** column, verify that the **IPv4** check box is checked and select **Next**.
 - d) On the **Service Accounts** page, in the **SQL Server Database Engine** row, in the **Account Name** column, in the **Password** column, type **server@2012**.
 - e) In the **SQL Server Agent** row, in the **Password** column, type **server@2012** and select **Next**.
 - f) On the **Error Reporting** page, select **Next**.
 - g) On the **Add Node Rules** page, verify that the rules have completed successfully and select **Next**.
 - h) On the **Ready to Add Node** page, select **Install**.



Note: Once you select **Install**, the installation process begins. This could take a few minutes to complete.

 - i) On the **Complete** page, verify that the node has been added successfully and select **Close**.



Check with participants to ensure that they were able to progress to the **Add a Failover Cluster** dialog box without encountering any problems.



Check with participants to ensure that they were able to progress to the **Ready to Install** page of the **Install a SQL Server Failover Cluster** dialog box without encountering any problems.

ACTIVITY 10-4

Recovering From a Failed Node

Before You Begin

Switch to the Server3 virtual machine.

Scenario

You have setup two nodes on a failover cluster. You now want to test them to make sure that one node is available when another fails.

1. Test the currently active node in the cluster.
 - a) In the Server Manager window, select **Tools→Failover Cluster Manager**.
 - b) In the Failover Cluster Manager window, in the left pane, expand **Failover Cluster→SQL2012Cluster.adwcycles.local** and then expand **Nodes**.
 - c) Observe that the two nodes in the cluster, **Server3** and **Server4** are listed.
 - d) Select **Server3**.
 - e) In the right pane, observe that the Server3 node is currently running in the **SQLDefaultCluster** role.
 - f) Select **Server4**.
 - g) In the right pane, observe that there are no roles for this node indicating that this node is not running currently.
 - h) From the **Start** screen, launch SSMS.
 - i) In the **Connect to Server** dialog box, in the **Server Name** drop-down list, verify that **SQL2012Cluster** is selected and select **Connect**.
 - j) Observe that the Server3 node is connected to the SQL2012Cluster and it is currently active.

2. Change the preferred owner of the cluster from Server3 to Server4.
 - a) In the Failover Cluster Manager window, in the left pane, select **Server3**.
 - b) In the middle pane, select **SQLDefaultCluster**.
 - c) In the **Actions** pane, in the **SQLDefaultCluster** section, scroll down and select **Properties**.
 - d) In the **SQLDefaultCluster Properties** dialog box, in the **Preferred Owners** list, check both the **Server3** and **Server4** check boxes.
 - e) Select **Server4** and select **Up** to move it to the first position in the list.
 - f) Select **OK** to save the changes.

3. Test failover between Server3 and Server4.
 - a) Switch to the physical machine.
 - b) In the **Hyper-V Manager**, select **Server3**.
 - c) In the **Actions** pane, in the **Server3** section, scroll down and select **Pause**.
 - d) Observe that Server3 has stopped running.
 - e) Switch to **Server4** and launch the Failover Cluster Manager.
 - f) In the Failover Cluster Manager window, in the left pane, expand **Failover Cluster→SQL2012Cluster.adwcycles.local** and then expand **Nodes**.
 - g) In the left pane, observe that the Server3 icon has a red arrow indicating that it is offline.



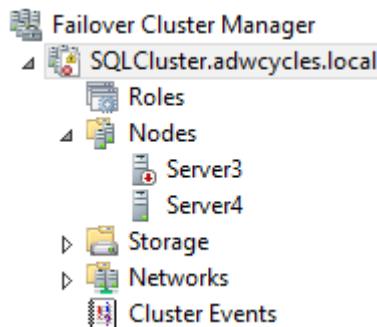
Check with participants to ensure that Server3 is the currently running node on their systems.



Check with participants to ensure that they are able to connect to SQL2012Cluster in Server3.



Check with participants to ensure that Server3 has stopped running in their systems.



- h) Select **Server4** and in the right pane observe that a failover has occurred and Server4 is running in **SQLDefaultCluster**.
- i) From the **Start** screen, launch SSMS.
- j) In the **Connect to Server** dialog box, in the **Server Name** drop-down list, verify that **SQL2012Cluster** is selected and select **Connect**.
- k) Observe that the Server4 node is connected to SQL2012Cluster and it is currently active.



Check with participants to ensure that Server4 is the currently active node in their systems. In case Server4 appears to be inactive, ask participants to wait until it comes back online.



Check with participants to ensure that they are able to connect to SQL2012Cluster in Server4. In case Server4 is not connected, ask participants to select the **Refresh** button in the **Object Explorer** pane.



Check with participants to ensure that Server3 indicates the correct system state. If necessary, ask participants to wait or refresh the window until the node appears online.

4. Perform a manual failover from Server4 to Server3.

- a) Switch to the physical system, and in the Hyper-V Manager, select **Server3**.
- b) In the **Actions** pane, in the **Server3** section, select **Resume**.
- c) Switch to the Server3 virtual machine.
- d) In the Failover Cluster Manager window, in the left pane, right-click the **Server3** node and select **Refresh**.
- e) In the Failover Cluster Manager window, observe that the red arrow has disappeared indicating that the node is running currently.
- f) Switch to the Server4 virtual machine.
- g) In the Failover Cluster Manager window, right click **SQLDefaultCluster** and select **Move→Select Node**.
- h) In the **Move Clustered Role** dialog box, verify that **Server3** is selected and select **OK**.
- i) In the Failover Cluster Manager window, observe that Server4 shows a **Pending** state and then disappears.
- j) Select **Server3** and observe that it is in the **Running** state.

TOPIC B

Work with Replication

You have set up failover clusters. Replication is another high availability solution in SQL Server that enables copies of synchronized data to be available at different locations at the same time. This technique enables administrators to easily manage data in multiple locations. In this topic, you will work with replication.

Replication

Replication is a SQL Server feature that copies and distributes database objects and data from one database server to another. Databases across the network are synchronized so that changes in one database are updated in the other copies too. Replication enables local users to access data locally and provides a real-time backup in case of a failure. A SQL Server instance that publishes the data and database objects to be replicated is called a *publisher* and the instance that receives the published data is called the *subscriber*. This method of using a publisher and subscriber is also referred to as the Pub-Sub model.

The Replication Process

In addition to the publisher and subscriber, replication involves two other entities—the Log Reader Agent and Distribution Agent. The replication process involves the following steps.

1. The server instance, which is identified as the publisher generates an initial snapshot of database objects and data and transmits them to the subscribers.
2. Whenever there are changes to the data and the schema in the publisher, the changes are transmitted to the subscribers for synchronization in real-time.
3. The Log Reader Agent reads the changes from the database log and then informs the Distribution Agent.
4. The Distribution Agent immediately sends the changes to all the subscribers of that database.



Ask the students to identify the benefits of maintaining copies of a database using replication. Ask them to list the critical databases in their enterprise that would benefit from replication.



The Replication Process

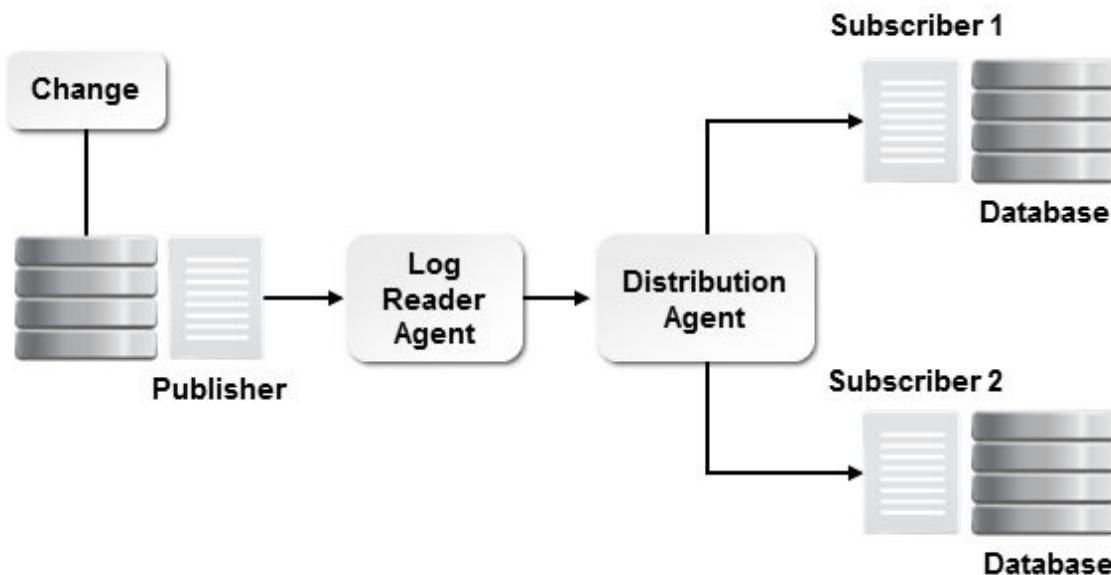


Figure 10-3: Data and schema changes in a database replicated in two subscribers.

Types of Replication

There are three types of replication in SQL Server—transactional replication, merge replication, and snapshot replication.

Type of Replication	In this Mode
Transactional replication	Whenever there are changes to the data and database objects in the publisher, the changes are pushed to the subscriber for synchronization. Typically, transactional replication is used in server to server scenarios because it helps improve scalability, ensure data availability, and enable geographical distribution of servers.
Merge replication	A central data store serves as the main replication source and target. Other replication nodes synchronize with this store. Filters are used to avoid conflicts during synchronization. Merge replication can be used in server to client scenarios such as mobile applications or Point-of-Sale kiosks.
Snapshot replication	A snapshot of data and database objects is distributed to the subscribers. This mode is primarily used to provide an initial step to the other two modes. During synchronization, the entire snapshot is generated and distributed again. Snapshot replication is suitable in cases where large volumes of data changes occur.



Note: Normally in transactional replication, only publishers can push data to the subscribers and the data in subscribers is considered read-only. However, SQL Server 2012 provides options to update data on subscribers and synchronize the changes with the publisher.



Discuss the advantages of each type of replication and ask the students to identify scenarios in which they would use each type of replication.



You may want to show **LearnTO Implement Replication Strategies** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOs for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.

Guidelines to Select a Replication Strategy

To ensure that replication works as required, you need to select the right type of replication that suits your requirements and monitor the performance of the replication strategy. Some of the best practices to implement a replication strategy include:

- Determining the requirements of your setup and selecting the right strategy based on the requirements.
- Backing up the publication, distribution, and subscription databases along with the master and msdb databases on each node.
- Automating recovery from a replicated database.
- Optimizing network connectivity between each node.
- Using alerts to generate notifications in case of any issues.



Note: For more information about replication strategies, check out the **LearnTO Implement Replication Strategies** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Work with Replication

ACTIVITY 10–5

Configuring the Distributor

Before You Begin

Switch to the physical machine. In SSMS, in the Object Explorer pane, ensure that both the default instance and the INSTANCE01 instance are running.

Perform the steps to delete the existing SalesDB database.

1. In the Object Explorer pane, expand **SQL01** and select **Databases**.
2. Select the **SalesDB** database.
3. Right-click and select **Delete**.
4. In the **Delete Object** dialog box, select **OK** to delete the database.
5. In the **Object Explorer** pane, verify that the SalesDB database has been removed.

Scenario

You wish to replicate the SalesDB database so that a copy of the database is available on one instance of the server even if another instance fails. As the first step in the replication process, you want to configure the distributor.

1. Configure SQL01\Instance01 using the **Configure Distribution** wizard.
 - a) In the **Object Explorer** pane, in **SQL01\Instance01**, right-click **Replication** and select **Configure Distribution**.
 - b) In the **Configure Distribution** wizard, in the welcome screen, select **Next**.
 - c) On the **Distributor** page, verify that the '**SQL01\INSTANCE01**' will act as its own Distributor; SQL Server will create a distribution database and log option is selected and select **Next**.

The Distributor is the server responsible for storing replication information used during synchronizations.

- 'SQL01\INSTANCE01' will act as its own Distributor; SQL Server will create a distribution database and log
- Use the following server as the Distributor (Note: the server you select must already be configured as a Distributor):

- d) On the **SQL Server Agent Start** page, select the **No, I will start the SQL Server Agent service manually** option and select **Next**.
- e) On the **Snapshot Folder** page, observe the default folder path and select **Next**.
- f) On the **Distribution Database** page, observe the default values in the **Distribution database name** text box and the folders where the database and log files are stored and select **Next**.



Ask participants to check whether the **SQL Server Agent** service appears with a green arrow in the **Object Explorer** pane. If the service has not yet started, then ask them to select the **Yes, configure the SQL Server Agent service to start automatically** option.

Distribution database name:	<input type="text" value="distribution"/>
Folder for the distribution database file:	<input type="text" value="C:\Program Files\Microsoft SQL Server\MSSQL11.INSTANCE01\MSSQL\Data"/>
Folder for the distribution database log file:	<input type="text" value="C:\Program Files\Microsoft SQL Server\MSSQL11.INSTANCE01\MSSQL\Data"/>
The paths must refer to disks that are local to the Distributor and begin with a local drive letter and colon (for example, C:). Mapped drive letters and network paths are invalid.	

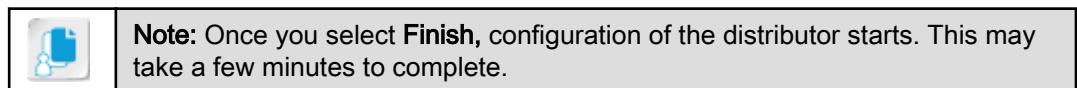
2. Enable publishers to use the distributor.
 - a) On the **Publishers** page, verify that the **SQL01\INSTANCE01** server is selected for distribution and select **Next**.

Publishers:		Distribution Database	
<input checked="" type="checkbox"/>	SQL01\INSTANCE01	distribution	[...]

- b) On the **Wizard Actions** page, verify that the **Configure distribution** check box is checked and select **Next**.

At the end of the wizard:	
<input checked="" type="checkbox"/>	Configure distribution
<input type="checkbox"/>	Generate a script file with steps to configure distribution

- c) On the **Complete the Wizard** page, select **Finish**.



- d) Verify that the configuration has been completed successfully and select **Close**.



Check with participants to ensure that they have completed the configuration successfully without any errors.

ACTIVITY 10-6

Publishing a Database for Replication

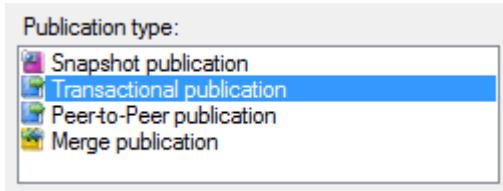
Before You Begin

In INSTANCE01, ensure that the SQL Server Agent Service is running.

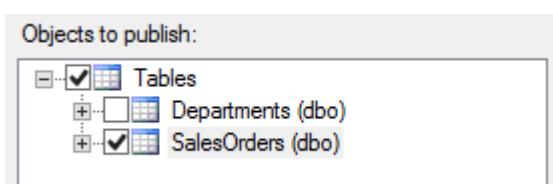
Scenario

Now that you have configured the server instance for distribution, you want to perform the next step in the replication process. You want to publish the SalesDB database so that it is available for replication.

1. Select the database objects to be published.
 - a) In the **Object Explorer** pane, in the SQL01\Instance01 server instance, expand **Replication**.
 - b) Right-click **Local Publications** and select **New Publication**.
 - c) In the **New Publication** wizard, in the welcome screen, select **Next**.
 - d) On the **Publication Database** page, in the list of databases, select the **SalesDB** database and select **Next**.
 - e) On the **Publication Type** page, select **Transactional publication** and select **Next**.

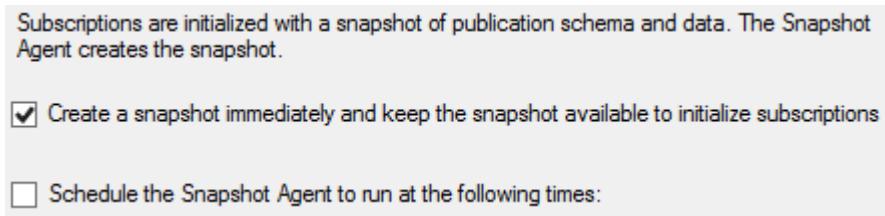


- f) On the **Articles** page, expand the **Tables** object, select the **SalesOrders (dbo)** check box and select **Next**.



Check with participants to make sure that SalesOrders is available in the list.

- g) On the **Filter Table Rows** page, select **Next**.
- h) On the **Snapshot Agent** page, check the **Create a snapshot immediately and keep the snapshot available to initialize subscription** check box and select **Next**.



2. Configure the security settings.

- On the **Agent Security** page, select **Security Settings**.
- In the **Snapshot Agent Security** dialog box, verify that the **Run under the following Windows account** option is selected.
- In the **Process account** text box, type **SQL01\Administrator**.
- In the **Password** text box, type **server@2012**.
- In the **Confirm Password** text box, type **server@2012**.
- In the **Connect to the Publisher** section, verify that the **By impersonating the process account** option is selected and select **OK**.



- In the **New Publication** wizard, select **Next**.
- On the **Wizard Actions** page, verify that the **Create the publication** check box is checked and select **Next**.
- On the **Complete the Wizard** page, in the **Publication name** text box, type **SalesPub** and select **Finish**.



Note: Once you select **Finish**, configuration of the publication starts. This may take a few minutes to complete.

- On the **Creating Publication page**, verify that the publication has been configured successfully and select **Close**.



Check with participants to make sure that they have completed the configuration successfully without any errors.

 Success	3 Total	0 Error
	3 Success	0 Warning
Details:		
Action	Status	Message
 Creating Publication 'SalesPub'	Success	
 Adding article 1 of 1	Success	
 Starting the Snapshot Agent	Success	

ACTIVITY 10-7

Subscribing to a Publication

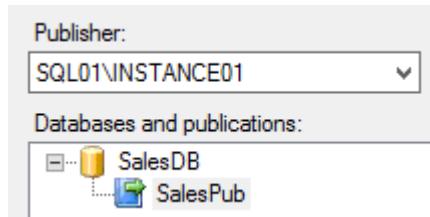
Before You Begin

Ensure that the SQL Server Agent is running on both the instances. If it is not running, then right-click the agent and select **Start**.

Scenario

You have published the SalesDB database in the SQL01\INSTANCE01 instance for replication. You now want to replicate the database in the SQL01 default instance by subscribing to the publication.

1. Select the publication you want to subscribe.
 - a) In the **Object Explorer** pane, expand the SQL01 default instance folder.
 - b) Expand the **Replication** folder, right-click **Local Subscriptions**, and select **New Subscription**.
 - c) In the **New Subscription** wizard, in the welcome screen, select **Next**.
 - d) On the **Publication** page, from the **Publisher** drop-down list, select **Find SQL Server Publisher**.
 - e) In the **Connect to Server** dialog box, from the **Server Name** drop-down list, select **SQL01\INSTANCE01** and select **Connect**.
 - f) On the **Publication** page, verify that the **SalesPub** publication appears.



- g) Select the **SalesPub** publication and select **Next**.
2. Specify the publication settings.
 - a) On the **Distribution Agent Location** page, select the **Run all agents at the Distributor, SQL01\INSTANCE01 (push subscriptions)** option and select **Next**.

For the subscriptions I create in this wizard:

Run all agents at the Distributor, SQL01\INSTANCE01 (push subscriptions)

This option makes it easier to administer the synchronization of subscriptions centrally.

Run each agent at its Subscriber (pull subscriptions)

This option reduces the processing overhead at the Distributor and lets each Subscriber administer the synchronization of its subscription.
 - b) On the **Subscribers** page, from the drop-down list in the **Subscription Database** column, select **<new database>**.
 - c) In the **New Database** dialog box, in the **Database Name** text box, type **SalesDB01** and select **OK**.

- d) On the **Subscribers** page, select **Next**.
3. Configure the distribution agent security settings.
- On the **Distribution Agent Security** page, select the **ellipsis (...)** button.
 - In the **Distribution Agent Security** dialog box, verify that the **Run under the following Windows account** option is selected.
 - In the **Process Account** text box, type **SQL01\Administrator**.
 - In the **Password** text box, type **server@2012**.
 - In the **Confirm Password** text box, type **server@2012**.

Run under the following Windows account:

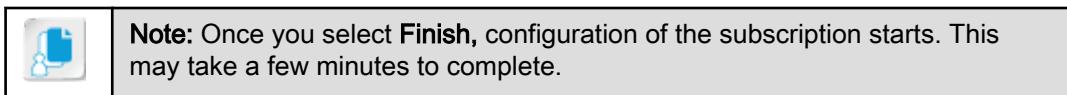
Process account:	SQL01\Administrator
Example: domain\account	
Password:	*****
Confirm Password:	*****

Run under the SQL Server Agent service account (This is not a recommended security best practice.)

- In the **Connect to the Subscriber** section, verify that the **By impersonating the process account** option is selected and select **OK**.
- In the **New Subscription** wizard, select **Next**.
- On the **Synchronization Schedule** page, in the **Agent Schedule** column, verify that **Run continuously** is selected and select **Next**.

Agent schedule:		
Subscriber	Agent Location	Agent Schedule
SQL01	Distributor	Run continuously

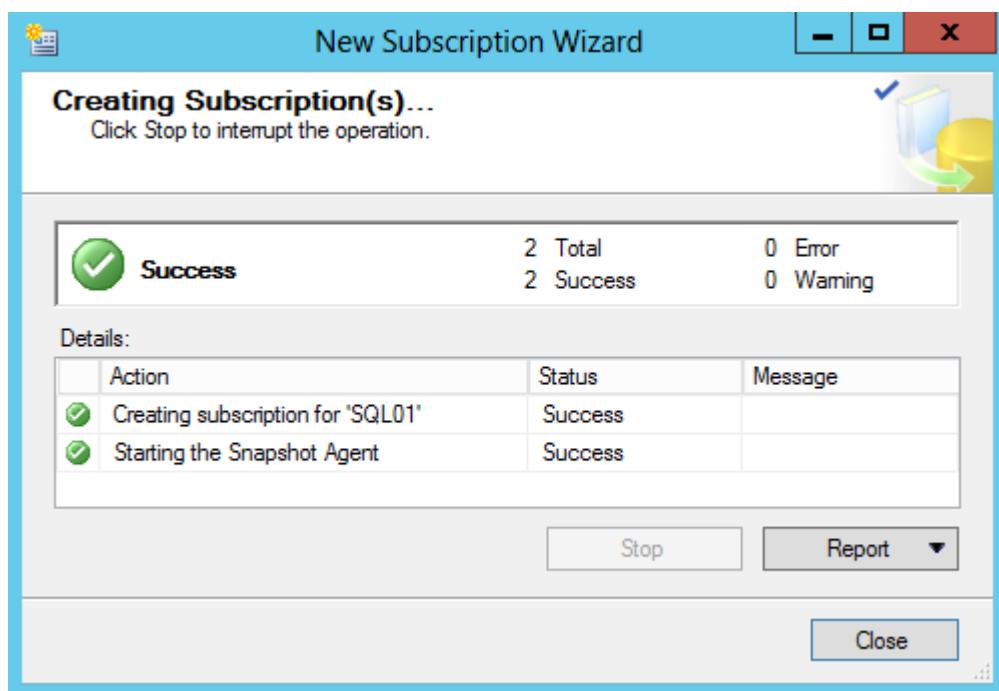
4. Create the subscription.
- On the **Initialize Subscriptions** page, verify that the check box in the **Initialize** column is checked and select **Next**.
 - On the **Wizard Actions** page, verify that the **Create the subscription(s)** check box is checked and select **Next**.
 - On the **Complete the Wizard** page, select **Finish**.



- On the **Creating Subscriptions** page, verify that the subscription has been created successfully and select **Close**.



Check with participants to make sure that they have completed the configuration successfully without any errors.



ACTIVITY 10-8

Testing Replication of Data Between Servers

Before You Begin

In the SQL01 default instance, expand the **Databases** node and verify whether the SalesDB01 database exists. If it does not exist, then right-click the **Databases** node and select **Refresh**.

Scenario

You have implemented the replication of SalesDB database by setting up a subscriber to the published database. You now want to test whether the replication works as intended. You want to do this by testing whether the changes performed on the publisher are available on the subscriber.

1. Display the rows in the subscriber table.

a) In the SQL01 default instance, in the **Databases** node, select **SalesDB01**.

b) In the standard toolbar, select **New Query**.

c) In the **Query Editor** pane, type the query.

```
SELECT * FROM SalesOrders;
```

d) Select **Execute** to execute the query.

e) Observe the rows displayed on the **Results** tab.

	SalesOrderID	SalesDate	ProductID	ListPrice	Quantity	TotalSalesAmount
1	1	2013-01-01	1	100.2	10	501
2	2	2013-01-01	1	100.2	5	501
3	3	2013-01-01	1	100.2	5	501
4	4	2013-01-01	1	100.2	5	501
5	5	2013-01-01	1	100.2	5	501

f) Close the **Query Editor** pane without saving the changes.

2. Insert rows into the publisher database.

a) In **Object Explorer** pane, expand the **SQL01\INSTANCE01** server instance.

b) Expand the **Databases** node and select **SalesDB**.

c) In the standard toolbar, select **New Query**.

d) Type the code to insert the SalesDate, ProductID, ListPrice, Quantity, and TotalSalesAmount columns into the SalesOrders table.

```
INSERT INTO dbo.SalesOrders
(SalesDate, ProductID, ListPrice, Quantity, TotalSalesAmount)
```

e) Enter the code to provide the list of values to be stored in five rows of the SalesOrders table.



Note: You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```
INSERT INTO dbo.SalesOrders
(SalesDate, ProductID, ListPrice, Quantity, TotalSalesAmount)
VALUES
('2013-01-10', 3, 10.00, 4, 40.00),
('2013-01-10', 1, 21.00, 3, 63.00),
```

```
('2013-01-11', 4, 30.00, 4, 120.00),
('2013-01-12', 2, 150.00, 1, 150.00),
('2013-01-12', 3, 10.00, 2, 20.00);
```

- f) Execute the query.
- g) Close the **Query Editor** pane without saving the changes.

3. Test the subscriber database for replication.

- a) In the SQL01 default instance, in the **Databases** node, select **SalesDB01**.
- b) In the standard toolbar, select **New Query**.
- c) In the **Query Editor** pane, type the query.

```
SELECT * FROM SalesOrders;
```

- d) Execute the query.
- e) Observe that the table displays the new rows that you added to the publisher database.

	SalesOrderID	SalesDate	ProductID	ListPrice	Quantity	TotalSalesAmount
5	5	2013-01-01	1	100.2	5	501
6	6	2013-01-01	1	100.2	5	501
7	7	2013-01-01	1	100.2	5	501
8	8	2013-01-01	1	100.2	5	501
9	9	2013-01-01	1	29.95	3	89.85
10	10	2013-01-10	3	10	4	40
11	11	2013-01-10	1	21	3	63
12	12	2013-01-11	4	30	4	120
13	13	2013-01-12	2	150	1	150
14	14	2013-01-12	3	10	2	20



Note: If the new rows are not displayed, then refresh the SalesDB01 database and run the query again.

TOPIC C

Work with Mirroring and AlwaysOn

In addition to failover clustering and replication, SQL Server provides two more high availability solutions. While database mirroring provides a high availability solution at the database level, the AlwaysOn feature provides high availability and disaster recovery solutions at the enterprise level. In this topic, you will work with database mirroring and AlwaysOn.

Database Mirroring

Database mirroring is a high availability technique in which two copies of a database are maintained on different SQL Server instances. The original database is called the principal database. The copy of the database is called the mirror database. Database mirroring enables quick failover to the mirror database in case the principal database fails. When the principal database is active, the mirror database is inaccessible to users. If a failure occurs, then the mirror database becomes active.

Database Mirroring Performance Monitoring can be performed by using the Database Mirroring Monitor or the Windows Performance Monitor and adding the relevant mirroring counters.

Disadvantages of Database Mirroring

One of the disadvantages of using database mirroring is that the mirror database on an SQL Server instance is inaccessible when the principal database is active. The mirror database becomes active and accessible only when a switch occurs. Another disadvantage is that only one database from a server can be mirrored onto another server. If you need to mirror more databases, you need to mirror them onto different servers.

Difference Between Replication and Database Mirroring

In a database mirroring implementation, only one copy of a database is active at a time. The mirror database is inactive until a failover occurs. However, in replication, all the copies of a database are active at multiple nodes simultaneously.



Note: The database mirroring feature in SQL Server is expected to be discontinued in future versions. In SQL Server 2012, this feature exists only to support backward compatibility. As database administrator, you need to consider using newer techniques, such as AlwaysOn, to ensure high availability of your databases.

The AlwaysOn Feature

AlwaysOn is a new high-availability and disaster recovery feature in SQL Server 2012. It provides an alternative to database mirroring at the enterprise level. AlwaysOn works on top of the WSFC infrastructure and is implemented through AlwaysOn Availability Groups (AAG). These groups contain a set of databases called Availability Databases (AD) that failover as one unit.

Each AAG supports five AlwaysOn Availability Replicas (AAR). An AAR is an instance of an AAG that resides in an SQL Server instance and maintains a copy of all the databases in the AAG. In an AAG, there can be up to four secondary replicas and one primary replica. AlwaysOn supports automatic, planned manual, and unplanned forced failover.



The AlwaysOn Feature

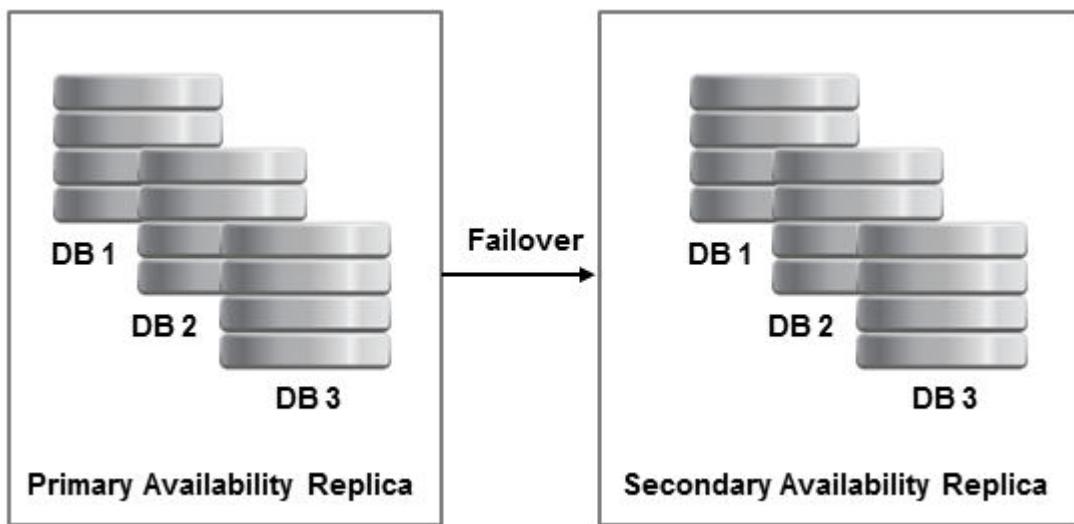


Figure 10-4: The primary and secondary AlwaysOn Availability Replicas.

	Note: Ask the students to identify the differences between database mirroring and AlwaysOn. Discuss the advantages of using AlwaysOn over database mirroring.
	Note: For more information about AlwaysOn, check out the LearnTO Implement High Availability with Mirroring and AlwaysOn presentation by selecting the LearnTO tile on your LogicalCHOICE Course screen.



You may want to show LearnTO Implement High Availability with Mirroring and AlwaysOn from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOS for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.

	Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Work with Mirroring and AlwaysOn
--	---

ACTIVITY 10–9

Setting Up Mirroring on a Database

Before You Begin

Specify the login credentials of the two SQL Server instances.

1. Press the **[WINDOWS]** key to launch the **Start** screen.
2. Select **SQL Server Configuration Manager**.
3. In the SQL Server Configuration Manager window, in the left pane, select **SQL Server Services**.
4. In the right pane, right-click **SQL Server (MSSQLSERVER)** and select **Properties**.
5. In the **SQL Server (MSSQLSERVER) Properties** dialog box, select the **LogOn** tab.
6. In the **Username** text box, replace the existing username with **Administrator**.
7. In the **Password** text box, type **server@2012**.
8. In the **Confirm Password** text box, type **server@2012** and select **OK**.
9. In the message box that appears, select **Yes** to restart the service.
10. In the SQL Server Configuration Manager window, modify the logon credentials of the **SQL01\INSTANCE01** service with the same username and password that you specified for the **SQL01** default instance.

Scenario

The AdventureWorks2012 database contains all the data necessary for the day-to-day business. Because this database is critical to the functioning of AdventureWorks Cycles, you want to ensure that in case of a failure in one server instance, a copy of the database is always available on another server instance. To ensure the availability of the AdventureWorks2012 database in case of a failure, you want to set up a database mirror.

1. Perform a full backup of the principal database.
 - a) In the **Object Explorer** pane, in the **SQL01** default instance, right-click the **AdventureWorks2012** database and select **Tasks→Backup**.
 - b) In the **Backup Database — AdventureWorks2012** dialog box, in the **Backup type** drop-down list, verify that **Full** is selected.
 - c) In the **Destination** section, select **Add**.
 - d) In the **Select Backup Destination** dialog box, select the **Ellipsis** button.
 - e) In the **Locate Database Files — SQL01** dialog box, navigate to the **C:\Temp** folder.
 - f) In the **File name** text box, type **AdventureWorks_Latest.bak** and select **OK**.
 - g) In the **Destination** section, select **C:\Temp\AdventureWorks_Latest.bak**. Remove the path **C:\Temp\Backup\ADW2012.bak**.
 - h) In the **Select Backup Destination** dialog box, select **OK**.
 - i) In the **Microsoft SQL Server Management Studio** message box, select **OK** to close the database backup completion message.
2. Perform a backup of the transaction log.
 - a) In the **Object Explorer** pane, in the **SQL01** default instance, right-click the **AdventureWorks2012** database and select **Tasks→Backup**.
 - b) In the **Backup Database — AdventureWorks2012** dialog box, from the **Backup Type** drop-down list, select **Transaction Log**.
 - c) In the **Destination** section, select **Remove** to remove the existing path and select **Add**.
 - d) In the **Select Backup Destination** dialog box, select the **Ellipsis** button.
 - e) In the **Locate Database Files — SQL01** dialog box, navigate to the **C:\Temp** folder.

- f) In the **File name** text box, type **Adventure_Latest_log.bak** and select **OK**.
 - g) In the Back Up database window, select **OK**.
 - h) In the **Microsoft SQL Server Management Studio** message box, select **OK** to close the database backup completion message.
3. Restore the database on the mirror.
- a) In the **Object Explorer** pane, in the **SQL01\INSTANCE01** server instance, right-click **Databases** and select **Restore Database**.
 - b) In the **Restore Database** dialog box, select the **Device** option.
 - c) Select the **ellipsis (...)** button to the right of the **Device** text box.
 - d) In the **Select backup devices** dialog box, select **Add** and navigate to the **C:\Temp** folder, select the **AdventureWorks_Latest.bak** file, and select **OK**.
 - e) In the **Select a page** section, select **Options**.
 - f) In the **Restore options** section, from the **Recovery state** drop-down list, select the **RESTORE WITH NORECOVERY** option.
 - g) In the **Select a page** section, select **Files**.
 - h) In the **Restore database files as** section, check the **Restore all files to folder** check box and select **OK**.
 - i) In the **Microsoft SQL Server Management Studio** message box, select **OK**.
 - j) In the **Restore Database** dialog box, select **OK**.
 - k) In the **Object Explorer** pane, verify that the restored database is displayed.
4. Restore the database log.
- a) In the **Object Explorer** pane, right-click the **Databases** node and select **Restore Files and Filegroups**.
 - b) In the **Restore Files and Filegroups** window, in the **Destination to restore** section, from the **To database** drop-down list, select **AdventureWorks2012**.
 - c) In the **Source for restore** section, select the **From device** option and select the **ellipsis** button.
 - d) In the **Select backup devices** dialog box, select **Add**.
 - e) In the **Locate Backup File** window, navigate to **C:\Temp**, select the **Adventure_Latest_log.bak** and select **OK**.
 - f) In the **Select the backup sets to restore** section, check the check box in the **Restore** column.
 - g) In the **Select a page** section, select **Options**.
 - h) In the **Recovery state** section, select the second option which ends with (RESTORE WITH NORECOVERY) and select **OK**.
 - i) In the **Microsoft SQL Server Management Studio** message box, select **OK**.
5. Configure the principal server.
- a) In the **SQL01** default instance, right click **AdventureWorks2012** and select **Tasks→Mirror**.
 - b) In the **Database Properties — AdventureWorks2012** dialog box, select **Configure Security**.
 - c) In the **Configure Database Mirroring Security** wizard, on the welcome page, select **Next**.
 - d) On the **Include a Witness Server** page, select the **No** option and select **Next**.
 - e) On the **Principal Server Instance** page, in the **Principal server instance** drop-down list, verify that **SQL01** is selected.
 - f) Observe the default values provided in the **Listener port** and the **Endpoint name** text boxes and select **Next**.

Principal server instance:

SQL01

Specify the properties of the endpoint through which the principal server instance will accept connections from the mirror and witness server instances:

Listener port: 5022 Encrypt data sent through this endpoint

Endpoint name: Mirroring

NOTE: If the principal, mirror or witness are instances on the same server, their endpoints must use different ports.

6. Configure the mirror instance and complete the configuration process.
 - a) On the **Mirror Instance** page, in the **Mirror server instance** drop-down list, verify that **SQL01\INSTANCE01** is selected.
 - b) In the **Connect to Server** dialog box, verify the server name and the authentication mode and select **Connect**.
 - c) Observe the default values provided in the **Listener port** and the **Endpoint name** text boxes and select **Next**.

Mirror server instance:

SQL01\INSTANCE01

Specify the properties of the endpoint through which the mirror server instance will accept connections from the principal and witness server instances:

Listener port: 5023 Encrypt data sent through this endpoint

Endpoint name: Mirroring

NOTE: If the principal, mirror or witness are instances on the same server, their endpoints must use different ports.

- d) On the **Service Accounts** page, select **Next**.
- e) On the **Complete the Wizard** page, select **Finish**.



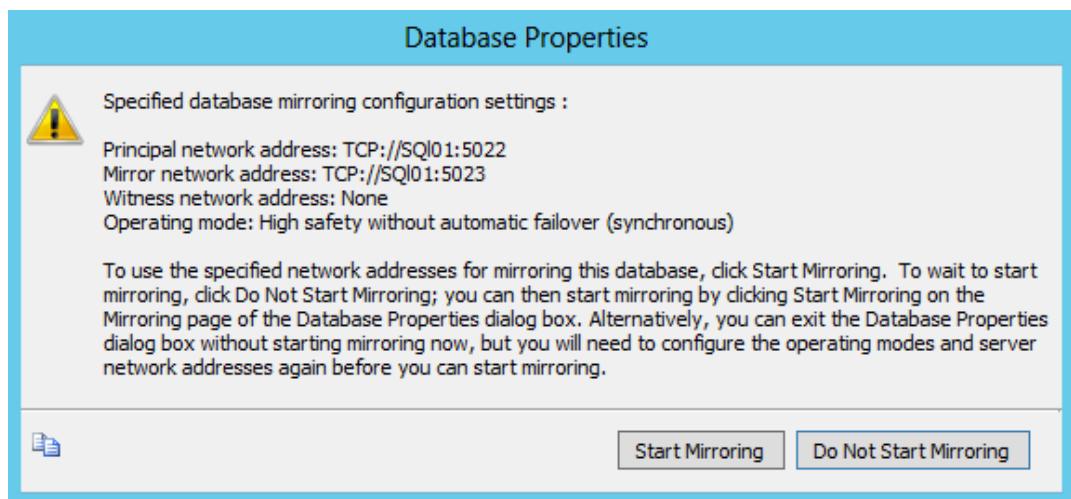
Note: Once you select **Finish**, the configuration process starts. It might take a few minutes to complete the configuration of the endpoints.

Success 2 Total 0 Error
2 Success 0 Warning

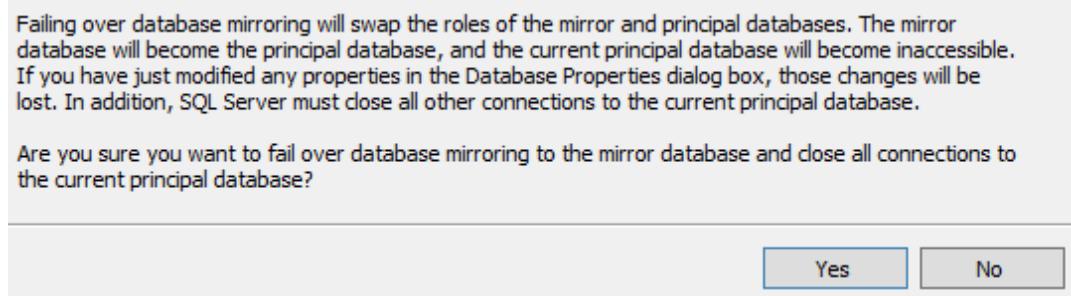
Details:

Action	Status	Message
Configuring endpoint on principal server(...)	Success	
Configuring endpoint on mirror server(SQ...	Success	

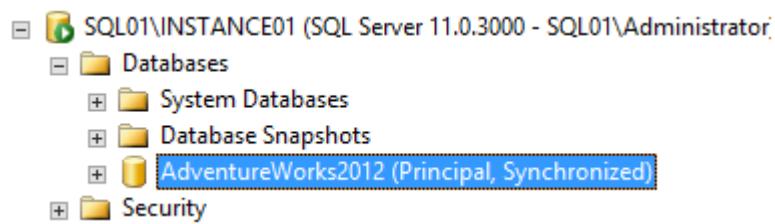
- f) In the **Configure Database Mirroring Security** wizard, on the **Configuring Endpoints** page, verify that the endpoints are configured successfully and select **Close**.
7. Initiate database mirroring.
- a) In the **Database Properties** message box, select **Start Mirroring**.



- b) In the **Database Properties** message box that requests you to specify whether to continue configuration without using fully qualified domain names for the server addresses, select **Yes**.
- c) In the **Database Properties** dialog box, select **OK**.
- d) In the **Object Explorer** pane, in the SQL01 default instance, observe that the AdventureWorks2012 database appears as the principal instance and in the SQL01\INSTANCE01 instance the database appears as the mirror.
8. Verify that database mirroring has completed successfully.
- a) In the SQL01 default instance, right-click **AdventureWorks2012** and select **Tasks->Mirror**.
- b) In the **Database Properties** dialog box, select **Failover**.
- c) In the **Database Properties** message box, select **Yes**.



- d) In the **Database Properties** dialog box, select **OK**.
- e) In the **Object Explorer** pane, in the SQL01 default instance, observe that the AdventureWorks2012 database appears as the mirror instance and in the SQL01\INSTANCE01 instance the database appears as the primary.



ACTIVITY 10-10

Configuring AlwaysOn on a Database

Before You Begin

Install a non-clustered instance of SQL Server 2012 named **NCINSTANCE01** on Server3 and another non-clustered instance named **NCINSTANCE02** on Server4. In both instances, select only the Database Engine feature and complete the installation wizard by selecting the current user (Administrator) on the required steps.

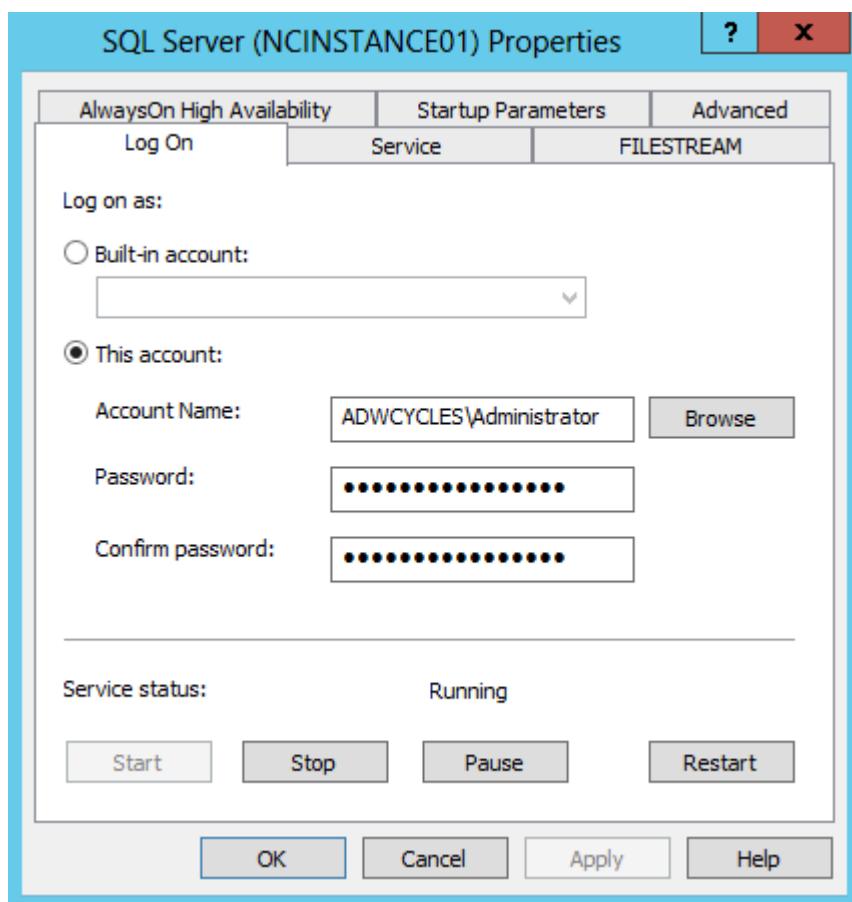
Scenario

As database administrator at Adventure Works Cycles, you want to ensure that the failure of physical systems in the network do not affect the services offered to clients. To achieve this, you want to configure AlwaysOn so that if one node in the network fails, another node takes over and provides access to the required databases.

1. Enable the AlwaysOn Availability Groups.
 - a) In the Server3 VM, from the **Start** screen, launch the **SQL Server Configuration Manager**.
 - b) In the SQL Server Configuration Manager window, in the left pane, select **SQL Server Services**.
 - c) In the right pane, right-click **MSSQLServer** and select **Properties**.
 - d) In the **SQL Server (MSSQLSERVER) Properties** dialog box, select the **AlwaysOn High Availability** tab.
 - e) In the **Windows failover cluster name** text box, verify that the failover cluster you created is displayed.
 - f) Check the **Enable AlwaysOn Availability Groups** check box.
 - g) Select the **Log On** tab.
 - h) Change the login credentials to **ADWCYCLES\Administrator** and the password to **server@2012**.
 - i) Select **OK** to save the changes.
 - j) In the message box that appears, select **Yes**.
 - k) In the SQL Server Configuration Manager window, right-click the **SQLServer (MSSQLSERVER)** service and select **Restart**.
 - l) Switch to Server4 and enable **AlwaysOn Availability Groups** in **NCINSTANCE02** and specify the same login credentials that you set in Server3.



Check with participants to ensure that they have enabled AlwaysOn Availability Groups and specified the login credentials on both the server VMs.



2. Restore the AdventureWorks2012 database.

- In the Server4 VM, launch SSMS and connect to the NCINSTANCE02 instance.
- In the **Object Explorer** pane, expand NCINSTANCE02, right-click **Databases**, and select **Restore Database**.
- In the **Restore Database** dialog box, in the **Source** section, select **Device** and select the **ellipsis (...)** button to the right of the **Device** text box.
- In the **Select backup devices** dialog box, select the ADW2012.bak file, and select **OK**.
- In the **Restore Database – AdventureWorks2012** dialog box, in the left pane, select **Options**.
- From the **Recovery state** drop-down list, select **RESTORE WITH NORECOVERY** and select **OK**.



Note: Once you select **OK**, it takes a few minutes for the database to be restored.

- In the message box that appears, observe that the database has been restored successfully and select **OK**.
- Observe that in the **Object Explorer** pane, the word **Restoring** appears to the right of the AdventureWorks2012 database.

3. Attach the AdventureWorks2012 database on Server3.

- Switch to the Server3 VM.
- In the Server3 VM, launch SSMS and connect to the NCINSTANCE01 instance.
- Attach the AdventureWorks2012 database.



Note: The AdventureWorks2012 database is stored on the physical machine. On the physical machine, you can copy the C:\093108Data\AdventureWorks2012_database folder and paste it at the \\server3\C\$ folder so that you can access it from the C:\ drive of the Server3 VM.

- d) In the **Object Explorer** pane, right-click **AdventureWorks2012**, and select **Properties**.
 - e) In the **Database Properties - AdventureWorks2012** dialog box, in the left pane, select **Options**.
 - f) In the right pane, from the **Recovery Model** drop-down list, select **Full** and select **OK**.
 - g) Perform a full backup of the AdventureWorks2012 database and its transaction log and save them in C:\Temp of the physical system as **ADW2012_DB.bak** and **ADW2012_trn_log.bak**, respectively.
4. Restore the database on the mirror.
- a) On Server4, launch SSMS and in the **Object Explorer** pane, in the SERVER4\NCINSTANCE02 server instance, right-click **Databases** and select **Restore Database**.
 - b) In the **Restore Database** dialog box, select the **Device** option.
 - c) Select the **ellipsis (...)** button to the right of the **Device** text box.
 - d) In the **Select backup devices** dialog box, select **Add** and navigate to the C:\Temp folder, select the **ADW2012_DB.bak** file, and select **OK**.
 - e) In the **Select a page** section, select **Options**.
 - f) In the **Restore options** section, from the **Recovery state** drop-down list, select the **RESTORE WITH NORECOVERY** option.
 - g) In the **Select a page** section, select **Files**.
 - h) In the **Restore database files as** section, check the **Restore all files to folder** check box and select **OK**.
 - i) In the **Microsoft SQL Server Management Studio** message box, select **OK**.
 - j) In the **Restore Database** dialog box, select **OK**.
 - k) In the **Object Explorer** pane, verify that the restored database is displayed.
5. Restore the database log.
- a) In the **Object Explorer** pane, right-click the **Databases** node and select **Restore Files and Filegroups**.
 - b) In the **Restore Files and Filegroups** window, in the **Destination to restore** section, from the **To database** drop-down list, select **AdventureWorks2012**.
 - c) In the **Source for restore** section, select the **From device** option and select the **ellipsis** button.
 - d) In the **Select backup devices** dialog box, select **Add**.
 - e) In the **Locate Backup File** window, navigate to C:\Temp of the physical system, select the **ADW2012_trn_log.bak** and select **OK**.
 - f) In the **Select the backup sets to restore** section, check the check box in the **Restore** column.
 - g) In the **Select a page** section, select **Options**.
 - h) In the **Recovery state** section, select the second option which ends with (RESTORE WITH NORECOVERY) and select **OK**.
 - i) In the **Microsoft SQL Server Management Studio** message box, select **OK**.
6. Create an AlwaysOn Availability Group.
- a) In the **Object Explorer** pane, expand **Server3(NCINSTANCE01)** and then expand **AlwaysOn High Availability**.
 - b) Right-click **Availability Groups** and select **New Availability Group Wizard**.
 - c) In the **Availability Group** wizard, on the **Introduction** page, select **Next**.
 - d) On the **Specify Availability Group Name** page, in the **Availability group name** text box, type **AVG_01** and select **Next**.
 - e) On the **Select Databases** page, observe that the AdventureWorks2012 database meets the prerequisites and check the **AdventureWorks2012** check box.

Select user databases for the availability group.

User databases on this instance of SQL Server:

Name	Size	Status
<input checked="" type="checkbox"/> AdventureWorks2012	189.5 MB	Meets prerequisites

- f) On the **Select Databases** page, select **Next**.

- g) On the **Specify Replicas** page, verify that **Server3\NCINSTANCE01** is already listed and check the **Automatic Failover** check box.
- h) Verify that the check box in the **Synchronous Commit** column is checked and select **Next**.
- i) Select **Add Replica**.
- j) In the **Connect to Server** dialog box, in the **Server Name** box, type the server name as **Server4\NCINSTANCE02** and select **Connect**.
- k) In the **SERVER4\NCINSTANCE02** row, check the check box in the **Automatic Failover** column.

Availability Replicas:				
Server Instance	Initial Role	Automatic Failover (Up to 2)	Synchronous Commit (Up to 3)	Readable Secondary
SERVER4\NCINSTAN...	Primary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No
SERVER3\NCINSTAN...	Secondary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No

- l) Verify that the check box in the **Synchronous Commit** column is checked and select **Next**.
- m) On the **Set the Initial Data Synchronization** page, select the **Join only** option and select **Next**.
- n) On the **Validation** page, verify that there are no errors and select **Next**.
- o) On the **Summary** page, verify the options selected and select **Finish**.
- p) On the **Results** page, verify that the wizard has completed successfully and select **Close**.

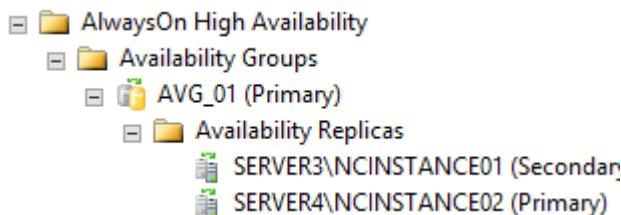
 The wizard completed successfully.

Summary:

	Name	Result
	Validating failover settings for secondary replica 'SERVER3\NCINSTANCE01'	Success
	Performing manual failover to secondary replica 'SERVER3\NCINSTANCE01'	Success
	Completing the role change for secondary replica 'SERVER3\NCINSTANCE01'	Success
	Validating WSFC quorum vote configuration.	Success

7. Check whether the availability replicas are configured properly.
- a) In the **Object Explorer** pane, expand **AVG_01 (Primary)** and then expand **Availability Replicas**.
 - b) Observe that the two servers are listed here.

 **Note:** If the servers are not listed, wait for a few minutes and then refresh the **AVG_01 (Primary)** folder.



- c) Expand **Availability Databases** and refresh the **AdventureWorks2012** database.
- d) Expand **Databases** and observe that the word **Synchronized** appears to the right of the **AdventureWorks2012** database.

8. Perform a manual failover on the availability group.
 - a) Switch to the Server4 VM.
 - b) In the **Object Explorer** pane, right-click the **AdventureWorks2012** database and select **Refresh**.
 - c) Expand **AlwaysOn High Availability** and in the **Availability Groups** node, right-click **AVG_01 (Secondary)** and select **Failover**.
 - d) In the **Failover Availability Group: AVG_01** wizard, on the **Introduction** page, select **Next**.
 - e) On the **Select New Primary Replica** page, verify that check box to the left of the **SERVER4\NCINSTANCE02** replica is checked and select **Next**.

Select the new primary replica for this availability group.

Current Primary Replica:	SERVER3\NCINSTANCE01			
Primary Replica Status:	Synchronous commit and Online			
Quorum Status:	Normal Quorum			
Choose new primary replica:				
	Server Instance	Availability Mode	Failover Mode	Failover Readiness
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> SERVER4\NCINSTA...	Synchronous com...	Automatic	No data loss

- f) On the **Summary** page, verify the options selected and select **Finish**.
- g) On the **Results** page, observe that the failover is successful and select **Close**.
9. Verify that the database is running.
 - a) In the **Object Explorer** pane, select the **AdventureWorks2012** database and create a new query.
 - b) Type the query to display all the records in the **Sales.Customer** table.

```
SELECT * FROM Sales.Customer;
```

 - c) Execute the query.
 - d) In the **Results** tab, verify that the rows in the table are displayed.

Results **Messages**

	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid
1	1	NULL	934	1	AW00000001	3F5AE95E-B87D-
2	2	NULL	1028	1	AW00000002	E552F657-A9AF-
3	3	NULL	642	4	AW00000003	130774B1-DB21-
4	4	NULL	932	4	AW00000004	FF862851-1DAA-
5	5	NULL	1026	4	AW00000005	83905BDC-6F5E-

- e) Close all the open windows on Server3 and Server4 VM.

TOPIC D

Work with Central Management Servers

You have worked with the different high availability solutions in SQL Server. These solutions typically use multiple servers, which require constant monitoring. Critical problems may arise even if one server is not monitored properly. Using the Central Management Servers feature can help database administrators maintain a list of registered servers and manage them. In this topic, you will work with central management servers.

CMSs

A *Central Management Server* (CMS) is an instance of SQL Server that stores a list of one or more SQL Server instances managed as a group. CMSs enable administrators manage multiple servers. They enable administrators to centrally manage multiple registered servers as a single group. All T-SQL commands, policies, and other actions taken on the group are processed on all the servers in the group. Some of the important points to be considered while setting up a CMS are:

- To access the registered users through CMS, an administrator can only use Windows authentication. You cannot login to a registered server using an SQL user name. However, if necessary, you can connect to the server directly without the CMS.
- The central policies you set for the network do not apply to the CMS. You have to manage the CMS separately.



CMSs



Ask the students to identify the advantages of using a CMS on an enterprise that is spread across different locations.

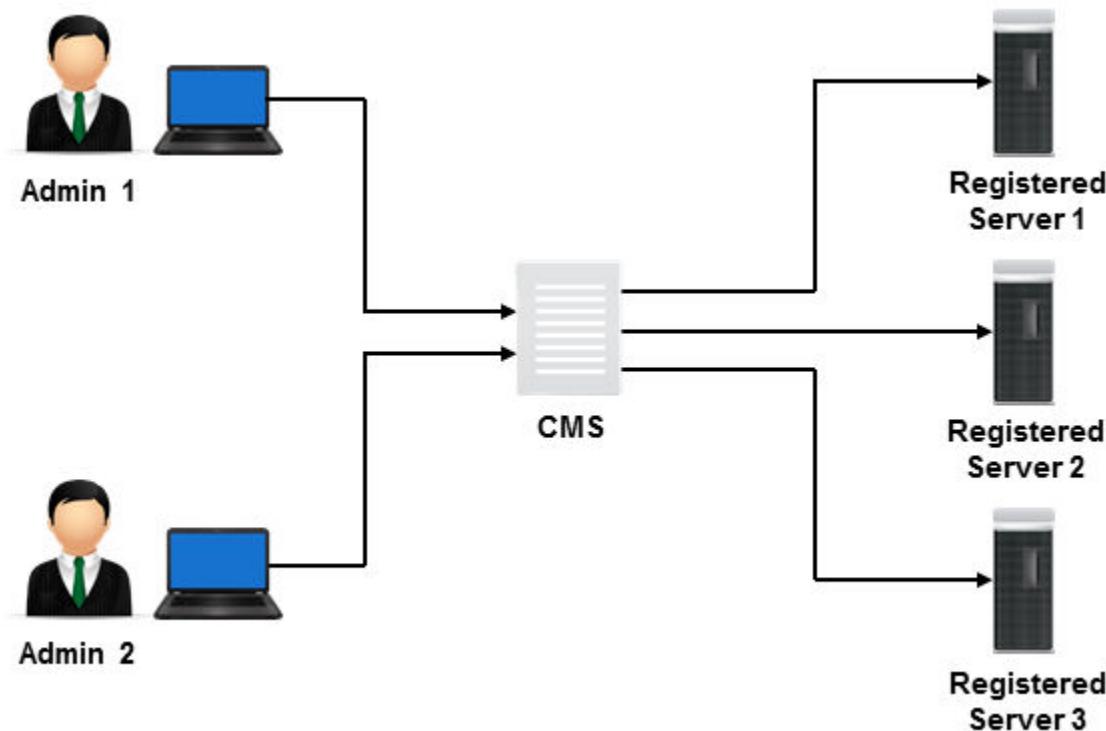


Figure 10–5: Multiple registered servers managed by database administrators using a CMS.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Work with Central Management Servers

ACTIVITY 10-11

Configuring a CMS

Before You Begin

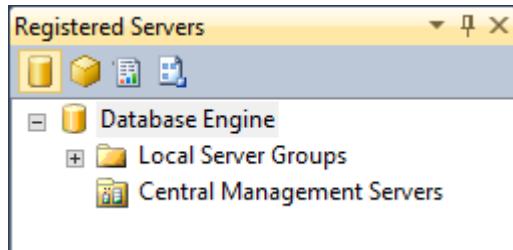
Ensure that **AdventureWorks2012** database is selected.

Scenario

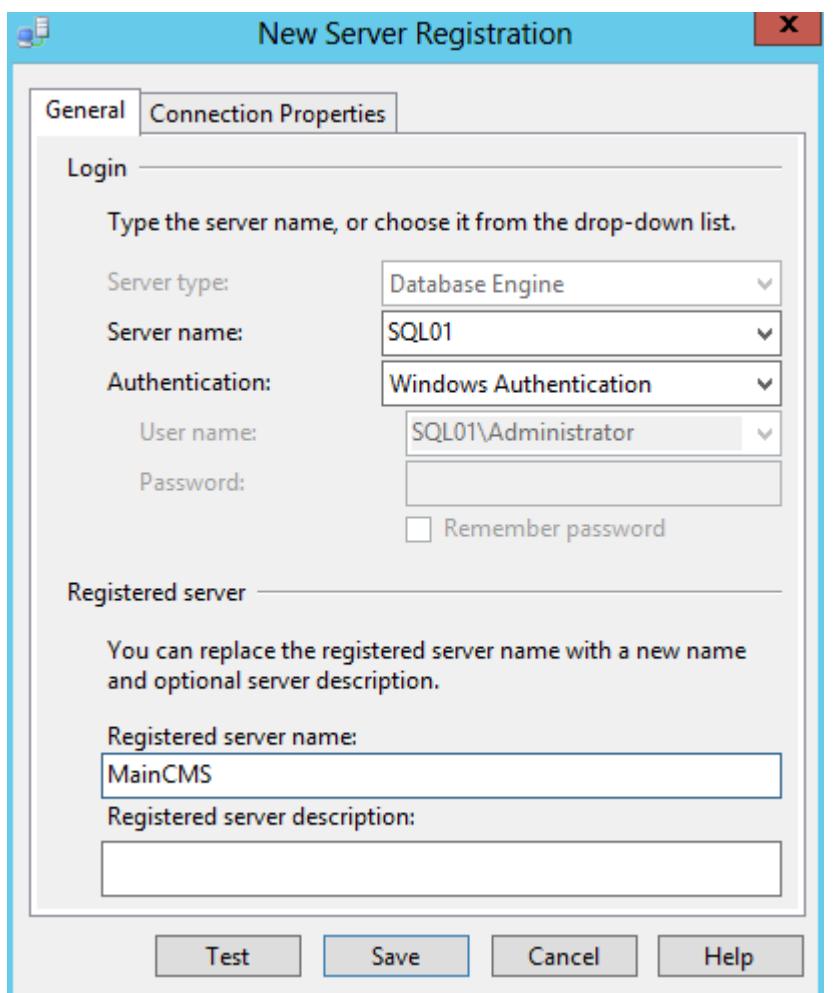
As Adventure Works Cycles continues to grow, the number of servers that are required to handle data and information has also increased. To efficiently manage all the servers, your database manager has asked you to implement a CMS.

1. Register a CMS

- a) In SSMS, select **View→Registered Servers**.
- b) In the Registered Servers pane, expand **Database Engine**.

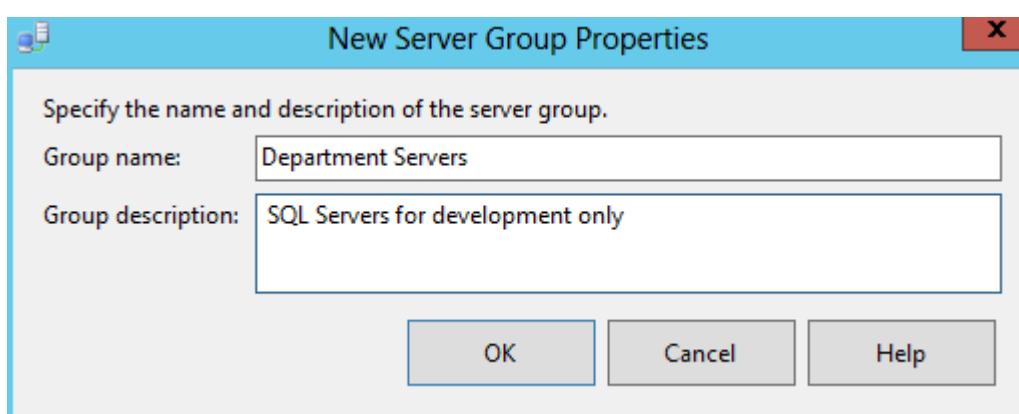


- c) Right-click **Central Management Servers** and select **Register Central Management Server**.
- d) In the **New Server Registration** dialog box, from the **Server name** drop-down list, select **SQL01**.
- e) In the **Authentication** drop-down list, verify that **Windows Authentication** is selected.
- f) In the **Registered server name** text box, type **MainCMS** and select **Test**.



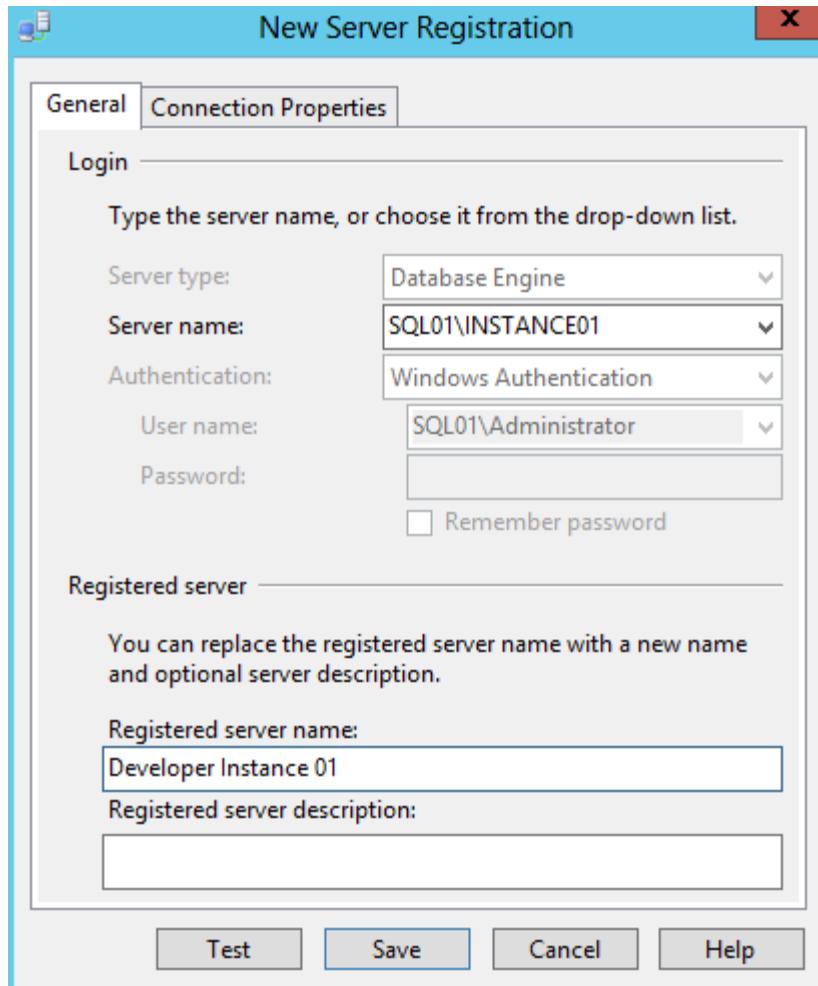
- g) In the **New Server Registration** message box, observe that the CMS has been tested successfully and select **OK**.
- h) In the **New Server Registration** dialog box, select **Save**.
2. Add servers to be managed by the CMS.

- In the **Registered Servers** pane, expand **Central Management Servers**.
- Right-click **MainCMS** and select **New Server Group**.
- In the **New Server Group** dialog box, in the **Group Name** text box, type ***Development Servers***.
- In the **Group description** text box, type ***SQL Servers for development only*** and select **OK**.

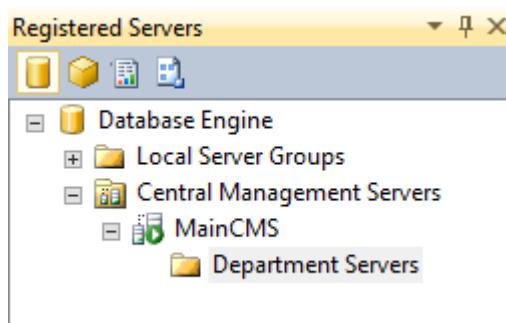


Check with participants to ensure that they have registered the CMS correctly. Ask them to verify that the newly registered CMS is appearing in the **Registered Servers** pane.

- e) Observe that the new server group is added to the **MainCMS** node.
- f) Right click the **Development Servers** server group and select **New Server Registration**.
- g) In the **New Server Registration** dialog box, from the **Server name** drop-down list, select **SQL01\Instance01**.
- h) In the **Registered server name** text box, type **Developer Instance 01** and select **Test**.



- i) In the **New Server Registration** message box, observe that the server has been registered successfully and select **OK**.
- j) In the **New Server Registration** dialog box, select **Save**.



Summary

In this lesson, you implemented different techniques that ensure high availability. You can now ensure that the databases in your enterprise are always available and services to clients are not affected even when there is a failure.

What are the advantages of using an AlwaysOn Availability Group over database mirroring?

A: Answers will vary but may include: ensures availability of all the databases on a server instance at a time, provides the flexibility of performing automatic and manual failovers, and provides options to synchronize the databases automatically.



Use the review questions provided to generate discussion among the participants.

What are the advantages of using a CMS in an enterprise network?

A: Answers will vary but may include: enables administrators to centrally manage multiple servers, maintains a list of servers in the network, controls access to the registered servers, and detects server failures quickly.



Note: Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

11

Optimizing Server Performance

Lesson Time: 2 hours, 45 minutes

Lesson Objectives

In this lesson, you will:

- Manage server performance.
- Optimize indexes.

Lesson Introduction

You have implemented different techniques that ensure high availability of the servers. As users continue to work on SQL Server®, over a period of time, if the server is not properly managed, the performance of the server tends to reduce gradually. The database administrator has to ensure that the server performance does not affect the working of client systems and in turn day-to-day operations. One of the things that fall under the purview of SQL Server database administrators and Server administrators is to optimize the way both the server as well as individual databases perform. This will ensure that the database performs at the expected level and the response time is minimal. In this lesson, you will optimize server performance using suitable tools and methods.

TOPIC A

Manage Server Performance

If the database server is not performing at the optimum level, it will not be able to handle the request of the required number of clients at a given time. The nature of rejected requests can affect the organization's business. As database administrator, you need to constantly monitor the activity on the server and in case of potential issues, be prepared with alternate plans to ensure that the database server downtime does not impact the business. In addition, you need to optimize the server performance. The first and foremost step to increase the performance of a database system is doing so at the server level. In this topic, you will manage the server performance.

Benchmarks

A *benchmark* refers to the standard tests that are run on systems to study their performance. In SQL Server, the total performance is based on a large number of factors such as CPU, RAM, disk I/O, network, and queries. Each factor has its different configuration options, which can increase or decrease the performance by either working in combination with other factors or working against other factors, respectively. For instance, even if you have an extremely fast CPU with multiple cores and lots of RAM, if your disk is slow to seek, read, and write, you will see a decrease in system performance.

You can run different types of benchmarks on each factor to see how it performs. You can also run these benchmarks against a combination of factors including when the SQL Server is performing under a baseline load. This will help you monitor all the different factors and identify any bottlenecks that might impact performance.



What are the benchmark tools that you have used in your organization?

What are their key features?



Note: There are many different third party benchmarking products that are available, including some that are specially optimized to benchmark components for SQL Server. Some built-in tools of Windows and SQL Server also allow you to do some basic sets of tests.



The System Performance Monitor

The System Performance Monitor

The System Performance Monitor (*PerfMon*) is one of the built-in tools you can use for benchmarking and monitoring system performance and is part of all Windows installations.

By default, PerfMon displays the current running information regarding different parameters of the system. At the minimum, you should select these parameters for monitoring when benchmarking SQL Server.

Object	Counter
Processor	% Processor Time
Physical Disk	Disk Transfers/sec
Memory	Pages/sec
Network Interface	Bytes total/sec
SQLServer:Access Methods	Full Scans/sec
SQLServer:Buffer Manager	Buffer Cache Hit Ratio
SQLServer:Databases Application Database	Transactions/sec
SQLServer:General Statistics	User Connections

Object	Counter
SQLServer:Latches	Average Latch Wait Time
SQLServer:Locks	Average Wait Time
SQLServer:Locks	Lock Timeouts/sec
SQLServer:Locks	Number of Deadlocks/sec
SQLServer:Memory Manager	Memory Grants Pending

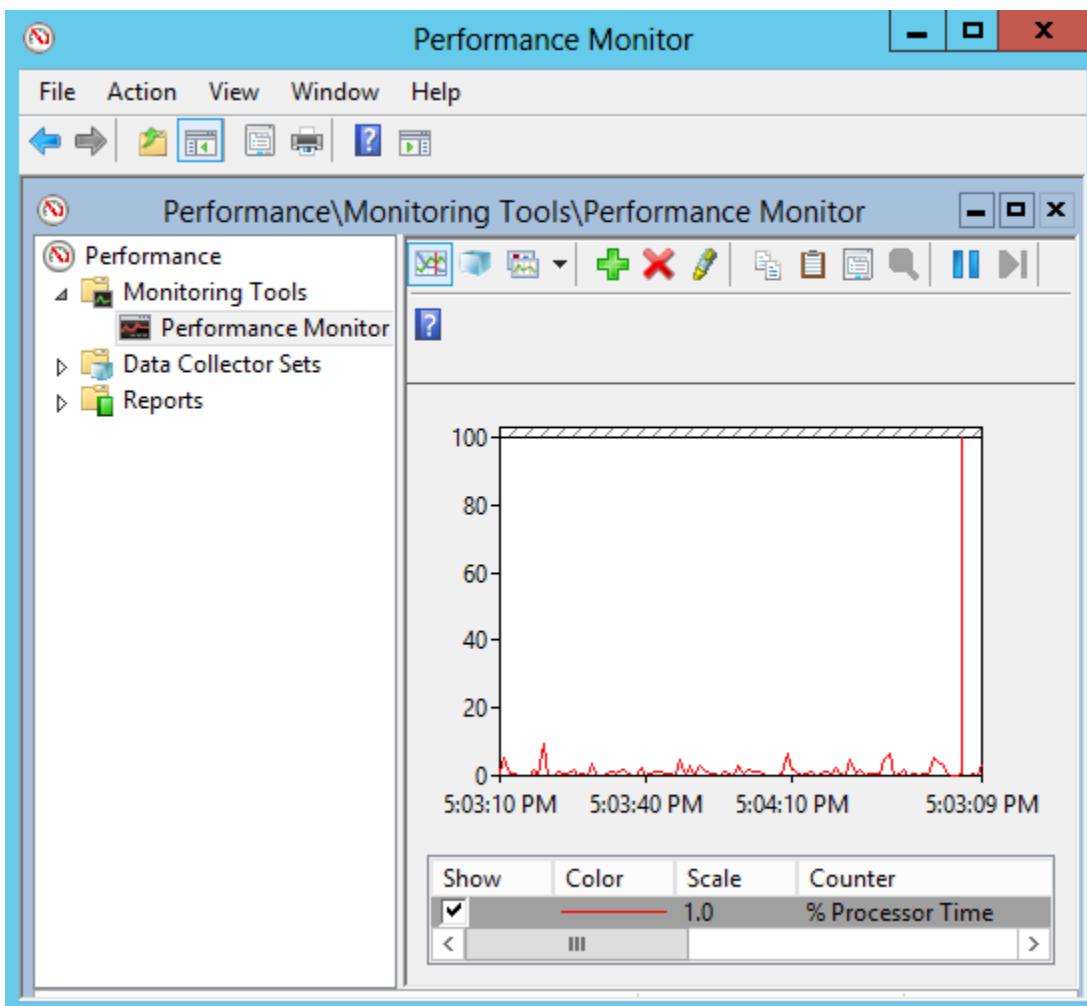


Figure 11-1: The default output of the Performance Monitor.



Note: You will want to add instance specific parameters also in case you wish to monitor the performance of a particular instance on the system as well. For example, you can add generic system parameters such as browser, cache, and database for monitoring. You can also add SQL Server specific parameters, such as SQLServer SIS Service 11.0, for monitoring.



What are the parameters that you usually monitor using PerfMon in your organization? Why?

The Resource Governor

The *Resource Governor* is a feature of SQL Server that allows you to manage the workload of SQL Server by specifying limits on CPU and memory that client requests can take up within a resource pool. The Resource Governor uses a concept of *workloads*, which are uniform sets of queries to limit



The Resource Governor

the amount of CPU and memory allocated. You can define more than one type of workload with different limits.

The workloads assigned to a resource pool with the query sets will get the guaranteed minimum CPU and memory allocated and will not exceed the maximum for both. SQL Server defines two pre-built Resource Pools—default and internal. You can define more pools as required with your own memory and CPU limits in percentages. Once a Resource Pool is created, you can define workload groups under it.

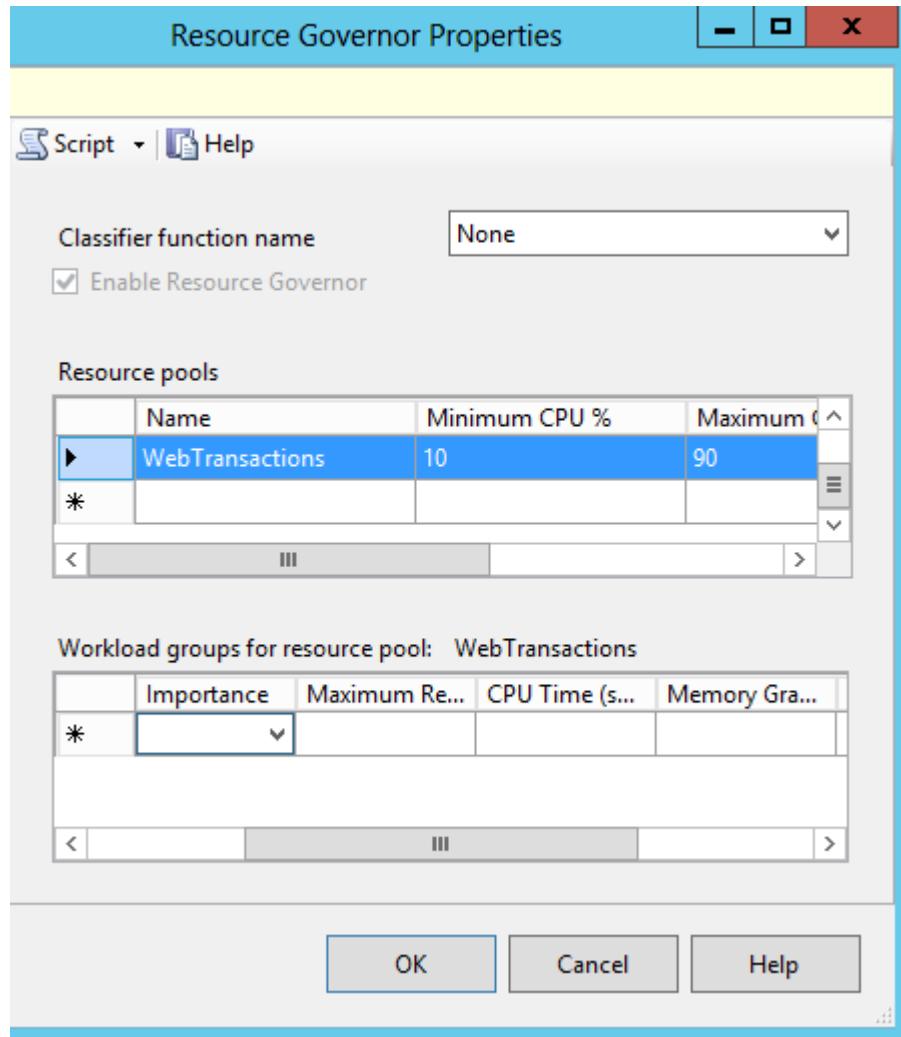


Figure 11-2: The Resource Governor Properties window with a new resource pool and a workload group defined.

DMVs



DMVs

Dynamic Management Views (DMVs) are special system views that allow you to query the system state of the server. You can query a DMV just like any other table or function. DMVs allow you to monitor different parameters of the server, check health, diagnose issues, and tune performance.

There are a number of different DMVs that are available in SQL Server. They are usually classified under logical group names that define their functionality. For instance, there are groups such as Database Related, Full-Text Search Related, Index Related, I/O Related, and Replication Related. Each group has a number of DMVs within them.

For example, the syntax to query and check the physical system information available to SQL Server using the “sys.dm_os_sys_info” is as follows:

```
SELECT cpu_count, physical_memory_kb/1024 AS physical_memory_mb
FROM sys.dm_os_sys_info;
```

This query will return the number of CPUs and physical memory in MB for the SQL Server instance.

A SELECT statement displaying the number of CPUs and physical memory in MB for the SQL Server instance.

	cpu_count	physical_memory_mb
1	2	2047

Figure 11-3: A sample DMV output.

	Note: You can use different DMVs to know about your system, its parameters, SQL Server, and its databases.
	Note: The SQL Server online help, "The Books Online of SQL Server 2012," contains the complete list of DMVs with their usage.



Which DMVs do you use on SQL Server in your organization? Why?

Factors That Affect Server Performance

There are many different factors that affect performance of SQL Server, both external and internal. The external factors that affect SQL Server are the RAM, CPU, disk, and network components. The internal factors are the design of the database, queries, indexes, and configuration settings of all of these factors. There are tools that allow you to monitor the internal factors when designing and running queries.

The SQL Profiler

The *SQL Profiler* is a tool available as part of the SQL Server installation that allows you to create and manage traces. These traces capture events that are occurring on the SQL Server Database Engine for an instance selected and store that in a file or table. This trace allows you to debug any issues.

For instance, if you perform a trace on a production SQL Server and notice that a certain stored procedure takes too long to run, which effectively blocks other transactions in the queue, you can optimize the procedure and ensure that the performance is increased.

These saved traces can also be passed on to the Database Engine Tuning Advisor. This allows the system to figure out any performance related issues and advise on optimizing it. The profile logs information related to events such as logins, all T-SQL statements, stored procedures, and



The SQL Profiler

permission checks. It also stores the start and end time for each event, which allows you to check the time required by the server to complete it.

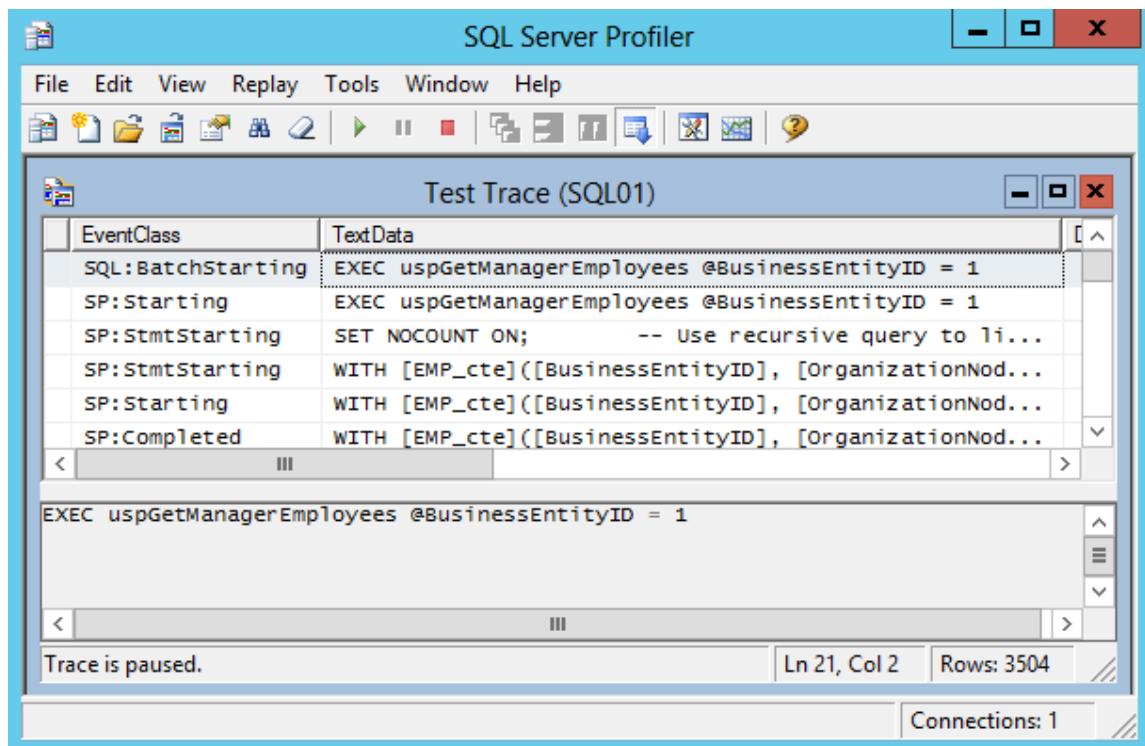


Figure 11-4: The SQL Server Profiler window with the SQL:BatchStarting EventClass selected.

Bottlenecks

Bottlenecks refer to factors, both external and internal, which can affect system performance. Similar to all other parameters, the external bottlenecks are based on the infrastructure of the system such as the CPU, RAM, hard disks, and network. These types of bottlenecks are usually easier to identify and rectify using tools and upgrading the components.

The internal bottlenecks are related to the design of the server instances, databases, indexes, and queries. These types of bottlenecks are more difficult to identify and require a good amount of monitoring as well as a deep understanding of the requirements and architecture of the database system to rectify. Performance issues could come up due to factors such as improper or missing indexes, using cursors, poorly constructed joins between tables, client-level application code, and badly set statistics.



What are the bottlenecks that you encountered on your SQL Server? How did you overcome them?



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Manage Server Performance

ACTIVITY 11-1

Monitoring Performance Using a DMV and SQL Profiler

Before You Begin

In SSMS, ensure that you are connected to the SQL01 instance.

Scenario

As database administrator in Adventure Works Cycles, you have received complaints from some departments that the server transactions such as queries and stored procedures are getting slower. You wish to quickly check the performance information for the current instance across databases as well as for the current instance to monitor what is happening at the server instance level. You decide to use T-SQL statements to query a DMV for performance parameters to view these. Finally, you want to check the performance of a stored procedure to see if it functions quickly. You decide to use the SQL Profiler for this purpose.

1. Monitor the performance of the server with a DMV query.

- Open a **Query Editor** pane, and enter the SELECT statement to select the object_name, counter_name, instance_name, cntr_value columns from the sys.dm_os_performance_counters DMV for your query to return a record for each of the performance counters.



Note: You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```
SELECT object_name,counter_name,instance_name,cntr_value
FROM sys.dm_os_performance_counters
b) Enter the WHERE clause to filter the counter name to list records that contain only the values "buffer cache hit ratio," "user connections," "transactions," "average wait time (ms)," "transactions/sec," "errors/sec," "target server memory (kb)," and "total server memory (kb)."
```

```
SELECT object_name,counter_name,instance_name,cntr_value
FROM sys.dm_os_performance_counters
WHERE counter_name IN ('buffer cache hit ratio', 'user connections',
'transactions', 'average wait time (ms)', 'transactions/sec', 'errors/
sec', 'target server memory (kb)', 'total server memory (kb)')
```

- Enter the ORDER BY clause to sort the query results on the object_name and counter name columns.

```
"...
WHERE counter_name IN ('buffer cache hit ratio', 'user connections',
'transactions', 'average wait time (ms)', 'transactions/sec', 'errors/
sec', 'target server memory (kb)', 'total server memory (kb)')
ORDER BY object_name, counter_name;
```

- Execute the query to view performance information.
- On the **Results** tab, observe that the different values are given for each parameter including for the server instance as well as for different databases on the instance.

```

SQLQuery5.sql - SQL01.master (SQL01\Administrator (60))*  

SELECT object_name,counter_name,instance_name,cntr_value  

FROM sys.dm_os_performance_counters  

WHERE counter_name IN ('buffer cache hit ratio', 'user connections',  

'transactions','average wait time (ms)', 'transactions/sec',  

'errors/sec','target server memory (kb)', 'total server memory (kb)')  

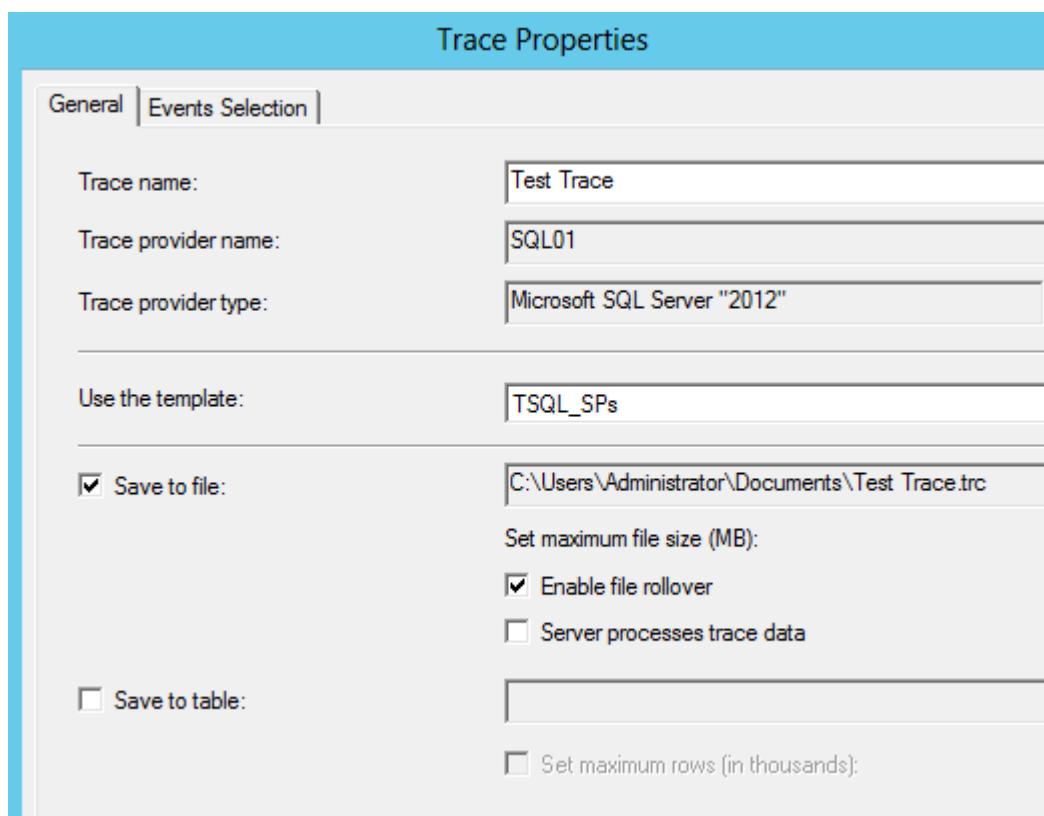
ORDER BY object_name, counter_name;

```

	object_name	counter_name	instance_name	cntr_value
1	SQLServer:Buffer Manager	Buffer cache hit ratio		112
2	SQLServer:Databases	Transactions/sec	SalesDB	290
3	SQLServer:Databases	Transactions/sec	tempdb	5035
4	SQLServer:Databases	Transactions/sec	model	5
5	SQLServer:Databases	Transactions/sec	CharityEventsDB	5
6	SQLServer:Databases	Transactions/sec	msdb	1456
7	SQLServer:Databases	Transactions/sec	ReportServerTempDB	9

Query executed... | SQL01 (11.0 SP1) | SQL01\Administrator (60) | master | 00:00:00 | 35 rows

- f) Close the **Query Editor** pane without saving the query.
2. In the SQL Profiler, create and start a new trace named Test Trace.
- From the Start screen, select **SQL Server Profiler**.
 - On the standard toolbar, select the **New Trace** button.
 - In the **Connect to Server** dialog box, ensure that **SQL01** instance is the server name and **Windows Authentication** is the authentication type and select **Connect**.
 - In the **Trace Properties** dialog box, in the **Trace name** text box, type **Test Trace** as the trace name.
 - From the **Use the template** drop-down list, select **TSQL_SPs** to select the template for stored procedures.
 - Check the **Save to file** check box.
 - In the **Save As** dialog box, in the **File name** text box, observe that **Test Trace** is the file name, and select **Save** to save the file in the **Documents** folder.
 - In the **Trace Properties** dialog box, select the **Run** button to start the trace.



3. Execute a stored procedure named `uspGetManagerEmployees`.
 - a) Switch back to the SSMS window.
 - b) In the **Object Explorer** pane, expand **SQL01→Databases** and select **AdventureWorks2012** database.
 - c) Open a **Query Editor** pane and enter the `EXEC` statement to run the stored procedure `uspGetManagerEmployees` with the name as `@BusinessEntityID` variable and value as one.


```
EXEC uspGetManagerEmployees @BusinessEntityID = 1
```
 - d) Execute the query.
 - e) Observe that the names of each manager are displayed for each employee.

SQLQuery5.sql - SQL01.AdventureWorks2012 (SQL01\Administrator (60))*

```
EXEC uspGetManagerEmployees @BusinessEntityID = 1
```

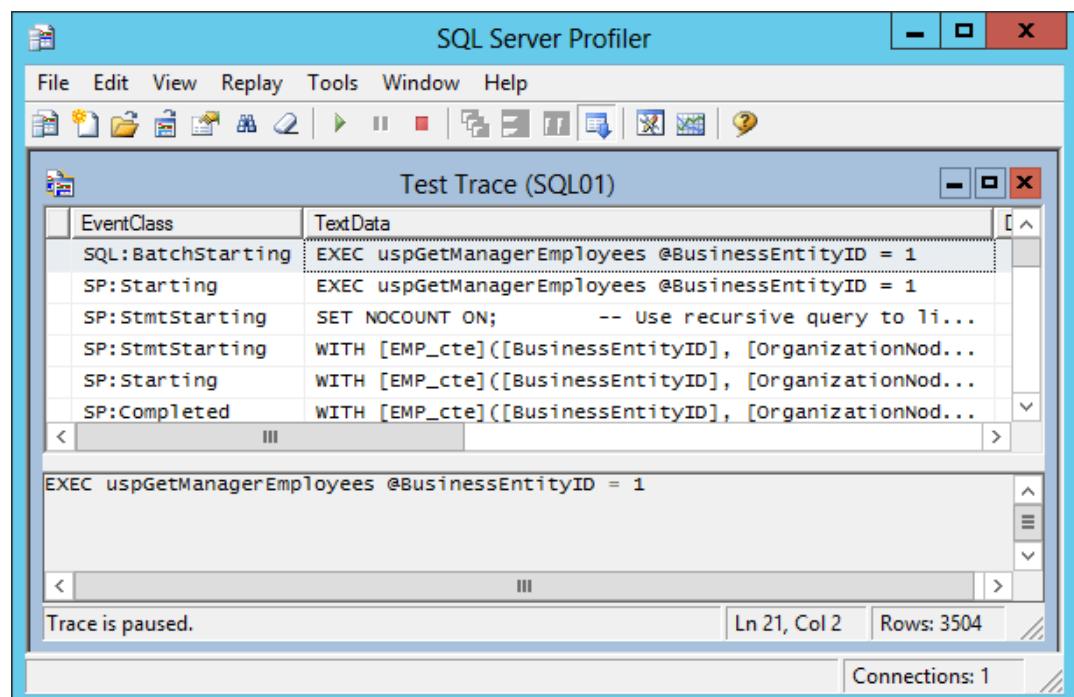
100 % < > III

	RecursionLevel	OrganizationNode	ManagerFirstName	ManagerLastName	BusinessEntityID
1	1	/1/	Ken	Sánchez	2
2	1	/2/	Ken	Sánchez	16
3	1	/3/	Ken	Sánchez	25
4	1	/4/	Ken	Sánchez	234
5	1	/5/	Ken	Sánchez	263
6	1	/6/	Ken	Sánchez	273

< > III

Q... | SQL01 (11.0 SP1) | SQL01\Administrator (60) | AdventureWorks2012 | 00:00:00 | 289 rows

4. Pause the tracing action and locate the beginning and end of the store procedure.
 - a) Switch back to the SQL Profiler window.
 - b) In the standard toolbar, select the **Pause Selected Trace** button  to pause the trace.
 - c) Select the last row in the trace output and press **Ctrl+Home** to navigate to the top of the trace output.
 - d) Press **Ctrl+F** to launch the search tool.
 - e) In the **Find** dialog box, in the **Find what** text box, type *uspGetManagerEmployees* as the search string and select **Find Next**.
 - f) Repeat the **Find Next** until the **SQL:BatchStarting** row, which indicates the beginning of the stored procedure, is selected. Close the **Find** dialog box.



- g) Scroll down the rows until you reach the **SP:Completed** event for the stored procedure execution.
 - h) Observe that there is a potential bottleneck because the stored procedure is actually performing a row-by-row check for each employee in the database, which can affect the query performance.
 - i) Select the **Close** button to close the SQL Profiler window.
 - j) In the **SQL Server Profiler** dialog box, select **Yes** to close it.
 - k) Close the **Query Editor** pane without saving the queries.
-

ACTIVITY 11-2

Identifying IO, Memory, and CPU Bottlenecks

Before You Begin

In the **Object Explorer** pane, select the SQL01 server instance.

Scenario

As part of the database server optimizing initiatives, you are asked to check your SQL Server to identify if there are any bottlenecks impacting performance in the context of I/O, memory, or CPU. For this, you decide to create a baseline set of measurements when the system is running fine under normal load, which can then be compared when any issue pops up later.

1. Create a base query named baseline1 to check the I/O performance.

- a) Open a **Query Editor** pane and enter the SELECT statement to select all the records from the sys.dm_os_wait_stats DMV, which will display the waiting time for each of the executed threads.

```
SELECT * FROM sys.dm_os_wait_stats
```

- b) Enter the WHERE clause to filter the records when the pattern of the values of the wait_type column starts with PAGEIOLATCH

```
SELECT * FROM sys.dm_os_wait_stats  
WHERE wait_type LIKE 'PAGEIOLATCH%'
```

- c) Enter the ORDER BY clause to sort the query output in ascending order of the wait_type column.

```
"...  
WHERE wait_type LIKE 'PAGEIOLATCH%'  
ORDER BY wait_type ASC;
```

- d) Execute the DMV query.

- e) On the Results tab, observe that the query displays the count of tasks waiting and the number of milliseconds each task has to wait to get an I/O connection as the query result.

```
SQLQuery6.sql - SQL01.AdventureWorks2012 (SQL01\Administrator (64))*
```

```
SELECT * FROM sys.dm_os_wait_stats  
WHERE wait_type LIKE 'PAGEIOLATCH%'  
ORDER BY wait_type ASC;
```

wait_type	waiting_tasks_count	wait_time_ms	max_wait_time_ms
PAGEIOLATCH_DT	0	0	0
PAGEIOLATCH_EX	48	5631	1557
PAGEIOLATCH_KP	0	0	0
PAGEIOLATCH_NL	0	0	0
PAGEIOLATCH_SH	1199	43835	1726
PAGEIOLATCH_UP	251	385	105

L01 (11.0 SP1) | SQL01\Administrator (64) | AdventureWorks2012 | 00:00:00 | 6 rows

- f) Select the **Results** tab and select **File→Save Results As** to display the **Save Grid Results** dialog box.
- g) Navigate to the C:\093108Data folder and in the **File name** text box, type **baseline1.csv** and select **Save** to save the query results as a baseline query.
- h) Clear the **Query Editor** pane.

2. Create a query named baseline2 to display the server statistics.

- a) In the **Query Editor** pane, enter the SELECT statement followed by the sum of the pages_kb column with a column alias "Total Pages."

```
SELECT SUM(pages_kb) AS Total_Pages,
```

- b) Enter the sum of the virtual_memory_reserved_kb column with a column alias "Total_Reserved_VM."

```
SELECT SUM(pages_kb) AS Total_Pages,
SUM(virtual_memory_reserved_kb) AS Total_Reserved_VM,
```

- c) Enter the sum of the virtual_memory_committed_kb column with a column alias "Total_Committed_VM."

```
"...
SUM(virtual_memory_reserved_kb) AS Total_Reserved_VM,
SUM(virtual_memory_committed_kb) AS Total_Committed_VM,
```

- d) Enter the sum of the awe_allocated_kb column with a column alias "Total_AWE."

```
"...
SUM(virtual_memory_committed_kb) AS Total_Committed_VM,
SUM(awe_allocated_kb) AS Total_AWE,
```

- e) Enter the sum of the shared_memory_reserved_kb column with a column alias "Total_Reserve_SM."

```
"...
SUM(awe_allocated_kb) AS Total_AWE,
SUM(shared_memory_reserved_kb) AS Total_Reserve_SM,
```

- f) Enter the sum of the shared_memory_committed_kb column with a column alias "Total_Committed_SM."

```
"...
SUM(shared_memory_reserved_kb) AS Total_Reserve_SM,
SUM(shared_memory_committed_kb) AS Total_Committed_SM
```

- g) Enter the FROM clause to select the sys.dm_os_memory_clerks DMV.

```
"...
SUM(shared_memory_committed_kb) AS Total_Committed_SM
FROM sys.dm_os_memory_clerks;
```

- h) Execute the query. Observe that the query results display the cumulative memory stats for the server.

	Total_Pages	Total_Reserved_VM	Total_Committed_VM	Total_AWE	Total_Reserve_SM
1	257616	132068	54336	0	96

- i) Select the **Results** tab and select **File→Save Results As** to display the **Save Grid Results** dialog box.
 - j) Navigate to the **C:\093108Data** folder and in the **File name** text box, type **baseline2.csv** and select **Save** to save the query results as a baseline query.
 - k) Clear the **Query Editor** pane.
3. Create a query named baseline3 to view the object-wise statistics for cache memory.
- a) In the **Query Editor** pane, enter the SELECT statement to select the columns cmcc.cache_address, mcc.name, mcc.type, mcc.pages_kb, mcc.pages_in_use_kb, and mcc.entries_count.



Note: You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in **C:\093108Data\Code.txt**.

```
SELECT
mcc.cache_address,
mcc.name,
mcc.type,
mcc.pages_kb,
mcc.pages_in_use_kb,
mcc.entries_count,
```

- b) Add the columns mcc.entries_in_use_count, mcch.removed_all_rounds_count, and mcch.removed_last_round_count from the sys.dm_os_memory_cache_counters DMV with alias "mcc."
- ```
"...
mcc.entries_in_use_count,
mcch.removed_all_rounds_count,
mcch.removed_last_round_count
FROM sys.dm_os_memory_cache_counters mcc
```
- c) ENTER the JOIN statement to join the mcc DMV and sys.dm\_os\_memory\_cache\_clock\_hands with "mcch" as alias.
- ```
"...
FROM sys.dm_os_memory_cache_counters mcc
JOIN sys.dm_os_memory_cache_clock_hands mcch
```
- d) Enter the ON keyword to join on the common column cache_address.
- ```
"...
JOIN sys.dm_os_memory_cache_clock_hands mcch
ON (mcc.cache_address = mcch.cache_address)
```
- e) Execute the query and in the query output view the memory being taken by each object in the server.

SQLQuery3.sql - SQ...Administrator (52)\* X

```

SELECT
 mcc.cache_address,
 mcc.name,
 mcc.type,
 mcc.pages_kb,
 mcc.pages_in_use_kb,
 mcc.entries_count,
 mcc.entries_in_use_count,
 mcch.removed_all_rounds_count,
 mcch.removed_last_round_count
FROM sys.dm_os_memory_cache_counters mcc
JOIN sys.dm_os_memory_cache_clock_hands mcch
ON (mcc.cache_address = mcch.cache_address)

```

100 % < |

|   | cache_address | name                   | type                   | pages_kb | pages_in_use_kb |
|---|---------------|------------------------|------------------------|----------|-----------------|
| 1 | 0x03CED340    | SOS_StackFramesStore   | CACHESTORE_STACKFRAMES | 8        | 8               |
| 2 | 0x03CED340    | SOS_StackFramesStore   | CACHESTORE_STACKFRAMES | 8        | 8               |
| 3 | 0x040A7040    | EventNotificationCache | CACHESTORE_EVENTS      | 16       | 0               |
| 4 | 0x040A7040    | EventNotificationCache | CACHESTORE_EVENTS      | 16       | 0               |
| 5 | 0x040DE068    | Object Plans           | CACHESTORE_OBJCP       | 30384    | 0               |

- Select the **Results** tab and select **File→Save Results As** to display the **Save Grid Results** dialog box.
- Navigate to the C:\093108Data folder and in the **File name** text box, type **baseline3.csv** and select **Save** to save the query results as a baseline query.
- Clear the **Query Editor** pane.

**4.** Create a query named baseline4 to view the CPU statistics of the query plans on the system.

- In the **Query Editor** pane, enter the SELECT statement to select the plan\_handle column along with sum of total\_worker\_time, and sum of execution\_count with aliases “Total\_Time” and “Total\_Exec,” respectively.



**Note:** You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```

SELECT plan_handle,
SUM(total_worker_time) AS Total_Time,
SUM(execution_count) AS Total_Exec,

```

- Enter the COUNT(\*) statement with “Num\_Statements” as alias followed by the FROM clause to select sys.dm\_exec\_query\_stats as the source DMV for the query.

```

SELECT plan_handle,
SUM(total_worker_time) AS Total_Time,
SUM(execution_count) AS Total_Exec,
COUNT(*) AS Num_Statements
FROM sys.dm_exec_query_stats

```

- Enter the GROUP BY statement to group the records on the plan\_handle column.

```

"...
COUNT(*) AS Num_Statements
FROM sys.dm_exec_query_stats
GROUP BY plan_handle

```

- Enter the ORDER BY clause to sort the query output in ascending order of the sum of total\_worker\_time column and in descending order of the sum of the execution\_count column.

```
"...
FROM sys.dm_exec_query_stats
GROUP BY plan_handle
ORDER BY SUM(total_worker_time), SUM(execution_count) DESC;
```

- e) Execute the query and view the CPU statistics as query output.

```
SQLQuery4.sql - SQ...Administrator (53)*
SELECT plan_handle,
SUM(total_worker_time) AS Total_Time,
SUM(execution_count) AS Total_Exec,
COUNT(*) AS Num_Statements
FROM sys.dm_exec_query_stats
GROUP BY plan_handle
ORDER BY SUM(total_worker_time), SUM(execution_count) DESC;
```

| plan_handle                                             | Total_Time | Total_Exec | Num_Statements |
|---------------------------------------------------------|------------|------------|----------------|
| 1 0x0600010046C6502AB02C7218010000000000000000000000... | 85         | 2          | 1              |
| 2 0x060001008EE1BD1660FA5118010000000000000000000000... | 86         | 2          | 1              |
| 3 0x06000100E0145735583D7E12010000000000000000000000... | 88         | 2          | 1              |
| 4 0x0600010043BA0B0180EE5118010000000000000000000000... | 92         | 2          | 1              |
| 5 0x06000A0011434331581F511E010000000000000000000000... | 95         | 1          | 1              |

- f) Select the **Results** tab and select **File→Save Results As** to display the **Save Grid Results** dialog box.  
g) Navigate to the **C:\093108Data** folder and in the **File name** text box, type ***baseline4.csv*** and select **Save** to save the query results as a baseline query.

|  |                                                                                                                                                                                                                                                              |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Note:</b> Now that you have saved the four query results as the baseline queries, in the future, you can run the same queries and compare the query results with the baseline results to determine the factors that are affecting the system performance. |
|  | <b>Note:</b> When comparing the values and noting a query plan that takes more CPU usage than required, you can use the <b>plan_handle</b> column to get the query that is being executed.                                                                   |

- h) Close the **Query Editor** pane without saving the queries.

## ACTIVITY 11–3

### Locating Processes Blocking or Claiming Resources

#### Before You Begin

In the **Object Explorer** pane, ensure that SQL01 is selected.

#### Scenario

Your users have complained that the server performance has degraded suddenly. You realize that running processes are responsible for the sudden drop in system performance. You wish to check the blocking processes within your server and identify a way to highlight and stop them from executing so that resources can be freed.



**Note:** In this activity, you will create a new database, and perform simultaneous actions in it that will not complete—essentially blocking each other. You will then try to identify and stop these actions.

1. Create a database named EmployeeDB and table named Departments with three columns.
  - a) Open a **Query Editor** pane and enter the CREATE DATABASE statement with EmployeeDB as the name of the required database followed by the GO statement to execute a batch of statements.
 

```
CREATE DATABASE EmployeeDB;
GO
```
  - b) Enter the USE statement to select the EmployeeDB database followed by the GO statement to execute a batch of statements.
 

```
"...
GO
USE EmployeeDB;
GO
```
  - c) Enter the CREATE TABLE Statement with Departments as the name of the new table.
 

```
"...
USE EmployeeDB;
GO
CREATE TABLE Departments
```
  - d) Enter the statement to define the DepartmentID as the first column with INT as its data type and an IDENTITY column of (1,1) size. Include the NOT NULL constraint to indicate that null values are not accepted.
 

```
"...
GO
CREATE TABLE Departments
```
  - e) Enter the statement to define the Department name as the second column of the table with the with VARCHAR(50) as its data type. Include the NOT NULL constraint to indicate that null values are not accepted.
 

```
"...
CREATE TABLE Departments
```
  - f) Enter the statement to define the Description as the third column of the table with VARCHAR(150) as its data type. Include the NULL constraint to indicate that null values are accepted.
 

```
"...
DepartmentName VARCHAR(50) NOT NULL,
```

```

 Description VARCHAR(150) NULL
);

```

- g) Execute the query.

2. Insert three new records in the Departments table.

- a) Open a new **Query Editor** pane.
- b) Enter the USE statement to select the EmployeeDB database as the current database followed by the GO statement to execute a batch of statements.

```

USE EmployeeDB;
GO

```

- c) Enter the BEGIN TRAN statement to indicate the beginning of the transaction.

```

"...
GO
BEGIN TRAN

```

- d) Enter the INSERT INTO statement to insert the following values for the **DepartmentName** and **Description** columns in the **Departments** table:

- HR, Human Resource
- IT, Information Technology
- Dev, Product Development

```

"...
BEGIN TRAN
INSERT INTO Departments VALUES
('HR', 'Human Resources'),
('IT', 'Information Technology'),
('Dev', 'Product Development');

```

- e) Execute the query to insert the three records into the Departments table.

3. Query the Departments table and view the execution status of the query.

- a) Open a new **Query Editor** pane.
- b) Enter the USE statement to switch to the EmployeeDB database and follow it with the GO statement to execute a batch of statements.

```

USE EmployeeDB;
GO

```

- c) Enter the SELECT statement to select all records from the Departments table.

```

"...
GO
SELECT * FROM Departments;

```

- d) Switch to the previous **Query Editor** pane, where you inserted records into the **Departments** table. Execute the query.

- e) Switch back to the **Query Editor** pane, where data is being retrieved.

- f) Execute the query and observe that the status bar displays the message “Executing query...” indefinitely, which indicates that there is a blocking transaction.

The screenshot shows the SSMS Query Editor with two tabs: 'SQLQuery9.sql - SQL...r (66) Executing...' and 'SQLQuery8.sql - SQ...Administrator (64)\*'. The current tab contains the following T-SQL code:

```
USE EmployeeDB;
GO
SELECT * FROM Departments;
```

The results pane is empty, showing '0 rows'. The status bar at the bottom indicates: 'Executing query... | SQL01 (11.0 SP1) | SQL01\Administrator (66) | EmployeeDB | 00:01:46 | 0 rows'.

4. Generate the Standard Report to view the SessionID of the Blocking Transaction.
  - a) On the **Object Explorer** pane, right-click **SQL01** and select **Reports→Standard Reports→Activity – All Blocking Transactions**.
  - b) After the report is loaded, in the **Transaction ID** column, select the **+** button in the report to view the blocking ID and the blocked ID.

The screenshot shows the 'Activity - All Blocking Transactions' report generated on SQL01 at 3/15/2013 5:58:35 PM. The report title is 'Activity - All Blocking Transactions' and it includes the Microsoft SQL Server logo.

This report provides information about each transaction on the instance which is blocking one or more other transactions.

**All Blocking Transactions**

The description of transactions which are blocking other transactions.

| Transaction ID | # Directly Blocked Transactions | # Indirectly Blocked Transactions | Transaction Name         | State  | Transaction Type | Start Time           | Resource Type | Session ID |
|----------------|---------------------------------|-----------------------------------|--------------------------|--------|------------------|----------------------|---------------|------------|
| 515182*        | 1                               | 0                                 | user_transaction         | Active | Full Transaction | 3/15/2013 5:52:32 PM | RID           | 64         |
|                | Direct/Indirect                 | Blocked Transaction ID            | Blocked Transaction Name | State  | Transaction Type | Start Time           | Resource Type | Session ID |
|                | Direct                          | 516248*+                          | SELECT                   | Active | Read Only        | 3/15/2013 5:55:29 PM | RID           | 66         |

\* The transactions span multiple databases on the same instance.

- c) Scroll right to view and note the Session ID of the transaction that is responsible for the block.

|  |                                                                                                                                                                                                  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Caution:</b> Note down the session ID for using it later with the KILL command. The correct session ID is required for you to avoid killing the wrong session and affecting the system state. |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

5. Create another query to double-check the session id of the blocking transactions.
  - a) Open a **Query Editor** pane.
  - b) Enter the SELECT statement to select the **blocking\_session\_id** column from the **sys.dm\_exec\_requests** DMV.

```
SELECT blocking_session_id FROM sys.dm_exec_requests
```

  - c) Enter a WHERE clause to filter the output on the criteria that the **blocking\_session\_id** is not equal to 0 and to confirm that the session id of the blocking transaction was correctly identified.

```
SELECT blocking_session_id FROM sys.dm_exec_requests
WHERE blocking_session_id<> 0;
```

  - d) Execute the query to view the session id of the blocking transaction.

```
SQLQuery11.sql - S...Administrator (67)*
SELECT blocking_session_id FROM sys.dm_exec_requests
WHERE blocking_session_id <> 0;

100 % < >
Results Messages
blocking_session_id
1 64
Q | SQL01 (11.0 SP1) | SQL01\Administrator (67) | EmployeeDB | 00:00:00 | 1 rows
```

- e) View and verify that the same ID displayed earlier is in the query result.
6. Terminate the blocking process and complete the execution of the earlier blocked query.
- In the **Query Editor** pane enter the KILL statement with the blocking process ID to terminate the process.  
KILL 64
  - Execute the query.
  - Once the command is successful, run the previous query to retrieve the blocking\_session\_id again. Check that it does not return a value now.

```
SQLQuery4.sql - SQL01.EmployeeDB (SQL01\Administrator (64))*
SELECT blocking_session_id
FROM sys.dm_exec_requests
WHERE blocking_session_id <> 0;

100 % < >
Results Messages
blocking_session_id
Q | SQL01 (11.0 SP1) | SQL01\Administrator (64) | EmployeeDB | 00:00:00 | 0 rows
```

- Switch to the **Query Editor** pane that was trying to retrieve the values from the table.
- Observe that query has now completed successfully and also that because the transaction was terminated, the values were rolled back automatically and therefore there are no rows to display.
- Close the **Query Editor** pane without saving the queries.
- Close the **Activity – All Blocking Transactions** pane.

## ACTIVITY 11–4

### Managing Resource Utilization Using Resource Governor

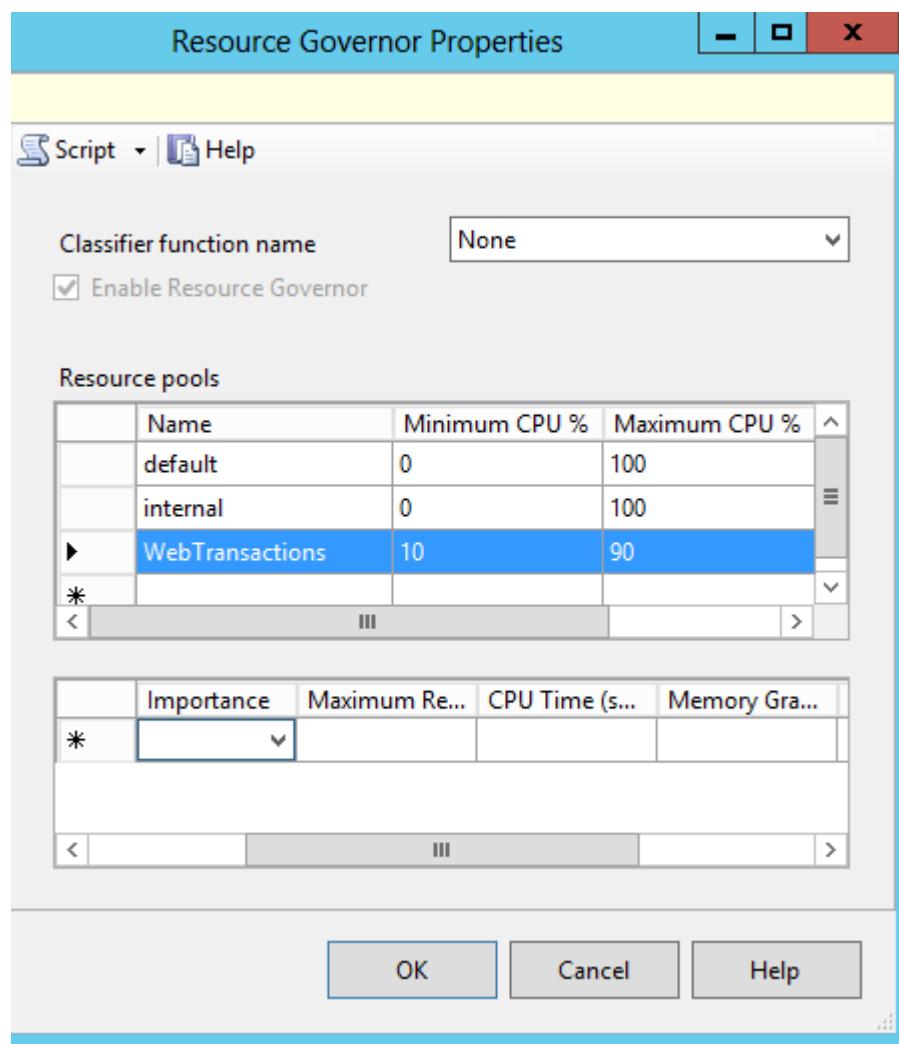
#### Scenario

You have noticed that the server performance is not uniform during an entire work day but tends to follow a fixed pattern of changes based on specific times. To ensure that this does not affect the day-to-day transactions, you decide to manage the resources that are utilized by the system during certain loads by using Resource Governor.

For the new resource pool, use the following values:

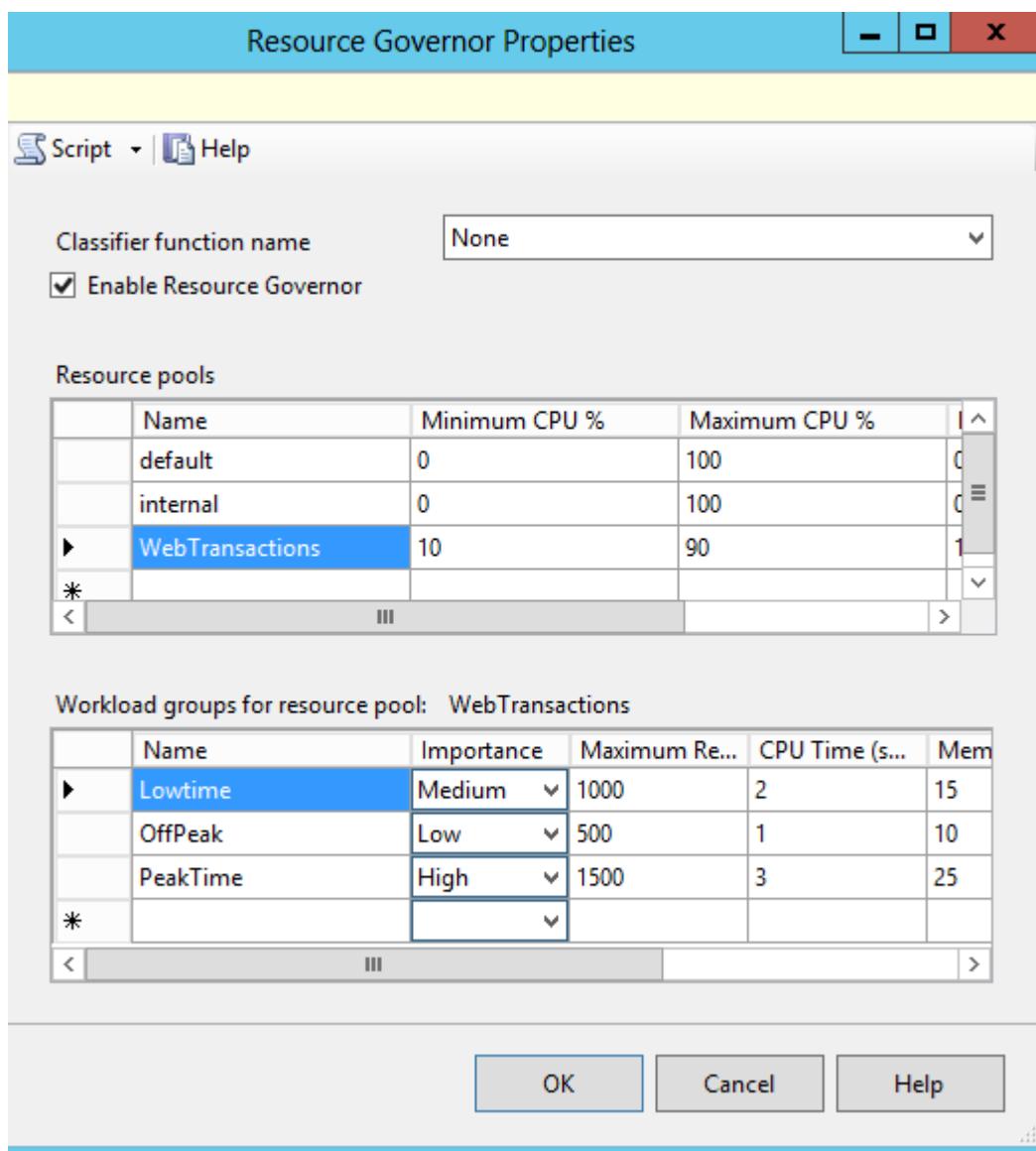
- Name: WebTransactions
- Minimum CPU %: 10%
- Maximum CPU %: 90%
- Minimum Memory %: 10%
- Maximum Memory %: 90%

- 
1. Add a new resource pool named WebTransactions with minimum and maximum values as 10 and 90 for both CPU and Memory.
    - a) In the **Object Explorer** pane, expand **SQL01–Management→Resource Governor**.
    - b) Right-click **Resource Governor** node and select **Enable**.
    - c) Right-click **Resource Pools** and select **New Resource Pool**.
    - d) In the Resource Governor Properties window, in the **Resource pools** section, in the **Name** column, in the blank row below the “default” and “internal” pools, enter the values as given in the scenario and milestone text.



2. Specify values for the new WebTransactions pool.
  - a) In the **Resource pools** section, select the **WebTransactions** pool.
  - b) In the Workload groups pane, enter the values as indicated in the table.

| Name    | Importance | Max Requests | CPU Time | Memory Grant | Grant Time-out | Deg. Of Parallelism |
|---------|------------|--------------|----------|--------------|----------------|---------------------|
| Peaktme | High       | 1500         | 3        | 25           | 0              | 1                   |
| LowTime | Medium     | 1000         | 2        | 15           | 0              | 0                   |
| OffPeak | Low        | 500          | 1        | 10           | 0              | 0                   |



- c) Select **OK** to save the pool.
- d) Observe that you have created three different workload groups.



**Note:** However, there is no way for the governor to target any SQL Server activity into these groups. For this, you need to create a classifier function.

3. Create a classifier function named RG\_Peak\_Time\_Calculator() in the master database that checks the current time and returns the workload group that should be used.
  - a) Open a **Query Editor** pane and ensure that the master database is selected.
  - b) Enter the CREATE FUNCTION statement to create a function named RG\_Peak\_Time\_Calculator().
 

```
CREATE FUNCTION RG_Peak_Time_Calculator()
```

- c) Enter SYSNAME as the return type WITH SCHEMABINDING enabled followed by the AS keyword to complete the definition part of the function.



**Note:** You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```
CREATE FUNCTION RG_Peak_Time_Calculator()
RETURNS SYSNAME
WITH SCHEMABINDING
AS
```

- d) Enter the BEGIN and ENTER container. In the container, enter the DECLARE statement to declare variables CurrentTime and WL\_Group of TIME and SYSNAME data types, respectively.

```
"...
RETURNS SYSNAME
WITH SCHEMABINDING
AS
BEGIN
DECLARE @CurrentTime TIME
DECLARE @WL_Group SYSNAME
END
```

- e) Enter the SET statement to initialize the CurrentTime variable with CONVERT(time, GETDATE()) keywords to extract and convert only the time part from the system date.

```
"...
BEGIN
DECLARE @CurrentTime TIME
DECLARE @WL_Group SYSNAME
SET @CurrentTime = CONVERT(time, GETDATE())
END
```

- f) Enter the IF statement to check whether the CurrentTime value is between 9:00 A.M. and 5:00 P.M.

```
"...
SET @CurrentTime = CONVERT(time, GETDATE())
IF (@CurrentTime >= '9:00 AM' AND @CurrentTime <= '5:00 PM')
END
```

- g) Enter the BEGIN END container and in the container, enter the SET statement to set the "Peaktme" as the value for the WL\_Group variable.

```
"...
IF (@CurrentTime >= '9:00 AM' AND @CurrentTime <= '5:00 PM')
BEGIN
SET @WL_Group = 'Peaktme'
END
END
```

- h) Enter the IF statement to check whether the CurrentTime value is between 5:01 P.M. and 11:59 P.M.

```
"...
BEGIN
SET @WL_Group = 'Peaktme'
END
IF (@CurrentTime >= '5:01 PM' AND @CurrentTime <= '11:59 PM')
END
```

- i) Enter the BEGIN END container and in the container, enter the SET statement to set the "Lowtime" as the value for the WL\_Group variable.

```
"...
IF (@CurrentTime >= '5:01 PM' AND @CurrentTime <= '11:59 PM')
BEGIN
SET @WL_Group = 'Lowtime'
END
END
```

- j) Enter the IF statement to check whether the CurrentTime value is between 12:00 A.M. and 8:59 A.M.

```
"...
BEGIN
```

```

SET @WL_Group = 'Lowtime'
END
IF (@CurrentTime >= '12:00 AM' AND @CurrentTime <= '8:59 AM')
END
END

```

- k) Enter the BEGIN END container and in the container, enter the SET statement to set "OffPeak" as the value for the WL\_Group variable.

```

"...
IF (@CurrentTime >= '12:00 AM' AND @CurrentTime <= '8:59 AM')
BEGIN
SET @WL_Group = 'OffPeak'
END
END

```

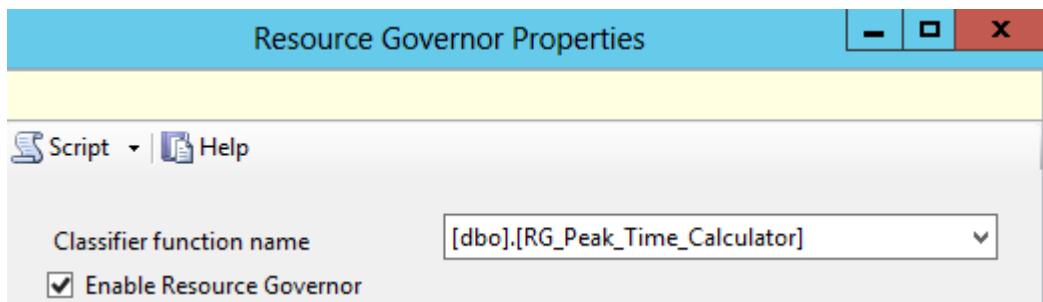
- l) Enter the RETURN statement to return the value of the WL\_Group variable followed by the END statement to indicate the end of the function.

```

"...
BEGIN
SET @WL_Group = 'OffPeak'
END
RETURN @WL_Group
END

```

- m) Execute the query to create the RG\_Peak\_Time\_Calculator() function.  
n) Switch back to the **Resource Governor Properties** window.  
o) In the **Object Explorer** pane, expand **Resource Pools** and right-click the **Web Transactions** pool and select **Properties**.  
p) In the Resource Governor Properties window, from the **Classifier function name** drop-down list, select **dbo.RG\_Peak\_Time\_Calculator**.



- q) Select **OK** to save the settings.

4. Specify a new query to verify that the new settings have taken effect.

- a) Open a **Query Editor** pane, enter the SELECT statement to select the group\_id, name, session\_id, login\_time, host\_name, and program\_name columns from the sys.dm\_exec\_sessions DMV with an alias "s."

```

SELECT s.group_id, g.name, s.session_id, s.login_time, s.host_name,
s.program_name
FROM sys.dm_exec_sessions s

```

- b) Enter the INNER JOIN statement to join the sys.dm\_exec\_sessions DMV with the sys.dm\_resource\_governor\_workload\_groups DMV.

```

"...
FROM sys.dm_exec_sessions s
INNER JOIN sys.dm_resource_governor_workload_groups g

```

- c) Enter the ON keyword to connect the common fields, g.group\_id and s.group\_id.

```
"...
INNER JOIN sys.dm_resource_governor_workload_groups g
ON g.group_id = s.group_id
```

- d) Enter the ORDER BY statement to sort the output in descending order of the g.name column.



**Note:** The following query checks if the settings are working and resource governor is actually using the correct set of resources.

```
"...
ON g.group_id = s.group_id
ORDER BY g.name DESC;
```

- e) Execute the query.  
f) On the **Results** tab, observe that the query returns the list of sessions and the name of the workload group they are using.



**Note:** You can use the workload group name returned, to study the pattern of usage of system resources and plan for further action to manage it.

SQLQuery2.sql - SQ...Administrator (53)\* X

```
SELECT s.group_id, g.name, s.session_id, s.login_time,
 s.host_name, s.program_name
 FROM sys.dm_exec_sessions s
 INNER JOIN sys.dm_resource_governor_workload_groups g
 ON g.group_id = s.group_id
 ORDER BY g.name DESC;
```

100 % < > Results Messages

|   | group_id | name     | session_id | login_time              | host_name | prog |
|---|----------|----------|------------|-------------------------|-----------|------|
| 1 | 257      | Lowtime  | 52         | 2013-03-15 23:09:01.363 | SQL01     | Rep  |
| 2 | 257      | Lowtime  | 53         | 2013-03-15 23:07:46.257 | SQL01     | Mic  |
| 3 | 257      | Lowtime  | 54         | 2013-03-15 23:08:20.940 | SQL01     | Rep  |
| 4 | 257      | Lowtime  | 59         | 2013-03-15 23:08:21.180 | SQL01     | Rep  |
| 5 | 1        | internal | 1          | 2013-03-15 22:57:53.157 | NULL      | NU   |



**Note:** The latest query sessions will be using the new groups that you created. Depending on the time of the day, you might see a different one listed as the time value.

# TOPIC B

## Optimize Indexes

You have managed server performance by identifying performance issues on the server and databases. Indexes determine the time taken for the server to respond to client queries and retrieve the required records. A database server can respond to client requests faster when the tables of the database have proper indexes. This will in turn ensure that the server can handle more client queries within a specific period. On the other hand, in case of non-index records, the time taken by the server to retrieve records is increased, resulting in lesser number of client queries being handled within the specific period. In this topic, you will optimize indexes.

### Statistics

*Statistics* is information stored about the distribution of values in columns of a table based on indexes created. The built-in query optimizer uses these statistics to estimate the number of rows that will be returned. Based on these estimates, SQL Server generates a query plan with the best performance.

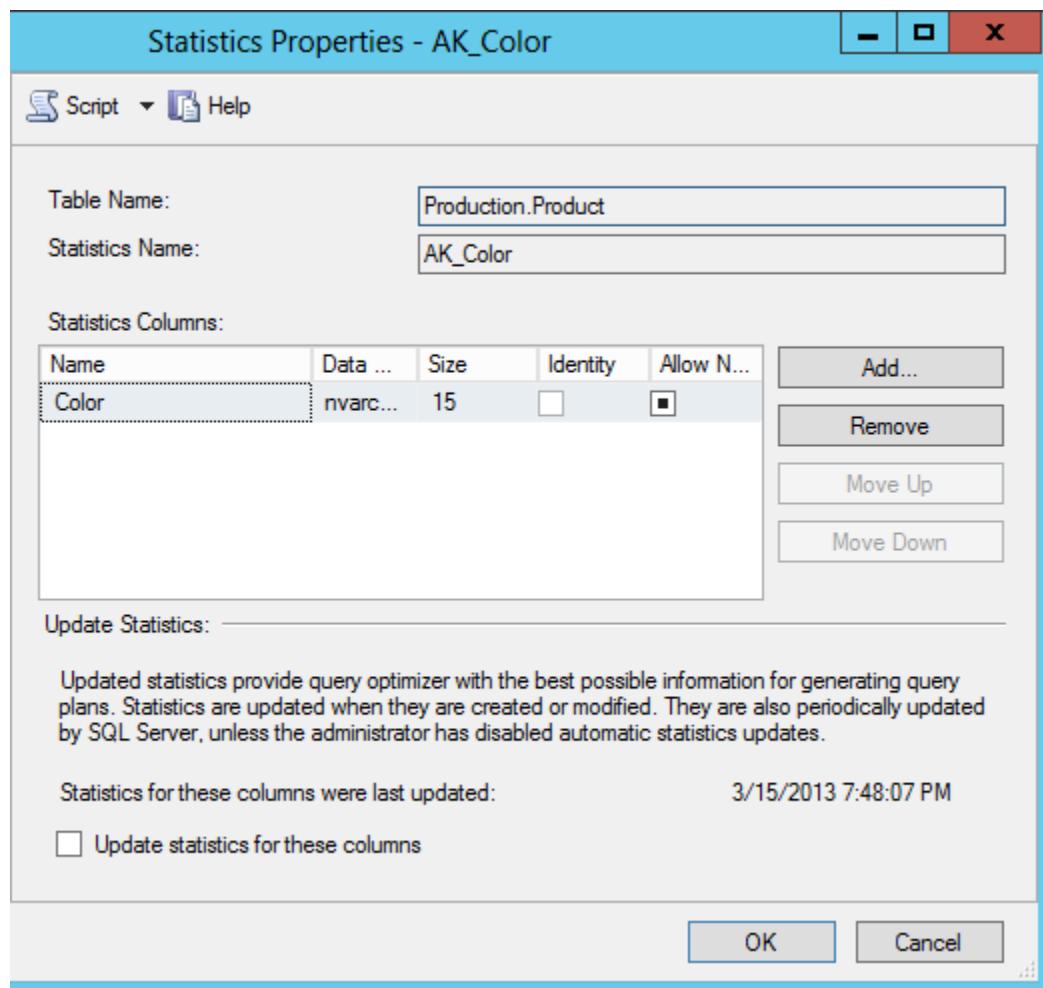


Figure 11-5: Statistics Properties window displaying the statistics column of Production.Product table.

## The Statistics Object

The **Statistics** object of the Object Explorer pane contains the various indexes and statistics. While there are a few indexes, there are many more statistics. The names with \_WA as the prefix are auto created based on queries run against the table, while those with a proper name (starting with AK) are pre-defined.

## Auto Statistics



### Auto Statistics (2 Slides)

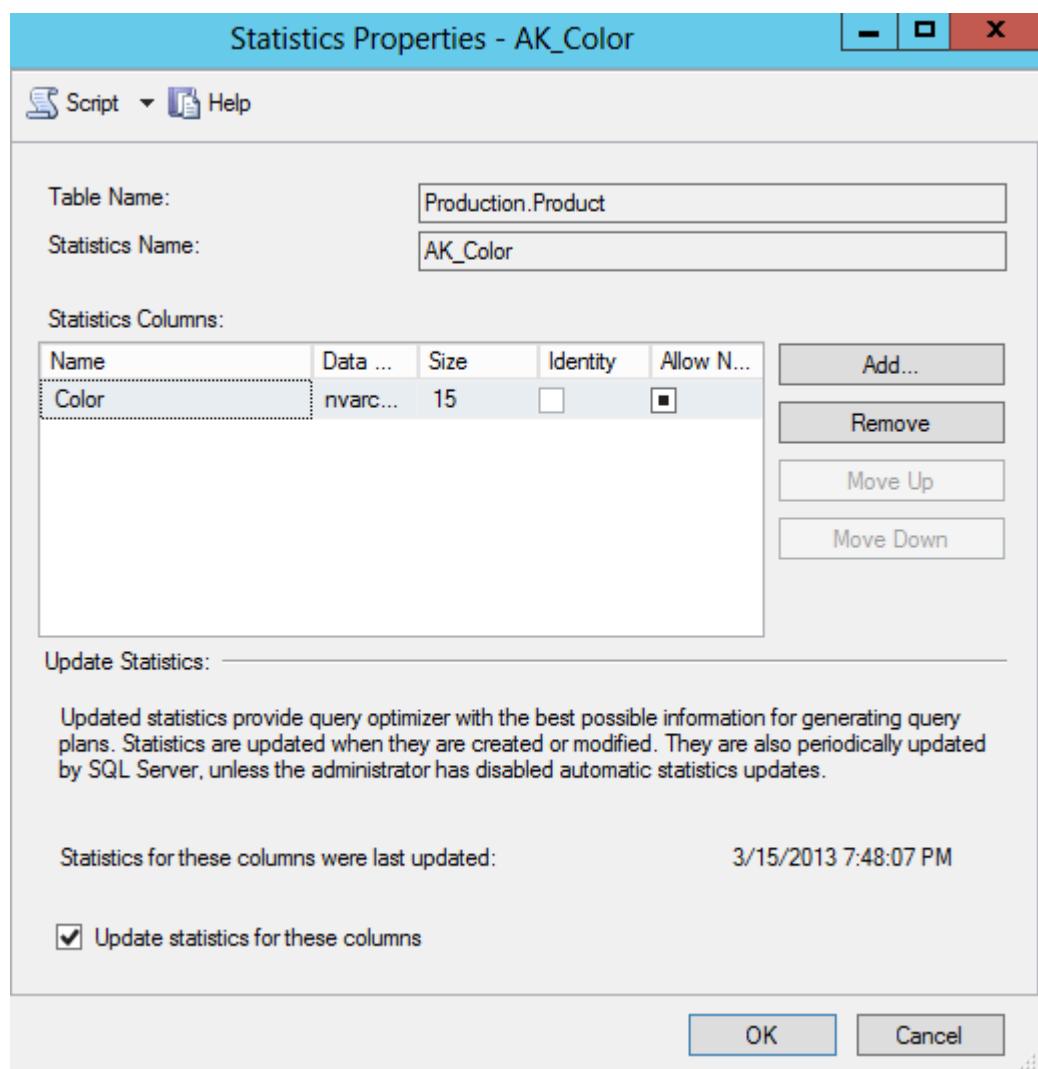
*Auto statistic* refers to statistics created automatically for all non-indexed columns in a query when a query that uses a column index is fired.

In most cases, statistics created are optimum. However, you can view, alter, or add new statistics, if required to optimize it further.

You need to run a query with a column name that is not in any index to create an auto statistic.

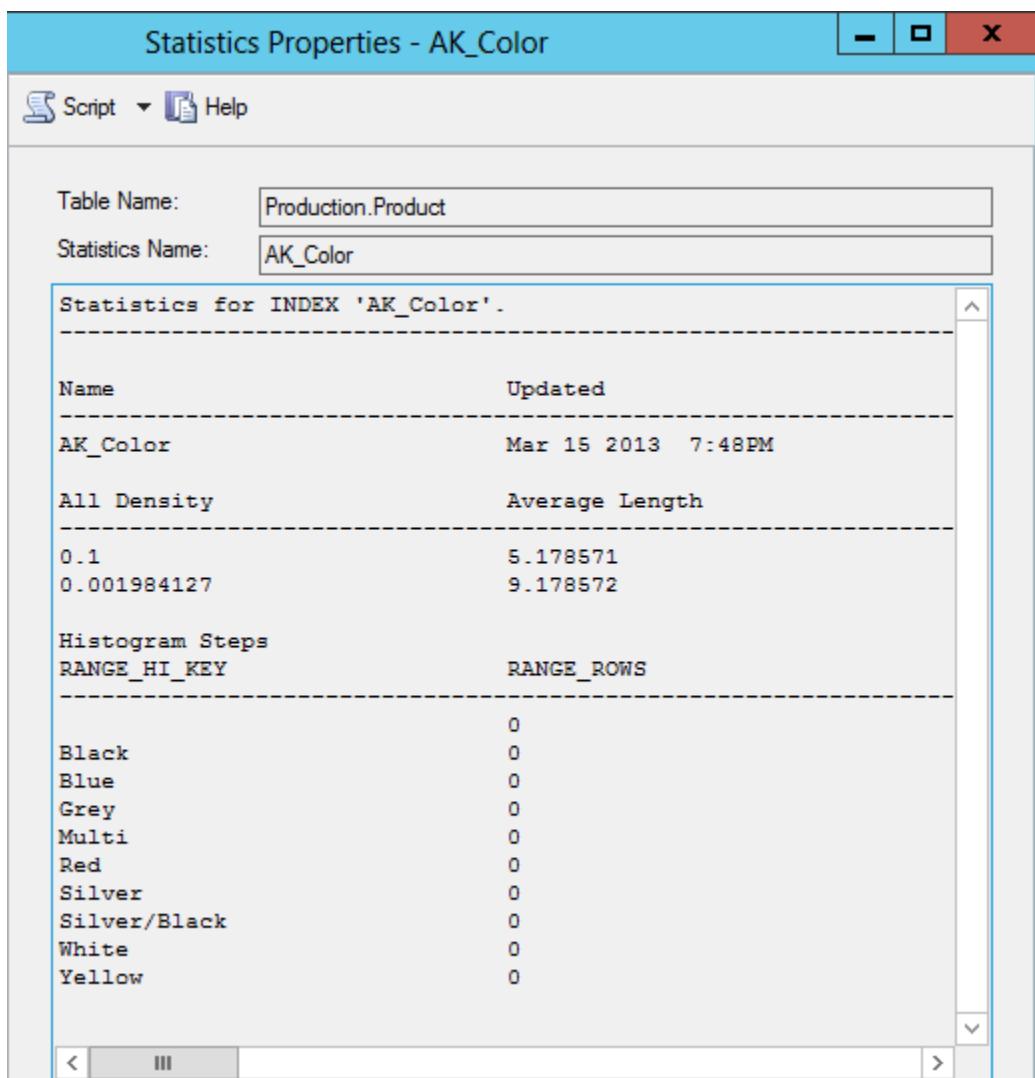
```
SELECT DISTINCT color FROM Production.Product;
```

For example, the above query is used to select unique rows from the color column of the Production.Product table. It will also create a new auto statistic for the column “color.”



**Figure 11-6: The Auto Statistics for the Color column with update statistics enabled.**

You can view the statistics associated for any column by opening the statistic and looking at the **Details** node.



**Figure 11-7:** The Statistics Properties dialog box with the Details page of AK\_Product\_Name statistics selected.



**Note:** In most cases, the details displayed by auto statistic are sufficient for any analysis. However, you can create your own statistic from the statistic node for each table and add the columns you want in the statistic.

## Fragmented Indexes

*Fragmented indexes* refer to indexes that have empty spaces in some pages or have split pages, which result in the logical order of indexes not matching the physical location of indexes in the database.

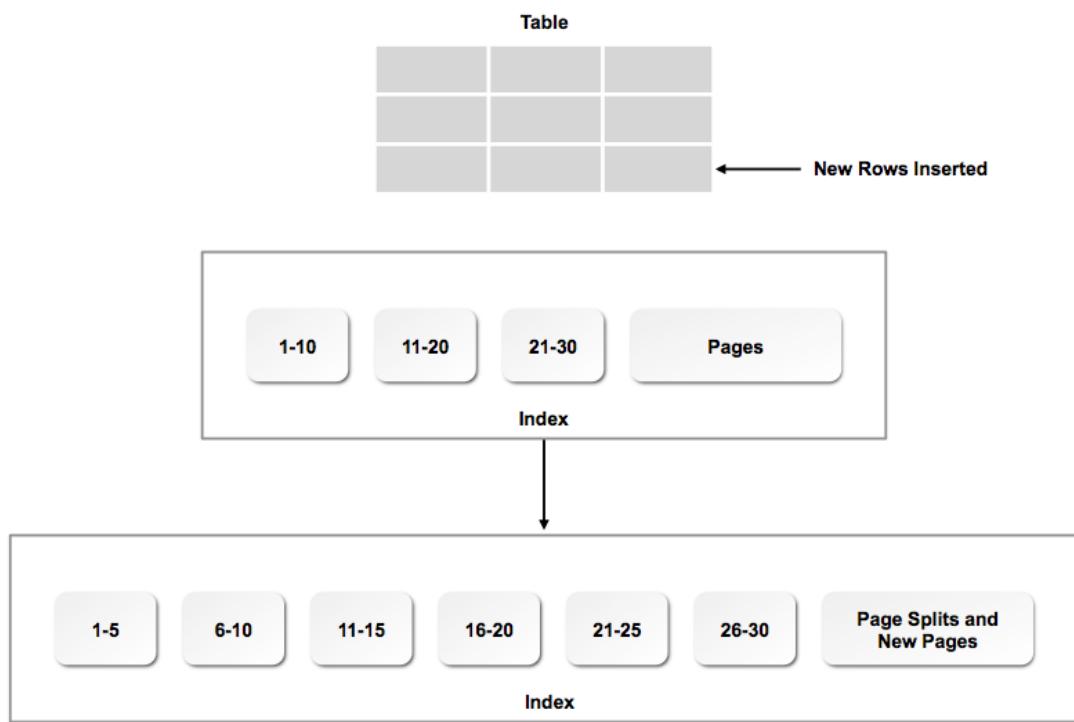
When an index is created on a table, the index contains a set of fixed size pages that contain the index entries. The pages themselves may not be full because they depend upon the fill factor.

However, when rows are inserted, updated, or deleted in the table, the index has to rearrange the entries within the pages. This can lead to either empty spaces in some pages or new pages being added or existing pages being split due to the addition of more entries.

This results in index fragmentation. This is very similar to fragmentation occurring on files on your hard disk, as they grow they are deleted or moved. When index fragmentation occurs, performance of queries on that table is affected. If there is only a little fragmentation, there will not be much impact, but with larger levels of fragmentation, the query performance will be severely affected.



Fragmented Indexes



*Figure 11-8: A fragmented index with page splits and new pages.*

## The DBCC SHOWCONTIG Method

  
The DBCC  
SHOWCONTIG Method

You can find out whether you have fragmentation issues in indexes by using two methods.

The first method is using the DBCC SHOWCONTIG statement. To view the details of all indexes of a table, you use the syntax:

```
DBCC SHOWCONTIG ('<table name>') WITH ALL_INDEXES;
```

This returns the number of pages, extents, and other fragmentation statistics.

```
SQLQuery37.sql - S...Administrator (69)* ×
DBCC SHOWCONTIG ('Production.Product') WITH ALL_INDEXES

100 % < |
Messages
DBCC SHOWCONTIG scanning 'Product' table...
Table: 'Product' (1973582069); index ID: 1, database ID: 5
TABLE level scan performed.
- Pages Scanned.....: 13
- Extents Scanned.....: 3
- Extent Switches.....: 2
- Avg. Pages per Extent.....: 4.3
- Scan Density [Best Count:Actual Count].....: 66.67% [2:3]
- Logical Scan Fragmentation: 15.38%
- Extent Scan Fragmentation: 66.67%
- Avg. Bytes Free per Page.....: 582.8
- Avg. Page Density (full).....: 92.80%
DBCC SHOWCONTIG scanning 'Product' table...
Table: 'Product' (1973582069); index ID: 2, database ID: 5
LEAF level scan performed.
- Pages Scanned.....: 2
- Extents Scanned.....: 1
- Extent Switches.....: 0
- Avg. Pages per Extent.....: 2.0
- Scan Density [Best Count:Actual Count].....: 100.00% [1:1]
- Logical Scan Fragmentation: 50.00%
- Extent Scan Fragmentation: 0.00%
- Avg. Bytes Free per Page.....: 437.0
- Avg. Page Density (full).....: 94.60%
DBCC SHOWCONTIG scanning 'Product' table...
Table: 'Product' (1973582069); index ID: 3, database ID: 5
LEAF level scan performed.
- Pages Scanned.....: 4
```

**Figure 11-9:** The output of the DBCC SHOWCONTIG method with a fragmented first index and non-fragmented second index.



**Caution:** DBCC SHOWCONTIG, while still supported, is going to deprecated in future versions of SQL Server. You should now use the other method for looking at index fragmentation.

## The DMV Method

The second method for identifying fragmentation issues uses a DMV named sys.dm\_db\_index\_physical\_stats. This accepts the database, index, and some other parameters to return the index statistics. To use the DMV, specify the required columns to it.

```
sys.dm_db_index_physical_stats (databaseID, tableID, indexID, partitionID,
mode);
```

For instance, to see the same data as shown by the DBCC statement, you can use the following statement:

```
SELECT *
FROM sys.dm_db_index_physical_stats
(DB_ID('Adventureworks2012'), OBJECT_ID('Production.Product'), NULL,
NULL, 'LIMITED');
```



The DMV Method

This will return a table with all information including fragmentation information.

**A SELECT statement displaying all information including the fragmentation information from the AdventureWorks2012 database.**

|  | database\_id | object\_id | index\_id | partition\_number | index\_type\_desc |
| --- | --- | --- | --- | --- | --- |
| 1 | 9 | 1973582069 | 1 | 1 | CLUSTERED INDEX |
| 2 | 9 | 1973582069 | 2 | 1 | NONCLUSTERED INDEX |
| 3 | 9 | 1973582069 | 3 | 1 | NONCLUSTERED INDEX |
| 4 | 9 | 1973582069 | 4 | 1 | NONCLUSTERED INDEX |

 The status bar at the bottom indicates 4 rows were returned."/>

```
SQLQuery20.sql - SQL01.AdventureWorks2012 (SQL01\Administrator (74))*
SELECT *
FROM sys.dm_db_index_physical_stats
(DB_ID('Adventureworks2012'),OBJECT_ID('Production.Product'),
NULL,NULL,'LIMITED');
```

|   | database_id | object_id  | index_id | partition_number | index_type_desc    |
|---|-------------|------------|----------|------------------|--------------------|
| 1 | 9           | 1973582069 | 1        | 1                | CLUSTERED INDEX    |
| 2 | 9           | 1973582069 | 2        | 1                | NONCLUSTERED INDEX |
| 3 | 9           | 1973582069 | 3        | 1                | NONCLUSTERED INDEX |
| 4 | 9           | 1973582069 | 4        | 1                | NONCLUSTERED INDEX |

SQL01 (11.0 SP1) | SQL01\Administrator (74) | AdventureWorks2012 | 00:00:00 | 4 rows

*Figure 11-10: The output of the DMV method.*



**Note:** To reduce the fragmentation of an index, you will need to either rebuild it or drop and recreate it.



How often do you rebuild the indexes on your SQL Server? What are the issues that you have faced because of fragmented indexes?



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Optimize Indexes

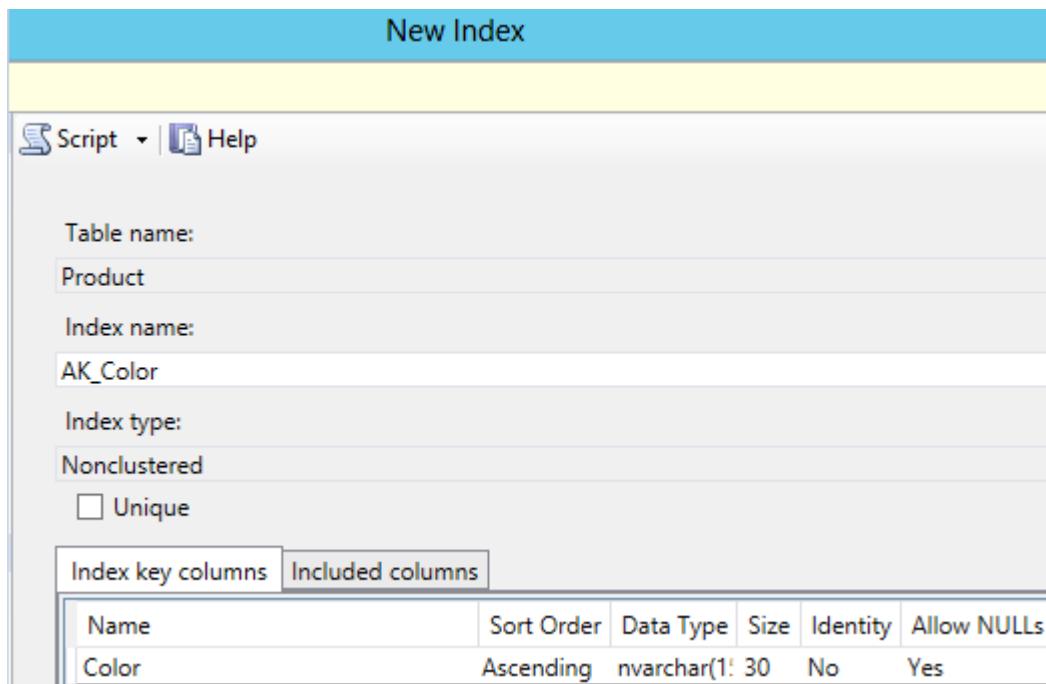
# ACTIVITY 11–5

## Adding an Index

### Scenario

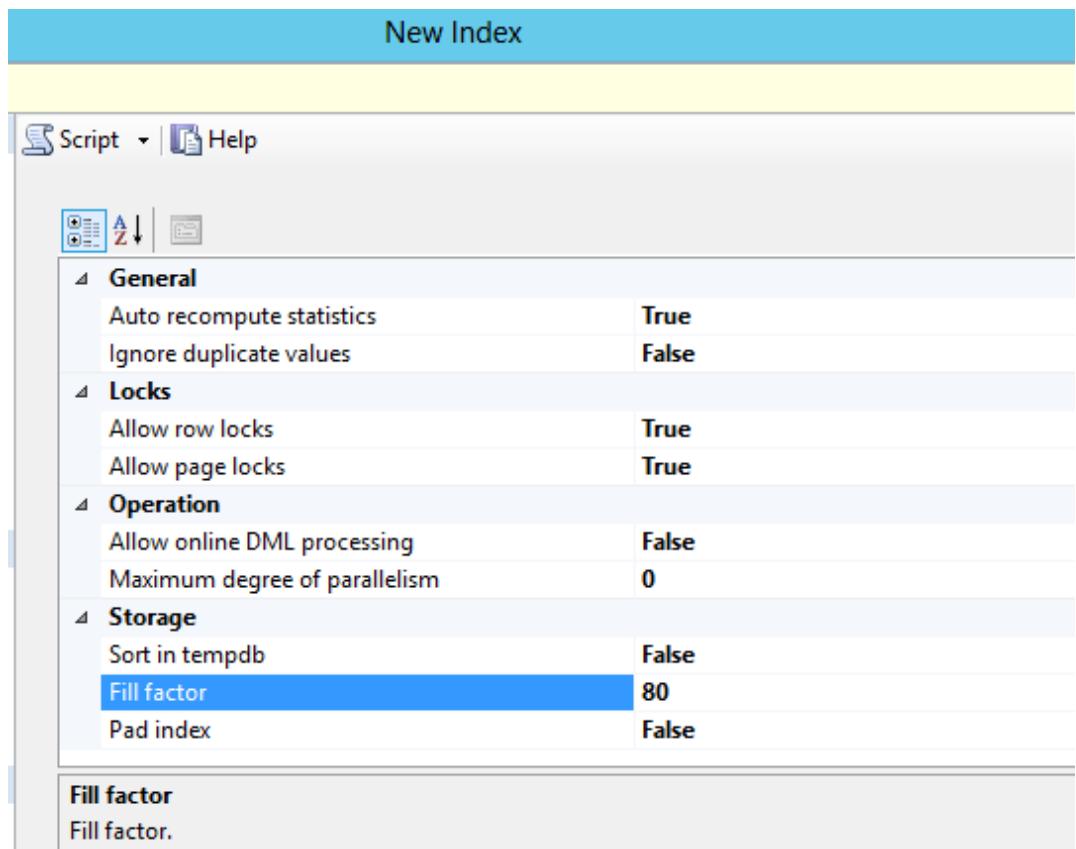
You have received complaints from users of the marketing and sales team that queries on a frequently queried column, Color, are slow. You decide to add a new index on that column to speed up the queries. After adding the new index, you decide to check the settings for the newly created index to ensure that the index is fully functional. Finally, after modifying the index, you need to manually generate the statistic to see the effect of the changes.

1. Add a non-clustered index named AK\_Color.
  - a) In the **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
  - b) In the **Object Explorer** pane, expand **SQL01→Databases→AdventureWorks2012→Tables→Production.Product→Indexes**.
  - c) Right-click the **Indexes** node and select **New Index→Non-clustered Index**.
  - d) In the **New Index** window, in the **Index name** text box, type **AK\_Color** as the index name.
  - e) On the **Index key columns** section, select **Add**.
  - f) In the **Select Columns from 'Production.Product'** dialog box, check the **Color** check box and select **OK**.

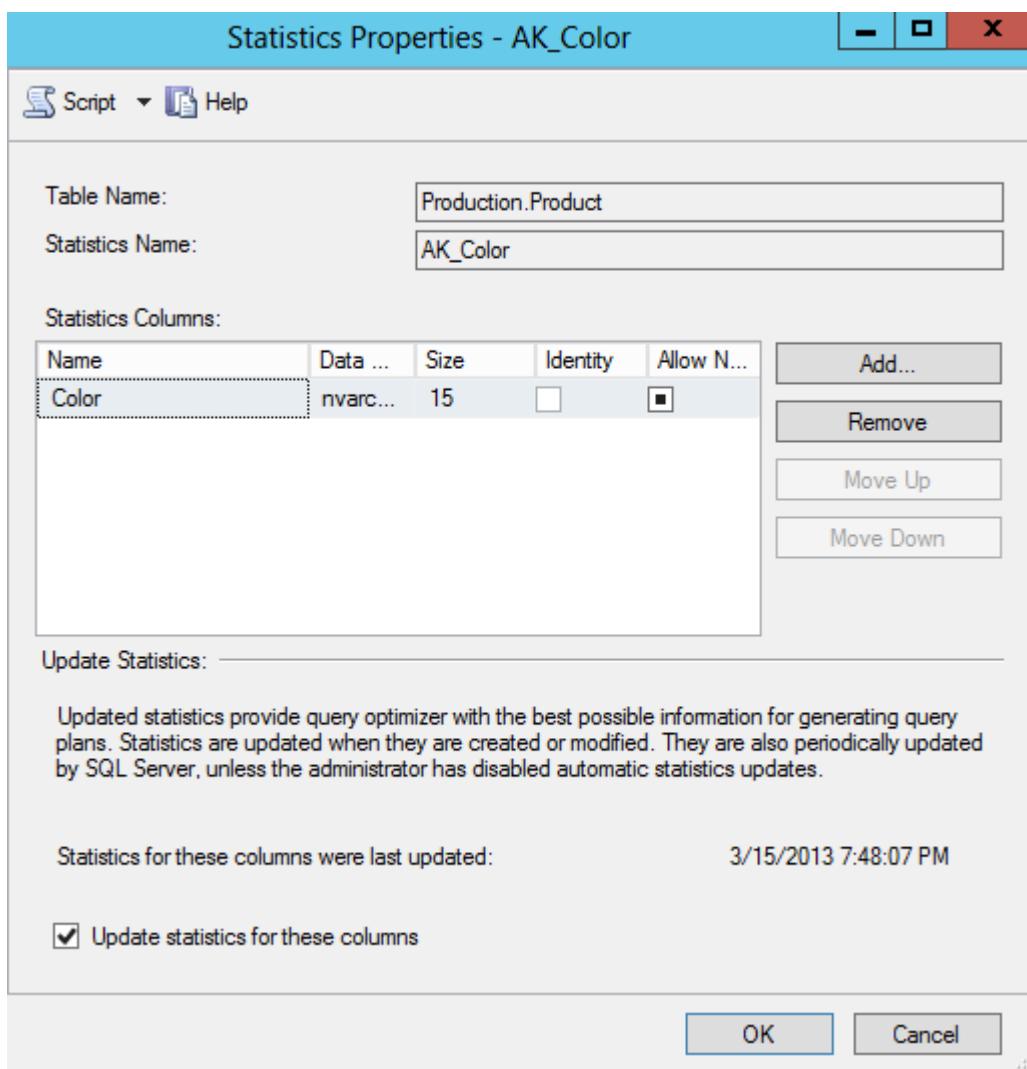


- g) In the **New Index** window, select **OK** to create the new index.
2. Modify the settings of the AK\_Color index and rebuild the index.
  - a) In the **Object Explorer** pane, expand **Indexes** and right-click the **AK\_Color** index and select **Properties**.
  - b) In the **Select a page** section, select **Options**.
  - c) On the **Options** page, view the current settings.

- d) In the **Storage** section, in the **Fill Factor** text box, type **80** to set the fill value in percentage.



- e) In the Index Properties window, select **OK** to save the changes.  
f) In the **Microsoft SQL Server Management Studio** message box, select **OK** to apply the changes and rebuild the index.
3. Manually update the statistics for the AK\_Color index.
- In the Object explorer, expand **SQL01→Databases→AdventureWorks2012→Tables→Production.Product→Statistics→AK\_Color**.
  - Right-click the **AK\_Color** statistic and select **Properties**.
  - In the **Statistics Properties - AK\_Color** dialog box, in the **Select a page** section, select the **Details** page to view the details of the statistics collected.
  - In the **Select a page** section, select the **General** page and check the **Update statistics for these columns** check box.



- e) Select **OK** to run and update the statistics whenever modifications are carried out on the selected columns. Close the **Query Editor** panes without saving the queries.
4. True or False? When you create an index on the table, the index will contain only fixed size pages with index entries.
- True  
 False
5. True or False? When you change the settings of an index, SQL Server will automatically rebuild the index.
- True  
 False
6. True or False? Index fragmentation affects the performance of queries on the related table.
- True  
 False
7. True or False? You cannot modify the statistics that is generated by the built-in query optimizer.
- True  
 False

## Summary

In this lesson, you managed the server performance by using monitoring tools such DMV and Profiler to identify the factors that affect the server performance. Such periodic monitoring ensures that the server always gives optimum performance and any potential issues with the server can be identified upfront and suitable corrective action planned to ensure that day-to-day activities are not affected in the event of server failure. In addition, you optimized the indexes to ensure that the server response time is kept to the minimum, which will allow the server to handle more client requests within a given time.



Use the review questions provided to generate discussion among the participants.

**What do you think are the benefits of using DMVs to query the server system state?**

**A:** Answers may vary, but may include: DMVs are easy to use because they are similar to the SELECT query and can be used on tables and functions; they are organized into groups in the database, so they can be easily identified by their names that indicate their functionality; and a DMV is a simple tool that can be easily handled by database administrators.

**What do you think are the similarities between Index Fragmentation and Disk Fragmentation?**

**A:** Answers may vary, but may include: both are similar in that they severely affect the system performance, both occur on pages-index page and memory page, respectively, and both can be easily detected using built-in tools and rectified by periodic monitoring.



**Note:** Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

# 12

# Troubleshooting Issues and Recovering Databases

**Lesson Time:** 2 hours, 15 minutes

## Lesson Objectives

In this lesson, you will:

- Troubleshoot database issues.
- Perform a database recovery.

## Lesson Introduction

You are now familiar with optimizing the server performance. When maintaining an organizational database, it is important to know the issues that may occur during database operations. You should also be aware of the recovery methods used to recover data that may be lost in the event of database failures. A better understanding on database troubleshooting and recovery methods will enable you to manage and recover the database data in case of a failure. In this lesson, you will troubleshoot database issues and recover databases.

# TOPIC A

## Troubleshoot Database Issues

When working with organizational data, it is important to identify the root cause of an issue and resolve it. SQL Server 2012 includes built-in tools that help you identify and troubleshoot database issues that can creep up during data transaction. Familiarizing yourself with the various techniques to troubleshoot database issues will enable you to monitor and manage your organizational database. In this topic, you will troubleshoot database issues.

### DBCC

Database Console Commands (*DBCC*) is a set of commands that are executed in the database engine and SQL Server to return the results of the execution back to the user.

The syntax to use the DBCC command is as follows.

```
DBCC <command name> <command parameters>;
```

Whenever a DBCC command is executed in a database, SQL Server first tries to take a read-only snapshot of the database and then run the command against the snapshot. In some cases, when the snapshot is unable to be taken or is not required, the command runs against the original database itself. In this case, it sets up locks on whichever resource it is accessing at the time to prevent any concurrency issues.



**Note:** For more information about DBCC, check out the LearnTO **Identify DBCC Commands to Troubleshoot Issues** presentation by selecting the **LearnTO** tile on your LogicalCHOICE Course screen.



You may want to show LearnTO **Identify DBCC Commands to Troubleshoot Issues** from the LogicalCHOICE Course screen or have students navigate out to the Course screen and watch it themselves as a supplement to your instruction. If not, please remind students to visit the LearnTOS for this course on their LogicalCHOICE Course screen after class for supplemental information and additional resources.

### The DBCC Command Categories

DBCC commands are primarily categorized into four categories.

| DBCC Command Category | Used To                                                                                                                                   |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Maintenance           | Run maintenance processes against databases and other objects.                                                                            |
| Informational         | Collate information from different places and present them to the user.                                                                   |
| Validation            | Perform validation processes on different objects.                                                                                        |
| Miscellaneous         | Perform miscellaneous operations such as enabling trace flags and removing the Dynamic Link Library (DLL) files from the database memory. |

For instance, the DBCC CHECKDB is one of the most used and popular of such commands. This command has the ability to internally run a number of other DBCC commands, such as CHECKALLOC, CHECKTABLE, and CHECKCATALOG, along with other validation processes.

The syntax to declare the DBCC CHECKDB is as follows.

```
DBCC CHECKDB (<database>);
```

### Example

The DBCC CHECKDB command to check the SalesDB database is as follows.

```
DBCC CHECKDB (SalesDB);
```

When you execute this DBCC command, the state of the SalesDB database and the objects within it will be displayed.

## Maintenance DBCC Commands

Maintenance DBCC commands are used to monitor the database and database objects. Some of the most commonly used maintenance DBCC commands are listed in the table.

| <b>Maintenance DBCC Command</b> | <b>Used To</b>                                                              |
|---------------------------------|-----------------------------------------------------------------------------|
| DBREINDEX                       | Rebuild the indexes for a specified database object in a database.          |
| INDEXDEFRAG                     | Defragment indexes for a specified table in a database.                     |
| SHRINKDATABASE                  | Reduce the size of the database and log files in a database.                |
| SHRINKFILE                      | Shrink the size of a specified file in a database.                          |
| UPDATEUSAGE                     | Correct row and page count discrepancies in the catalog view of a database. |

## Informational DBCC Commands

Informational DBCC commands are used to collect database information such as database server configuration, resource usage, and configuration of various database objects. Some of the most commonly used informational DBCC commands are listed in the table.

| <b>Informational DBCC Command</b> | <b>Used To</b>                                                                                                                                  |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| OPENTRAN                          | Display information, such as oldest active transactions and distributed and non-distributed replicated transactions, from a specified database. |
| OUTPUTBUFFER                      | Display the active output buffer in ASCII and hexadecimal format.                                                                               |
| INPUTBUFFER                       | Display the last transaction or statement sent from a client to a server instance.                                                              |
| PROCCACHE                         | Display the information about procedure cache in a table format.                                                                                |
| SHOWCONTIG                        | Display the information about the data and indexes in a specified table or view.                                                                |
| SQLPERF                           | Display the information about the transaction log space usage statistics of all databases in the server.                                        |

## Validation DBCC Commands

Validation DBCC commands are used to perform a consistency check on a database. Some of the most commonly used validation DBCC commands are listed in the table.

| <b>Validation DBCC Command</b> | <b>Used To</b>                                                       |
|--------------------------------|----------------------------------------------------------------------|
| CHECKALLOC                     | Check the disk space allocation consistency in a specified database. |

| <b>Validation DBCC Command</b> | <b>Used To</b>                                                                              |
|--------------------------------|---------------------------------------------------------------------------------------------|
| CHECKCATALOG                   | Check the catalog consistency in a specified database.                                      |
| CHECKCONSTRAINTS               | Check the reliability of the constraints placed in a specific table of a database.          |
| CHECKDB                        | Check the integrity of all database objects in a database.                                  |
| CHECKFILEGROUP                 | Check the space allocation and structural integrity of the tables in a specified filegroup. |
| CHECKIDENT                     | Check the current identity value for a specific table.                                      |

## Miscellaneous DBCC Commands

Miscellaneous DBCC commands are used to perform additional database tasks such as activating locks, enabling row-locking feature, enabling trace feature, and removing DLL from memory. Some of the most commonly used miscellaneous DBCC commands are listed in the table.

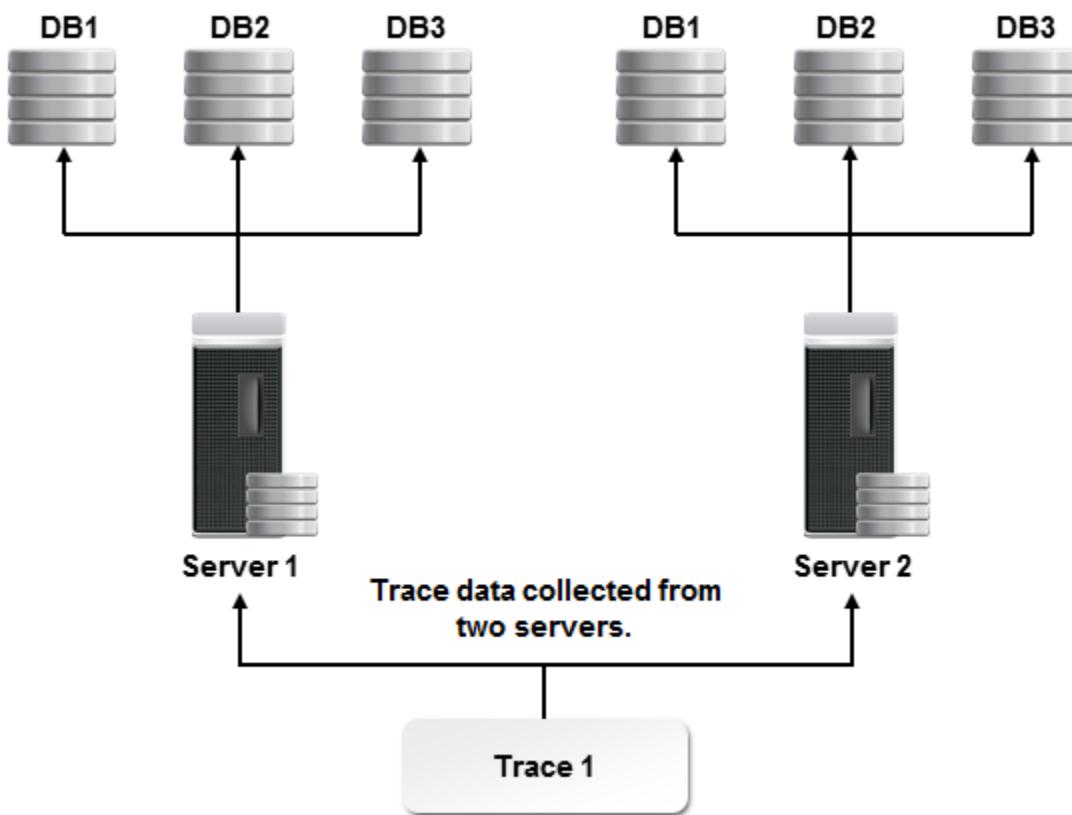
| <b>Validation DBCC Command</b> | <b>Used To</b>                                                                      |
|--------------------------------|-------------------------------------------------------------------------------------|
| HELP                           | Display the DBCC commands with help information.                                    |
| ROWLOCK                        | Enable row-lock feature in tables.                                                  |
| PINTABLE                       | Mark a table that needs to be pinned.                                               |
| TRACEON                        | Enable the trace feature on a table.                                                |
| TRACEOFF                       | Disable the trace feature on a table.                                               |
| UNPINTABLE                     | Mark a table as unpinned and then clears the buffer cache on a specific table page. |

## Trace Data



*Trace data* is a collection of events and data returned by the database engine. When working with SQL Server, the engine collects data about the components that run on the system. This includes items such as connections being made, queries being fired, tables, rows, and indexes. Trace data plays a vital role in monitoring performance or in troubleshooting issues.

You can use the SQL Profiler tool to collect the trace data and figure out any optimizations you can perform or steps you can take to troubleshoot issues.



**Figure 12-1: Trace data being collected from two servers.**

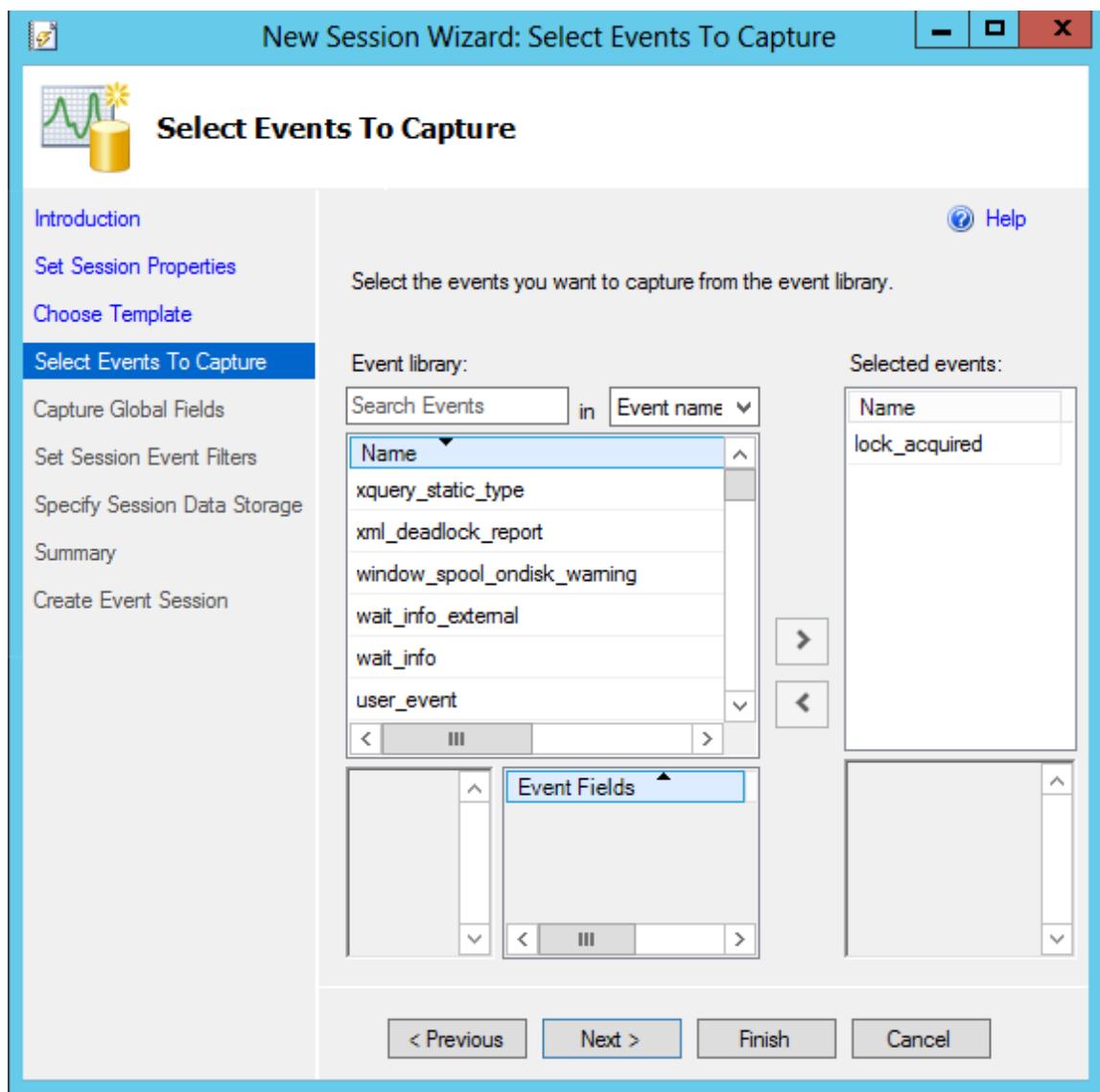
## XEvents

*Extended Events*, or XEvents, is a system feature that lets you query the SQL Server engine to gather as much or as little information as you require to troubleshoot any issues. These XEvents themselves are lightweight programs and take up minimal resources.



XEvents collect data from SQL Server, the OS, and even client applications. Any event that can be captured and submitted into the Events Tracing for Windows (ETW) system can be used by the XEvents to collect data.

SQL Server 2012 allows you to use XEvents from both a query as well as through the interface.



**Figure 12-2:** The *Select Events To Capture* page displaying the entire list of XEvents on the server.

## Example

The code snippet to display the entire list of XEvents that you can use on the server is as follows.

```
SELECT p.name, c.event, k.keyword, c.channel, c.description FROM
(
 SELECT event_package = o.package_guid, o.description,
 event=c.object_name, channel = v.map_value
 FROM sys.dm_xe_objects o
 LEFT JOIN sys.dm_xe_object_columns c ON o.name = c.object_name
 INNER JOIN sys.dm_xe_map_values v ON c.type_name = v.name
 AND c.column_value = cast(v.map_key AS nvarchar)
 WHERE object_type = 'event' AND (c.name = 'CHANNEL' or c.name IS NULL)
) c LEFT JOIN
(
 SELECT event_package = c.object_package_guid, event = c.object_name,
 keyword = v.map_value
 FROM sys.dm_xe_object_columns c INNER JOIN sys.dm_xe_map_values v
 ON c.type_name = v.name AND c.column_value = v.map_key
)
```

```

AND c.type_package_guid = v.object_package_guid
INNER JOIN sys.dm_xe_objects o ON o.name = c.object_name
AND o.package_guid = c.object_package_guid
WHERE object_type = 'event' AND c.name = 'KEYWORD'
) k
ON
k.event_package = c.event_package AND (k.event=c.event or k.event IS NULL)
INNER JOIN sys.dm_xe_packages p ON p.guid = c.event_package
ORDER BY keyword desc, channel, event

```

This code snippet will display the name, event, keyword, channel, and description columns from the server.



**Note:** This is a generic code sample to list the entire list of XEvents available in your server, you can either type the code snippet or just copy and paste the code snippet from MSDN (<http://msdn.microsoft.com/en-us/library/ff878413.aspx>).

## Guidelines to Test Connectivity Issues

Testing the server connectivity issues requires careful planning and implementation. There are some guidelines that you can follow while testing the server for connectivity issues.

- Verify that your server name and the IP address are correct.
- Ensure that you do not have any firewall settings blocking your server ports.
- Check for correct wiring, router, or switch configurations.
- Make sure that your database state is maintained in the online state.
- Ensure that you are able to PING against the SQL Server.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Manage Server Performance

# ACTIVITY 12-1

## Checking a Database Using the DBCC Command

### Before You Begin

1. In the **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
2. Open a **Query Editor** pane and ensure that the **AdventureWorks2012** database is selected.

### Scenario

The sales team of Adventure Works Cycles is planning to present all the products in a sales exposition. Before that, your sales manager wants you to check the state of the AdventureWorks2012 database and the objects within it. You decide to run the DBCC command to ensure that everything is consistent and working properly.

1. Execute the DBCC command to display the state of the database.
  - a) In the **Query Editor** pane, enter the DBCC CHECKDB statement followed by the database name AdventureWorks2012 to display the state and the objects within the specified database.

```
DBCC CHECKDB (AdventureWorks2012);
```

- b) Execute the DBCC command.



**Note:** This may take some time to execute based on the system configuration.

- c) On the **Messages** tab, scroll-down and view each line of the results, and observe that each object in the database has been checked.

```
SQLQuery1.sql - SQ...Administrator (52)*
DBCC CHECKDB (AdventureWorks2012);

100 % <
Messages
CHECKDB found 0 allocation errors and
0 consistency errors in database 'Adventureworks2012'.
DBCC execution completed. If DBCC printed error messages,
contact your system administrator.
```



**Note:** The last two lines in the result set state if there were any consistency or allocation errors in the AdventureWorks2012 database.

- d) Close the **Query Editor** pane without saving the query.
2. Which DBCC command category is used to collate information from different places and present them to a specific user?
    - Maintenance

- Informational
- Validation
- Miscellaneous

**3. Which DBCC command has the ability to internally run a number of other DBCC commands?**

- CHECKALLOC
  - CHECKTABLE
  - CHECKDB
  - CHECKCATALOG
-

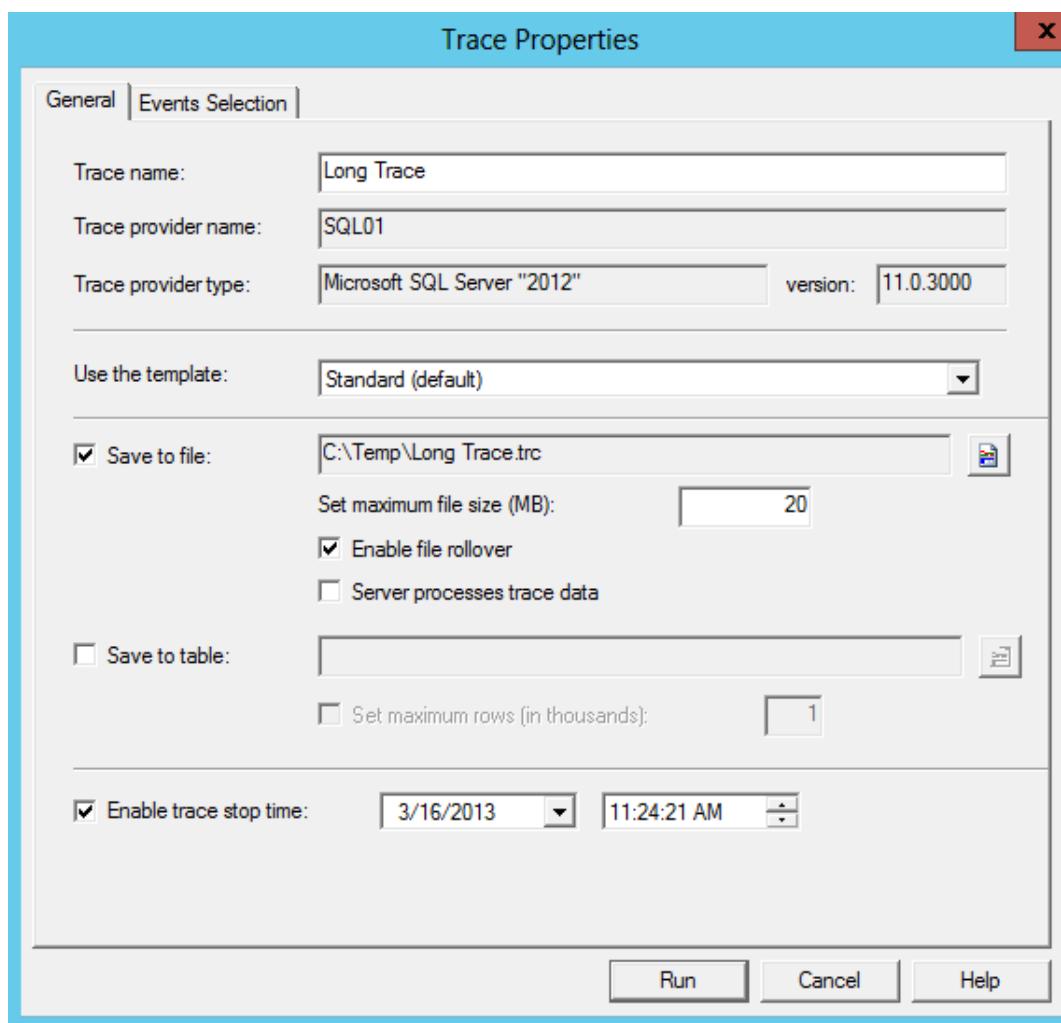
# ACTIVITY 12-2

## Collecting Trace Data for a Database

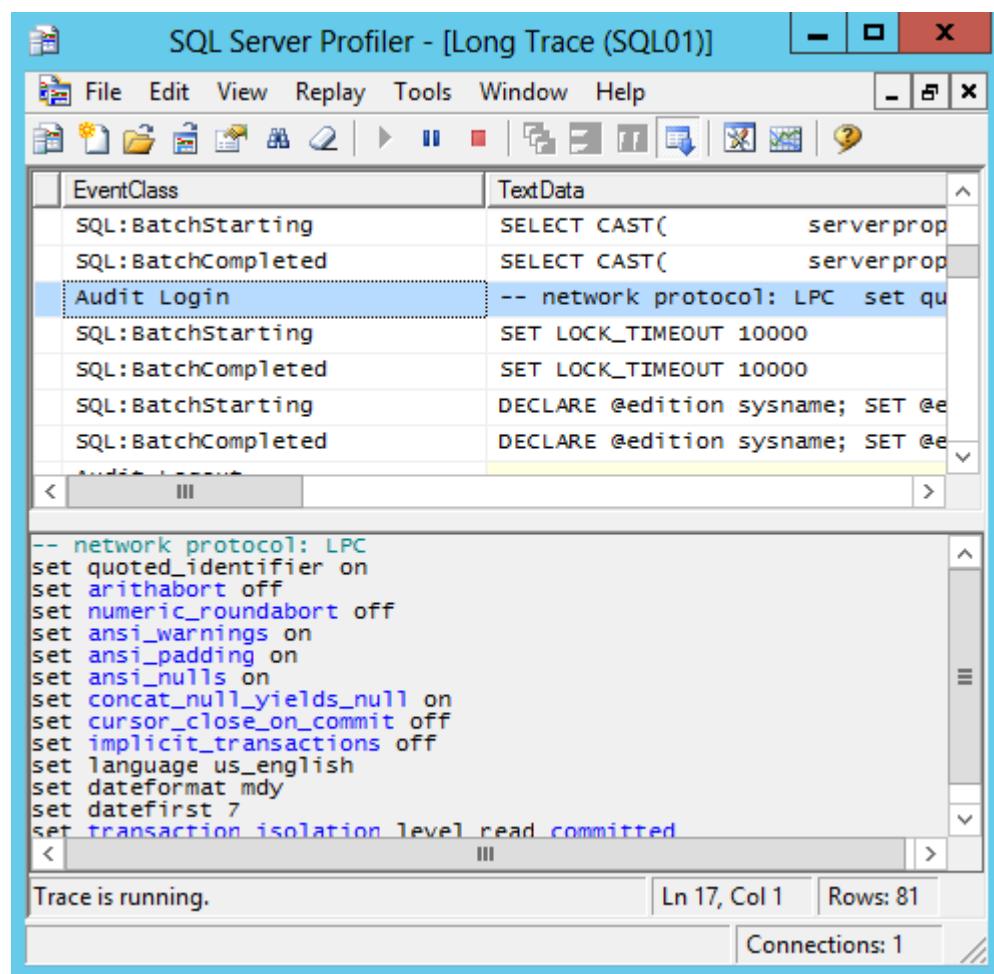
### Scenario

The database manager of Adventure Works Cycle wants you to collect the events and data returned by the database engine. You decide to collect tracing information to be used for troubleshooting or optimization of the server.

- 
1. Initiate the trace data collection process.
    - a) Close SSMS and from the Start Screen of Windows Server 2012, select the **SQL Server Profiler** icon.
    - b) In the **SQL Server Profiler** toolbar, select the **New Trace**  button.
    - c) In the **Connect to Server** dialog box, accept the default settings and select **Connect**.
    - d) In the **Trace Properties** dialog box, in the **Trace name** text box, type **Long Trace**.
    - e) From the **Use the template** drop-down list box, ensure that **Standard (default)** is selected.
    - f) In the **Trace Properties** dialog box, check the **Save to file** check box and in the **Save As** dialog box, select **Save**.
    - g) In the **Set maximum file size (MB)** text box, type **20** and then check the **Enable trace stop time** check box. Ensure that the stop time is five minutes more than the current time.
    - h) In the **Trace Properties** dialog box, select **Run** to start collecting the trace data.



2. Log in to SSMS and view the trace data.
  - a) From the Start Screen of Windows Server 2012, open SSMS by selecting the SSMS icon.
  - b) In the **Connect to Server** dialog box, in the **Server Name** text box, ensure that **SQL01** is selected and select **Connect**.
  - c) Switch back to the SQL Server Profiler window.
  - d) Once the trace data get collected, browse through the trace log file to view the processes that happened during the login and logoff of the server instance.



- e) Observe that the trace data for login and logoff processes of the server are collected and select the **Close** button to close the SQL Server Profiler window.
- f) In the **SQL Server Profiler** dialog box, select **Yes** to close it.

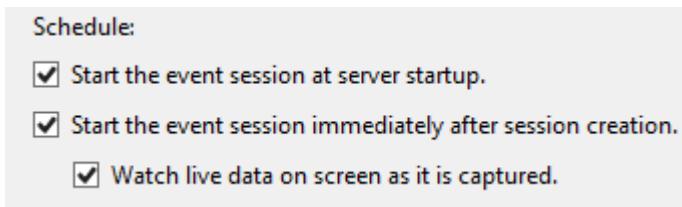
# ACTIVITY 12-3

## Troubleshooting Issues Using XEvents

### Scenario

The sales manager of Adventure Works Cycles is planning to host a sales presentation; during his presentation, he may require data from different databases in the server. Before getting into the presentation, he wants you to check if there are any issues on the database objects of the server. You decide to use XEvents to troubleshoot any issues on the database objects of the server.

1. Creating a new event session named Database Check.
  - a) In the **Object Explorer** pane, expand **SQL01** and expand the **Management** folder.
  - b) In the **Management** folder, expand **Extended Events** and select the **Sessions** folder.
  - c) Right-click the **Sessions** folder and from the fly-out pane, select **New Session**.
  - d) In the New Session window, in the **Session name** text box, type the session name as **Database Check**.
  - e) In the New Session window, check the following check boxes.
    - **Start the event session at server startup.**
    - **Start the event session immediately after session creation.**
    - **Watch live data on screen as it is captured.**



2. Select the required events from the event library.
  - a) In the **Select a Page** section, select **Events**.
  - b) In the **Event library** section, from the **Category** drop-down list, uncheck the **(Select All)** check box.
  - c) From the **Category** drop-down list, scroll down and check the **Index** check box.
  - d) In the **Name** section, select the **scan\_started** and then select the forward arrow  button to the right of the **Event library** section.

**Event library:**

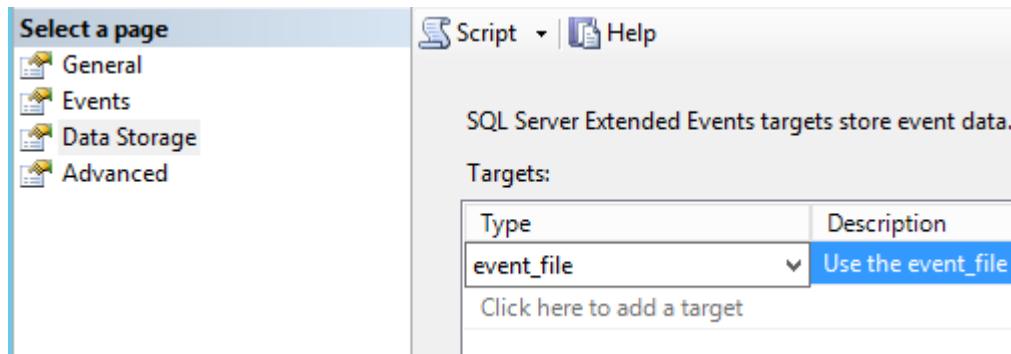
| Name                                   | Category | Channel |
|----------------------------------------|----------|---------|
| full_text_crawl_started                | index    | Admin   |
| full_text_crawl_stopped                | index    | Admin   |
| progress_report_online_index_operation | index    | Admin   |
| scan_started                           | index    | Admin   |
| scan_stopped                           | index    | Admin   |

- e) In the Name section, select the scan\_stopped row and then select the forward arrow  button.
- f) Observe that the selected rows are moved to the **Selected events** section.

**Selected events:**

| Name         |   |
|--------------|---|
| scan_started | 0 |
| scan_stopped | 0 |

3. Select the storage location to store the event file.
  - a) In the **Select a page** section, select **Data Storage**.
  - b) In the **Type** column, select **Click here to add a target** and from the drop-down list, select **event\_file**.



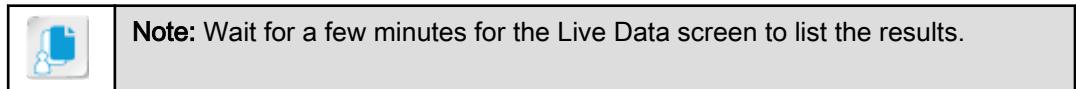
| Type       | Description        |
|------------|--------------------|
| event_file | Use the event_file |

- c) In the **File name on server** text box, type **C:\Temp\Database Check.xel**.
- d) In the **Maximum file size** text box, type the file size as **100** and from the drop-down list, select **MB**.

**Properties:**

|                                                          |                                                         |
|----------------------------------------------------------|---------------------------------------------------------|
| File name on server:                                     | <input type="text" value="C:\Temp\Database Check.xel"/> |
| Maximum file size:                                       | <input type="text" value="100"/> MB                     |
| <input checked="" type="checkbox"/> Enable file rollover |                                                         |
| Maximum number of files:                                 | <input type="text" value="5"/>                          |

- e) In the New Session window, select **OK** to save the session. This will open the **Live Data** screen. In the SSMS interface, observe the results of the scan getting listed on the screen.



SQL01 - Database Check: Live Data X

Displaying 385 Events

|  | name         | timestamp                   |
|--|--------------|-----------------------------|
|  | scan_started | 2013-03-16 11:39:18.9699389 |
|  | scan_started | 2013-03-16 11:39:18.9699532 |
|  | scan_started | 2013-03-16 11:39:18.9699671 |
|  | scan_started | 2013-03-16 11:39:18.9699810 |
|  | scan_started | 2013-03-16 11:39:18.9699939 |

Event: scan\_started (2013-03-16 11:37:03.9491549)

Details

| Field         | Value           |
|---------------|-----------------|
| database_id   | 1               |
| database_name |                 |
| hobt_id       | 281474978086912 |
| index_id      | 1               |
| object_id     | 21              |

- f) Close the **Live Data** screen after viewing the results.

# TOPIC B

## Perform a Database Recovery

Organizational databases are prone to hardware failure or can get corrupted due to continuous data transaction. As database administrator, it is your prime duty to protect your databases against known issues and you should be able to recover the database in case of an issue or data loss. A better understanding on how to recover a database in the event of a failure will enable you to safeguard and effectively manage your organizational data. In this topic, you will perform a database recovery.

### Database Recovery

*Database recovery* is the process of recovering the data from failed, corrupted, or inaccessible databases. When working with organizational data, it is important to protect and recover the database in case of failures such as database corruption and system failures. Data loss can become fatal when working with confidential organizational data. In such cases, extreme care must be taken to recover the data and return the database to working condition.

### Possible Reasons for Database Corruption

Database corruption is an unexpected error that occurs in a database and affects the data integrity and functionality available in a database. In the worst case, you may not be able to access the database and chances are that the data in it is lost.

The reason for database corruption is not confined to a single problem; there are several possibilities by which a database in SQL Server can get corrupted.

- Improper shutdown of the application.
- Inadvertent deletion of files such as log files, table, key, and certificates.
- Moving a database from one location to another location.
- Unexpected hard disk failures.
- Unexpected power failure while performing data transaction in a database.
- Operating system getting crashed when a database is open.
- Memory and network errors.

### Guidelines to Recover Data from a Corrupted Drive

Recovering database data from a corrupted drive requires careful examination and in some cases data recovery will be extremely expensive. There are some guidelines; you can follow while recovering the database data from a corrupted hard drive.

- Check the hard disk for external damages such as broken parts, hot spots, and rattling noises.
- Disconnect drive and try connecting to a different IDE/SCSI controller or different system.
- Replace the existing hard disk power cable with a new one.
- Use third party disk recovery software to read the files.
- Perform a sector scan using software to recover data if possible.
- Send drive to disk recovery specialization companies to attempt for data recovery.

### Guidelines to Identify and Troubleshoot Data Access Problems

There are some guidelines; you can follow while identifying and troubleshooting data access problems in SQL Server. You need to:

- Check if the server is on the network using the ping utility.

- Check if server firewall is off or configured correctly.
- Check if you are using the correct authentication.
- Check if user login has the required rights to access the database.
- Check if the required data exists on the server.
- Check if appropriate database rules are followed while creating the database.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Perform a Database Recovery

# ACTIVITY 12-4

## Performing a Database Recovery

### Before You Begin

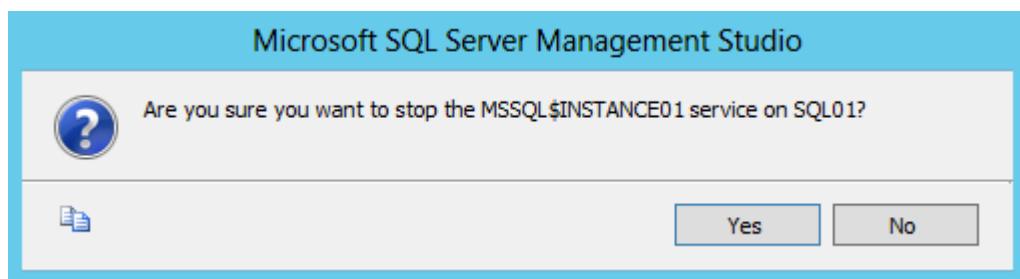
Attach the AdventureWorksLT2012 database to the SQL01\INSTANCE01 instance.

1. In **Object Explorer** pane, right-click **SQL01**, select **Stop** and select **Disconnect**.
2. In **Object Explorer** pane, select **Connect→Database Engine**.
3. In the **Connect to Server** dialog box, in the **Server Name** text box, select **SQL \INSTANCE01** and select **Connect**.
4. In the **Object Explorer** pane, in the **SQL01\INSTANCE01** instance, right-click the **Databases** folder and select **Attach**.
5. In the **Attach Databases** dialog box, select **Add**.
6. In the **Locate Database Files** dialog box, navigate to the **C:\093108Data\AdventureWorksLT2012\_Database** folder and select the **AdventureWorksLT2012\_Data.mdf** file.
7. In the **Locate Database Files** dialog box, select **OK**.
8. In the **Attach Databases** dialog box, select **OK**.

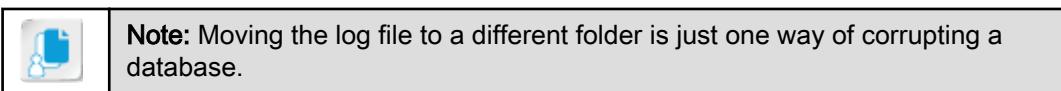
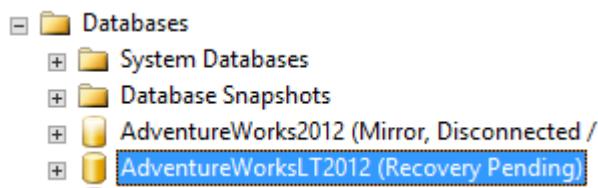
### Scenario

Your database manager wants you to train the database administration team on how to perform a database recovery during unexpected database failure. You are asked to perform a simulated crash test on the AdventureWorksLT2012 database by moving its log file to a different location and then recover the database back to its working condition.

1. Moving the log file of AdventureWorksLT2012.
  - a) In the **Object Explorer** pane, right-click **SQL01\INSTANCE01** and from the fly-out pane, select **Stop**. In the message box, select **Yes** to confirm each service when prompted.



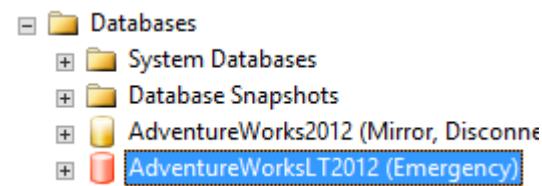
- b) Open the Windows Explorer, navigate to **C:\093108Data\AdventureWorksLT2012\_Database** containing the **AdventureWorksLT2012\_log.ldf** and move the **.ldf** file to the folder named **Drive1** in the **C** drive.
- c) In the SSMS, in the **Object Explorer** pane, right-click **SQL01\INSTANCE01** and from the fly-out pane, select **Start**. In the message box, select **Yes** to confirm each service when prompted.
- d) In the **Object Explorer** pane, expand the **Databases** folder and select **AdventureWorksLT2012**. Next to the database name, observe that the status of the database will be changed to **Recovery Pending**. This indicates that the AdventureWorksLT2012 database is corrupted.



2. Alter the AdventureWorksLT2012 database by changing its state to Emergency.
  - a) Open a **Query Editor** pane and enter the ALTER DATABASE statement followed by the database name AdventureWorksLT2012 and set the status as SET EMERGENCY to indicate that the database recovery process is initiated in this database.

```
ALTER DATABASE AdventureWorksLT2012 SET EMERGENCY;
```

  - b) Execute the ALTER DATABASE statement.
  - c) In the **Object Explorer** pane, right-click **AdventureWorksLT2012** and from the fly-out pane, select **Refresh**. Next to the database name, observe that the status of the database is changed to **Emergency**.



3. Check the errors in the SQL Server log file.
  - a) In the **Object Explorer** pane, expand **Management**→**SQL Server logs**.
  - b) Right-click **Current** and from the fly-out pane, select **View SQL Server Log**. In the Log File Viewer window, scroll down to the point where you can see the last place where it says Starting up database 'AdventureworksLT2012'.

```
Starting up database 'mssqlsystemresource'.
Starting up database 'AdventureWorksLT2012'.
```

- c) In the Log File Viewer window, scroll down and check the error that states FileMgr::StartLogFiles. The error message indicates that the log file is not to be found.

| Message                                                        |
|----------------------------------------------------------------|
| Starting up database 'tempdb'.                                 |
| Clearing tempdb database.                                      |
| File activation failure. The physical file name "D:\AdventureW |
| FileMgr::StartLogFiles: Operating system error 2(The system c  |
| Error: 17207, Severity: 16, State: 1.                          |
| Starting up database 'model'.                                  |

- d) Close the Log File Viewer window and return to the **Query Editor** pane.
4. Alter the AdventureWorksLT2012 database by changing its state from Emergency to SINGLE USER.

- a) In the ALTER DATABASE statement, change the SET EMERGENCY keyword to SET SINGLE USER WITH ROLLBACK IMMEDIATE. This keyword instructs the server to switch the database to single user mode and rollback to the previous working instance.

```
ALTER DATABASE AdventureWorksLT2012 SET SINGLE_USER
WITH ROLLBACK IMMEDIATE
```

- b) Execute the ALTER DATABASE statement.  
 c) In the **Object Explorer** pane, right-click **AdventureWorksLT2012** and from the fly-out pane, select **Refresh**. Next to the database name, observe that the status of the database will be still in **Emergency mode**.  
 d) Close the **Query Editor** pane without saving the query.



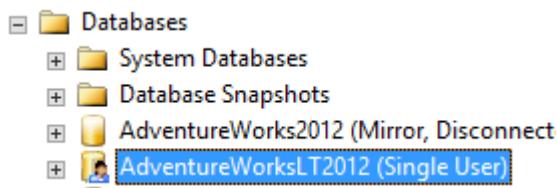
**Note:** Ensure that you close the Query Editor pane before proceeding to the next step.

- e) Open a new **Query Editor** pane and enter the DBCC CHECKDB command followed by the database name AdventureWorksLT2012.  
 f) Type the REPAIR\_ALLOW\_DATA\_LOSS keyword to repair and recover the damaged database.  
 g) On the **SQL Editor** toolbar, select **Execute**.  
 h) On the **Messages** tab, observe that a message stating 'File activation failure. The physical file name "c:\Drive2\AdventureWorksLT2012\_log.ldf" may be incorrect. Failed to restart the current database. The current database is switched to master.' is displayed.

```
DBCC CheckDB ('AdventureWorksLT2012', REPAIR_ALLOW_DATA LOSS);

File activation failure. The physical file name
"D:\AdventureWorksLT2012\Database\AdventureWorksLT2012_log.ldf" may be :
Warning: The log for database 'AdventureWorksLT2012' has been rebuilt.
Transactional consistency has been lost.
The RESTORE chain was broken, and the server no longer has context
on the previous log files, so you will need to know what they were.
You should run DBCC CHECKDB to validate physical consistency.
The database has been put in dbo-only mode.
When you are ready to make the database available for use,
you will need to reset database options and delete any extra log files.
DBCC results for 'AdventureWorksLT2012'.
```

- i) In the **Object Explorer** pane, right-click **AdventureWorksLT2012** and from the fly-out pane, select **Refresh**. Next to the database name, observe that the status of the database is changed to **Single User** mode.



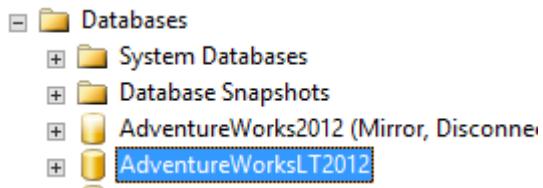
## 5. Recover the AdventureWorksLT2012 database.

- a) Right-click **AdventureWorksLT2012**, and from the fly-out pane, select **Properties**.  
 b) In the Database Properties window, in the **Select a page** section, select the **Options** page.

- c) In the **Other Options** section, scroll down to the bottom and in the **State** section, in the **Restrict Access** field change the setting from **SINGLE\_USER** to **MULTI\_USER**.

| State              |            |
|--------------------|------------|
| Database Read-Only | False      |
| Database State     | NORMAL     |
| Encryption Enabled | False      |
| Restrict Access    | MULTI_USER |

- d) In the Database Properties window, select **OK** to save the settings to the database. Select **Yes** to confirm the closure of all active connections.  
e) Observe that the **AdventureWorksLT2012** database is recovered and displayed in the **Databases** folder.



- f) Close the **Query Editor** pane without saving the query.

## Summary

In this lesson, you have identified and addressed database related issues that occur during data transaction. In addition, you have also performed a database recovery, which enables you to recover your data from corrupted database and hard disk failures.



Use the review questions provided to generate discussion among the participants.

### Which DBCC command would you use often? Why?

**A:** Answers will vary, but may include: CHECKDB, CHECKALLOC, CHECKTABLE, and CHECKCATALOG. These commands are simple, easy to implement, and can be extremely effective when working with large volumes of organizational data.

### Why do you think it's a good practice to backup your data periodically?

**A:** Answers will vary, but may include: it is recommended to backup your database data, because in case of system failures and database corruption you can easily restore your working backup at any point in time, thereby ensuring data protection and reducing data loss.



**Note:** Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.

13

# Performing Advanced Database Management Tasks

**Lesson Time:** 2 hours, 45 minutes

## Lesson Objectives

In this lesson, you will:

- Manage database infrastructure.
- Configure SSIS.

## Lesson Introduction

You have addressed database issues by tracing and recovering databases to optimize the server performance. However, as a database administrator, you might need to troubleshoot issues for which you will use advanced management features and tasks available in SQL Server® 2012. Familiarizing yourself with advanced database management tasks will enable you to resolve issues in your environment and ensure that your databases deliver optimum performance. In this lesson, you will perform advanced database management tasks.

# TOPIC A

## Manage Database Infrastructure

As SQL Server administrator, you need to manage, secure, and maintain information assets on your network. To enhance the performance of the server and maintain it in an up-to-date state, you need to perform several database management tasks such as virtualization, patch management, reporting database configuration, and data-tier application creation. By performing these tasks you can ensure that the database infrastructure is updated periodically and delivers the required services to clients. In this topic, you will manage the database infrastructure.



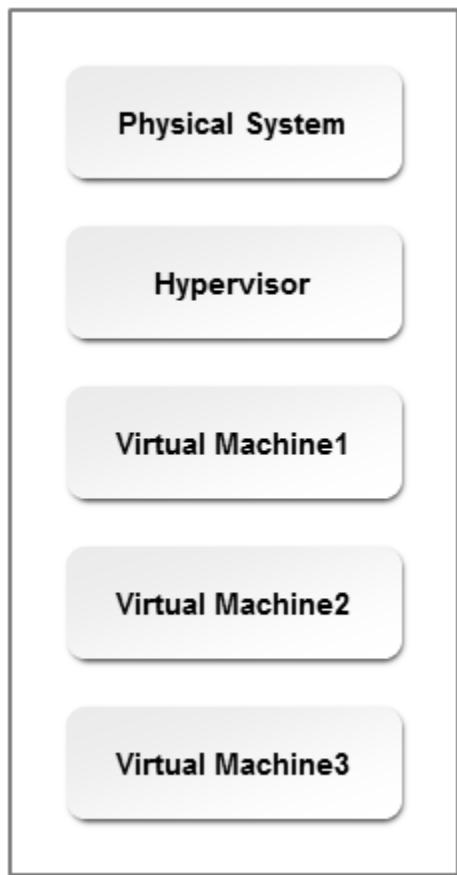
SQL Server  
Virtualization

### SQL Server Virtualization

*Virtualization* is a technology that is used to create and manage workloads in virtual machines. This technology enables you to replace multiple physical machines with virtual machines. SQL Server virtualization increases the availability, performance, and scalability of SQL Server databases.

In most organizations and cloud-based scenarios, SQL Server has the ability to run on virtual systems without encountering issues. Earlier versions of SQL Server are heavily dependent on system I/O performance. When the storage subsystem is virtualized, the performance of those versions degrades further. Unlike earlier versions, SQL Server 2012 is robust and stable when it runs on virtual systems.

In SQL Server 2012, native virtualization I/O performance is replaced by latest versions of hypervisor-based virtualization products, such as Microsoft Hyper-V and VMWare vSphere. By working together with Hyper-V, SQL Server 2012 takes advantage of virtualization and delivers results in an extremely robust and performance capable mode within a virtual environment.



**Figure 13-1: Three virtual machines managed by a hypervisor on a single physical machine.**

## Hyper-Based Virtualization Products

A *Hypervisor* is an application that manages multiple virtual machines on a single physical machine. The physical machine on which the hypervisor runs is called the host machine and the virtual machines are called guest machines. Microsoft Hyper-V and VMWare vSphere are two popular hypervisor-based virtualization products.

## Patches

*Patches* are updates to an application that resolve bugs or issues identified in the application after its release. SQL Server patch management enables you to apply security updates, critical updates, and service packs on SQL Server and the VM operating systems that are part of your virtual infrastructure. There are four types of patches.



Ask the students whether they have encountered situations in which they needed to use many number of physical machines. Discuss how virtualization helps reduce the need for multiple physical machines.

| Patch    | Description                                                                                                                                                                                                                                                                                                                                                              |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hotfixes | Fixes that address a critical bug or security issue. Hotfixes are important fixes that address a specific customer situation, are only available through a support relationship with Microsoft, and may not be distributed outside the customer's organization without written legal consent from Microsoft. They are also known as Quick Fix Engineering (QFE) updates. |
| Updates  | Fixes that update the application with new features, fix a specific problem, or address a non-security related bug.                                                                                                                                                                                                                                                      |

| <b>Patch</b>       | <b>Description</b>                                                                                                                                                         |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cumulative Updates | Fixes and updates released as a bundle to perform hotfixes on a product periodically.                                                                                      |
| Service Packs      | Fixes and updates that include a large number of enhancements and patches. Service packs may also include new features that are part of the latest version of the product. |

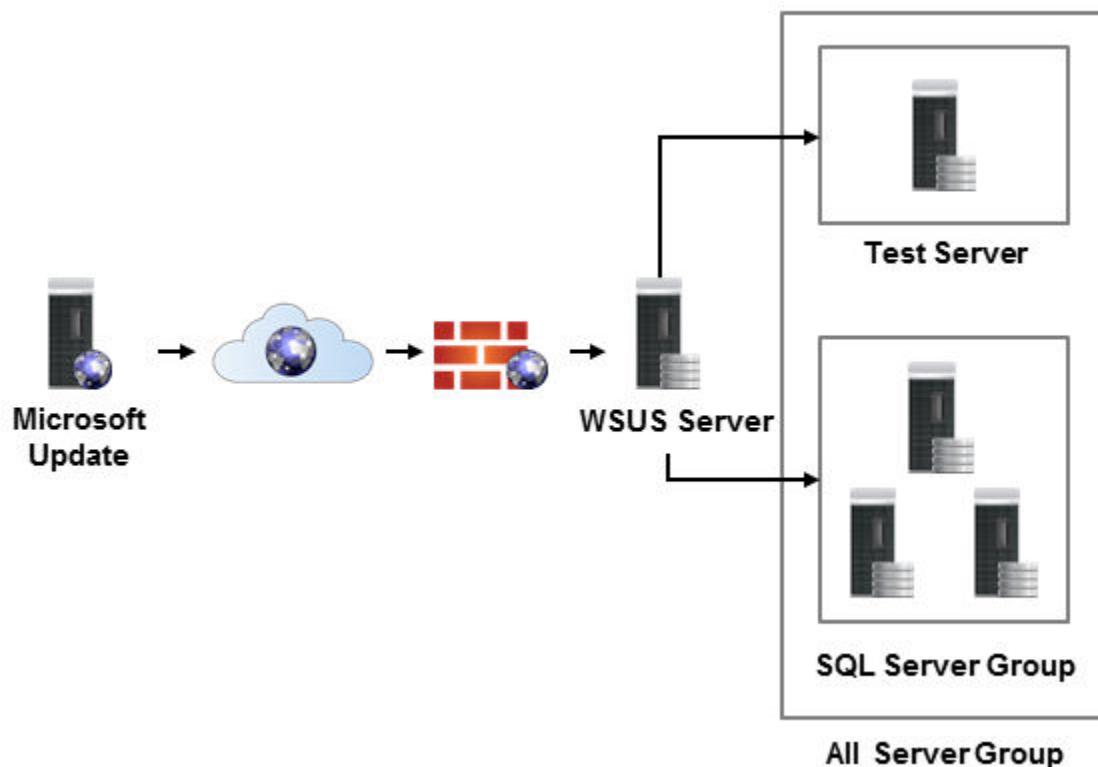
## SQL Server Patch Management



SQL Server Patch Management

Patch management refers to the way you apply patches to a product. Patches can be applied using different methods as listed in the table.

| <b>Methods to Apply Patches</b> | <b>In this Method</b>                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Microsoft Update                | You can set the operating system to automatically download the patches for the products installed on a machine from Microsoft's update website and apply those patches. This method is best suited for personal computers.                                                                                                                                                                                                                            |
| Manual update                   | You can manually download and apply a patch on each individual SQL Server system. By using manual updates, you can easily identify exceptional conditions and address them quickly. This method is best suited for mission critical systems.                                                                                                                                                                                                          |
| Scripted install                | You can create a remote deployment script that copies a patch automatically to the system, runs it using command line parameters, and maintains a log of the patches installed. This method is suited for updating systems in non-production environments.                                                                                                                                                                                            |
| Automated deployment            | You can use Microsoft Windows Server Update Services (WSUS) for advanced patch management. By using WSUS, you can automate many patch management tasks such as testing the patches on test machines, deploying tested and approved patches to systems, creating exceptions and rules in the deployment, and centralizing the patch download repository.                                                                                               |
| Slipstreamed setup              | <p>You can integrate the base installation files for an operating system with its patches. The resultant file enables you to install the updated software in a single step.</p> <p><i>Slipstreaming</i> enables you to update the original installation media with the latest patches. Once the installation media is updated with the latest patches, you can use it for new installations without running the patches on the new installations.</p> |



**Figure 13-2: Patch management using Microsoft Update and WSUS.**

|  |                                                                                                                                                                                                                                    |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Note:</b> Though Microsoft updates can be applied on personal computers, they are not recommended for enterprise computers because an administrator does not have control over the automatic updates and their appropriateness. |
|  | <b>Note:</b> Microsoft provides pre-slipstreamed packages for many of its products. If you have enabled the required licenses, you can obtain these packages from Microsoft.                                                       |

## Microsoft WSUS

*Microsoft WSUS* is a component that manages the download and installation of updates and patches on Windows operating systems. Using a WSUS server, administrators can automate updates. Once the Automatic Updates feature is enabled, the machines can receive Microsoft Updates whenever an update is available.

## Guidelines to Implementing Patch Management

To manage patches efficiently, you need to select the patch management method that suits your environment. Some of the guidelines to implement patch management are:

- Testing patches on environment machines that have the same setup as the production systems. The test systems must have a copy of the database and the other objects that are available on the production systems.
- Applying patches released by Microsoft on the test systems and checking whether all functions such as replication, backups, client applications, and user logins work correctly. If some function or component does not work as expected, check the patch management documentation to determine the issue.

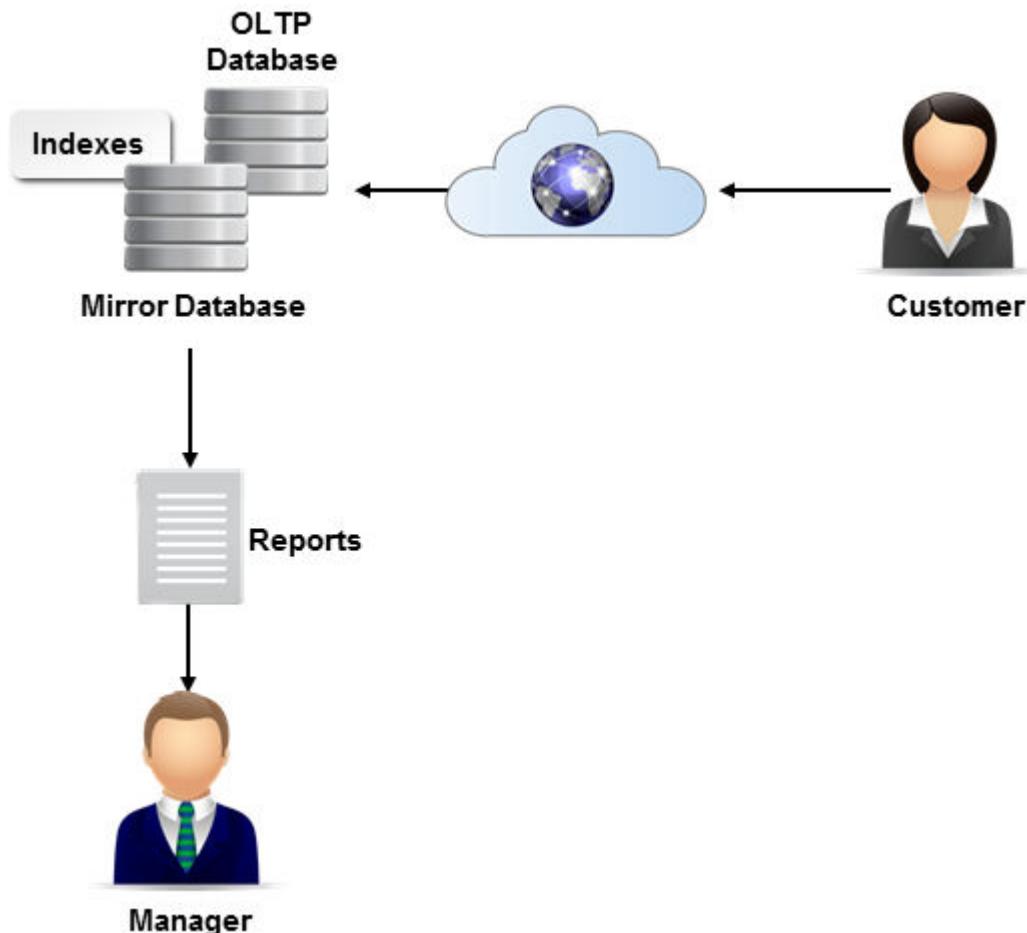
- Looking for workarounds and trying them as mentioned in the patch management documentation. You can test whether the workarounds resolve the issue by deploying them on test systems. After confirming that the problem is resolved, deploy the patch with the workaround on the actual production systems.
- Using a combination of patch management techniques to ensure that SQL Server systems are always up-to-date. By applying periodic updates, you can prevent known issues and security concerns from affecting your systems.

## The Reporting Database Infrastructure Configuration Process



Data structures in RDBMSs are optimized for transaction processing rather than querying. During transaction processing, normalization and quick retrieval techniques help you insert rows into tables and quickly identify rows for updating or deletion. While generating reports from the data stored in a database, indexing is the only major optimization method that enables the queries in the reports to run faster.

In an Online Transaction Processing System (OLTP), each change in database tables results in extensive write I/O operations leading to performance degradation. To enhance the performance of the databases, indexes need to be turned off. Therefore, it is preferable to create reports from a data warehouse rather than an RDBMS. However, in some scenarios, you may have to use the RDBMS database for reporting. In such cases, you can create a database mirror to act as a standby and use the mirror with indexes for reporting.



**Figure 13-3:** A reporting database infrastructure that includes an OLTP database, a mirror database, and indexes.

## Data-Tier Applications

A *Data-tier application* is a self-contained SQL Server entity that defines the database objects and instance objects associated with a database. A database administrator can create a DAC package of SQL Server objects that can be deployed across multiple machines. When an application developer modifies the structure of the database, the package is regenerated with a new version, the changes are synchronized with the original database, and the new version is distributed to the other machines.

While you can use a .DAC package to distribute the data structure, you can use a .BAC package to distribute both the data structure and the data in the database objects.



Data-Tier Applications

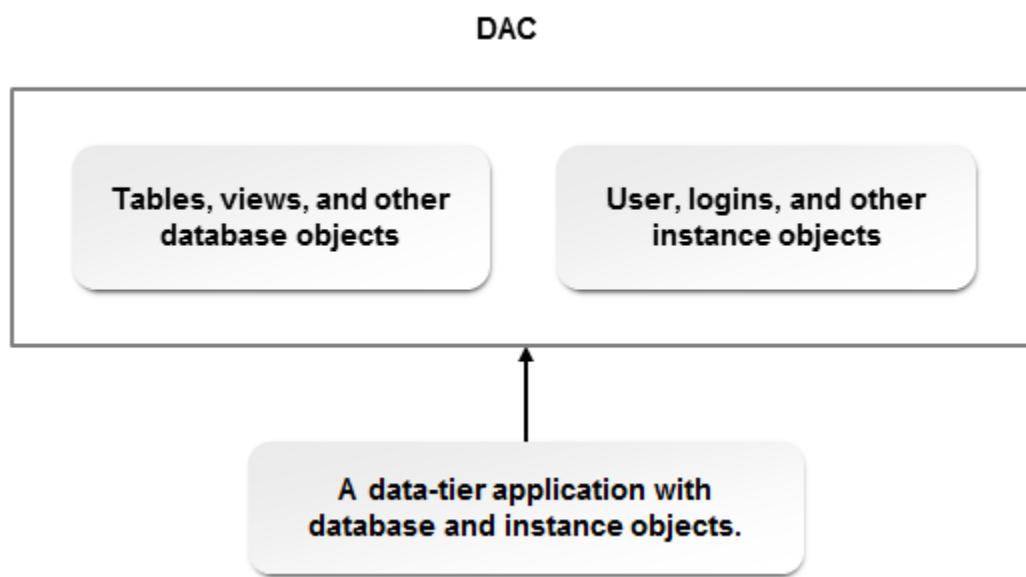


Figure 13-4: A data-tier application consisting of database objects and instance objects.

## Operations on Data-Tier Applications

In SQL Server 2012, you can perform several operations to control the .DAC and .BAC packages as listed in the table.

| Operation  | Used to                                                                                                                                                                                                                          |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Register   | Mark a database as available for data-tier applications and other operations.                                                                                                                                                    |
| Unregister | Mark a database as unavailable for data-tier applications.                                                                                                                                                                       |
| Extract    | Create a .DACPAC package that stores the structure of a database. The extract operation assigns a name and version number to the package and stores it at the specified location.                                                |
| Deploy     | Set up the database on an SQL Server instance by using the .DACPAC package.                                                                                                                                                      |
| Upgrade    | Synchronize the changes in the database objects to the original database by using the .DACPAC package. The version number of the package is used to ensure that upgrade operation does not update changes from an older version. |



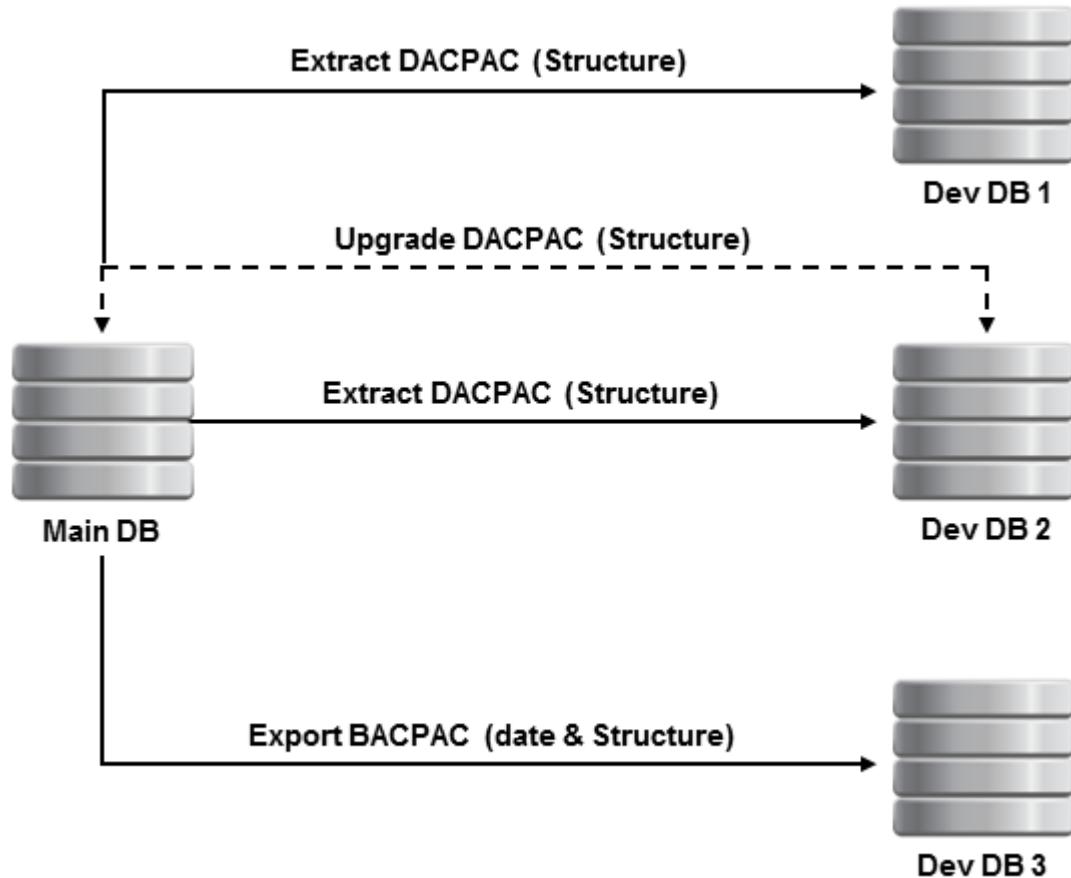
Ask the students whether they have encountered the need to install a database on multiple user machines. Discuss the benefits of using a data-tier application to perform this task easily.



Operations on Data-Tier Applications

| <b>Operation</b> | <b>Used to</b>                                                                                                                               |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Export           | Create a .BACPAC package that stores the database structure and the data in the database objects.                                            |
| Import           | Import the database structure along with the data into a new or existing database. This operation uses a previously created .BACPAC package. |

While performing an upgrade of the data structure, you can configure a number of options depending on the actions you wish to perform in case of an error. For example, you can ignore errors and conflicts and continue with the rest of the changes or fail and rollback the entire set of changes at the first occurrence of an error. You can also specify whether to continue if potential data loss is possible.



*Figure 13-5: Actions to control the .dcpac and .bacpac packages.*



Ask the students to identify situations on their day-to-day jobs that require upgrade operations. Ask them the volume of changes they typically perform on the databases and how frequently they would need to perform the upgrade operation.

## Need for a Data-Tier Application

Databases in an organizational environment do not work as standalone systems. Instead of accessing a database directly, users rely on client applications to communicate with the database at the backend. Client applications are created in a development platform and they communicate with databases using standard protocols such as the SQL Native client, ODBC, or OLEDB.

However, while developing or maintaining an application, a developer may perform structural changes on the databases. Usually, developers run a local copy of a database and develop code with it. When a database or object structure is modified on the local copy, it is important to ensure that the structural changes are synchronized with the original database. The structural changes must be

distributed to all the application developers working on the same database. Data-tier applications help distribute copies of the database structure and synchronize the structural changes across multiple systems.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on How to Manage the Database Infrastructure

# ACTIVITY 13-1

## Extracting a Data-Tier Application

### Before You Begin

1. Disconnect INSTANCE01 and connect to SQL01 default instance.
2. In the **Server Properties**→**Database Settings** page, set the **Default index fill factor** value to '0'.

### Scenario

Management of Adventure Works Cycles has hired a number of application developers to create applications that address the day-to-day querying requirements of the organization. Before the developers start creating applications, you want to create a new database to store the details of employees. As the developers need to work on the employee database from geographically dispersed locations, you need to provide them copies of the database structure that they can deploy on their local machines.

1. Create a new database and add a table to it.
  - a) In the **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
  - b) In the **Object Explorer** pane, in the SQL01 default server instance, right-click the **Databases** folder and select **New Database**.
  - c) In the **New Database** dialog box, in the **Database name** text box, type **DepartmentDB** and select **OK**.
  - d) In the **Object Explorer** pane, expand **Databases** folder and select the **DepartmentDB** database.
  - e) Open a **Query Editor** pane and type the CREATE TABLE statement to create a table named **Departments** with the **DepartmentID** column with integer data type and the NOT NULL constraint.

```
CREATE TABLE Departments
(
 DepartmentID int IDENTITY(1,1) NOT NULL,
```

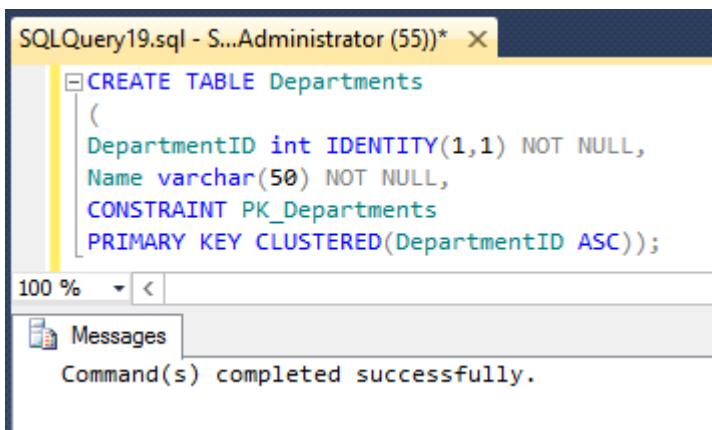
- f) Enter another column named **Name** of varchar data type with a maximum character length of 50 characters and the NOT NULL constraint.

```
CREATE TABLE Departments
(
 DepartmentID int IDENTITY(1,1) NOT NULL,
 Name varchar(50) NOT NULL,
```

- g) Add the PRIMARY KEY CLUSTERED constraint on the sorted **DepartmentID** column.

```
CREATE TABLE Departments
(
 DepartmentID int IDENTITY(1,1) NOT NULL,
 Name varchar(50) NOT NULL,
 CONSTRAINT PK_Departments PRIMARY KEY CLUSTERED(DepartmentID ASC));
```

- h) Execute the CREATE TABLE statement.



```
CREATE TABLE Departments
(
 DepartmentID int IDENTITY(1,1) NOT NULL,
 Name varchar(50) NOT NULL,
 CONSTRAINT PK_Departments
 PRIMARY KEY CLUSTERED(DepartmentID ASC));

```

100 % < Messages  
Command(s) completed successfully.

- i) Close the **Query Editor** pane without saving the query.
2. Create a table to store the details of employees and add foreign keys to relate it to the departments table.

- a) Open a **Query Editor** pane and type the CREATE TABLE statement to create a table named Employees.

```
CREATE TABLE Employees
```

- b) Enter the following columns or fields that need to be created along with the table with the specified data types.

- EmployeeID column with integer data type, IDENTITY value of 1,1, and NOT NULL constraint
- DepartmentID column with integer data type and NOT NULL constraint
- Fullname column with varchar data type with a maximum character length of 50 characters and NOT NULL constraint
- DoB column with date data type and NOT NULL constraint
- Gender column with char data type and NOT NULL constraint
- TotalSaleAmount column with float data type and NOT NULL constraint
- Job Title column with varchar data type with a maximum character length of 50 characters and NOT NULL constraint
- Active column with bit data type and NOT NULL constraint



**Note:** You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```
CREATE TABLE Employees
(
EmployeeID int IDENTITY(1,1) NOT NULL,
DepartmentID int NOT NULL,
Fullname varchar(50) NOT NULL,
DoB date NOT NULL,
Gender char(1) NOT NULL,
JobTitle varchar(50) NOT NULL,
Active bit NOT NULL,
```

- c) Enter the CONSTRAINT keyword followed by the constraint name PK\_Employees along with the PRIMARY KEY CLUSTERED keyword.

```
"...
DoB date NOT NULL,
Gender char(1) NOT NULL,
JobTitle varchar(50) NOT NULL,
Active bit NOT NULL,
CONSTRAINT PK_Employees PRIMARY KEY CLUSTERED
```

- d) Enter the EmployeeID column with the ASC keyword to arrange the data in ascending order.



Check with participants to ensure that they were able to create the **Departments** table without any problems.

```
"...
Gender char(1) NOT NULL,
JobTitle varchar(50) NOT NULL,
Active bit NOT NULL,
CONSTRAINT PK_Employees PRIMARY KEY CLUSTERED
(EmployeeID ASC));
```

- e) Execute the query to create the table.
- f) Close the **Query Editor** pane without saving the query.



Check with participants to ensure that they were able create the **Employees** table without any problems.

3. Using foreign keys, create a relationship between the Departments and the Employees table.

- a) Open a **Query Editor** pane and type the ALTER TABLE statement.

```
ALTER TABLE Employees
```

- b) Type the WITH CHECK ADD CONSTRAINT keywords to add a constraint to the Employees table.

```
ALTER TABLE Employees WITH CHECK ADD CONSTRAINT
```

- c) Enter the constraint name FK\_Employees\_Departments along with the FOREIGN KEY keyword and the DepartmentID column name to create a foreign key on the DepartmentID column.

```
ALTER TABLE Employees WITH CHECK ADD CONSTRAINT
FK_Employees_Departments FOREIGN KEY(DepartmentID)
REFERENCES Departments(DepartmentID);
```

- d) Enter the REFERENCES keyword followed by the Departments name and the DepartmentID column to reference the DepartmentID column in the Departments table.

```
ALTER TABLE Employees WITH CHECK ADD CONSTRAINT
FK_Employees_Departments FOREIGN KEY(DepartmentID)
REFERENCES Departments(DepartmentID);
```

- e) Enter the ALTER TABLE statement followed by the Employees table name and the CHECK CONSTRAINT statement to check the constraint added to the Employees table.

```
"...
FK_Employees_Departments FOREIGN KEY(DepartmentID)
REFERENCES Departments(DepartmentID);
ALTER TABLE Employees CHECK CONSTRAINT
```

- f) Enter FK\_Employees\_Departments to specify the name of the constraint to be checked.

```
"...
REFERENCES Departments(DepartmentID);
ALTER TABLE Employees CHECK CONSTRAINT
FK_Employees_Departments;
```

- g) Execute the query to create the foreign key.
- h) Close the **Query Editor** pane without saving the query.

4. Insert rows into the Departments and Employees tables.

- a) Open a **Query Editor** pane and type the INSERT INTO statement followed by the table name Departments.

```
INSERT INTO Departments
```

- b) Type the field name Name into which you want to insert values.

```
INSERT INTO Departments (Name)
```

- c) Enter the Values keyword followed by the values that need to be inserted into the Departments table.



**Note:** You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```
INSERT INTO Departments (Name)
VALUES
('Human Resources'),
('IT and Support'),
```

```
('Development'),
('Marketing'),
('Operations');
```

- d) Execute the query.

```
SQLQuery22.sql - S...Administrator (54)* X SQLQuery21.s
INSERT INTO Departments (Name)
VALUES
('Human Resources'),
('IT and Support'),
('Development'),
('Marketing'),
('Operations');

100 % < |
Messages | (5 row(s) affected)
```

- e) Close the **Query Editor** pane without saving the query.  
f) Open a Query Editor pane and type the INSERT INTO statement followed by the table name Departments.

```
INSERT INTO Employees
```

- g) Enter the field names DepartmentID, FullName, DoB, Gender, JobTitle, and Active into which you want to insert values.

```
INSERT INTO Employees (DepartmentID, FullName, DoB,
Gender, JobTitle, Active)
```

- h) Enter the Values keyword followed by the values that need to be inserted into the Employees table.



**Note:** You can either type the following code snippet or to save time, you can copy and paste the code snippet provided to you in a TXT file located in C:\093108Data\Code.txt.

```
INSERT INTO Employees (DepartmentID, FullName, DoB, Gender, JobTitle,
Active)
VALUES
(1, 'John Smith', '1975-03-12', 'M', 'R Manager', 1),
(2, 'Jane Doe', '1965-04-22', 'F', 'Tech Lead', 0),
(3, 'Jack Bauer', '1972-10-23', 'M', 'Senior Developer', 1),
(4, 'Clint McGee', '1982-04-15', 'M', 'Marketing Manager', 1),
(5, 'Sara Johnson', '1985-12-22', 'F', 'Operations Executive', 0),
(2, 'Bill Shaw', '1972-02-14', 'M', 'DBA', 1),
(3, 'Dorothy Samson', '1982-02-28', 'F', 'Designer', 1),
(5, 'Chloe o''Brian', '1989-07-01', 'F', 'Technical Operator', 1);
```

- i) Execute the query to insert the values into the table.

```

SQLQuery23.sql - S...Administrator (57)* X SQLQuery22.sql - S...Administrator (54))*

INSERT INTO Employees (DepartmentID, FullName, Dob,

Gender, JobTitle, Active)

VALUES

(1, 'John Smith', '1975-03-12', 'M', 'R Manager', 1),

(2, 'Jane Doe', '1965-04-22', 'F', 'Tech Lead', 0),

(3, 'Jack Bauer', '1972-10-23', 'M', 'Senior Developer', 1),

(4, 'Clint McGee', '1982-04-15', 'M', 'Marketing Manager', 1),

(5, 'Sara Johnson', '1985-12-22', 'F', 'Operations Executive', 0),

(2, 'Bill Shaw', '1972-02-14', 'M', 'DBA', 1),

(3, 'Dorothy Samson', '1982-02-28', 'F', 'Designer', 1),

(5, 'Chloe o''Brian', '1989-07-01', 'F', 'Technical Operator', 1);

```

100 % < III

Messages

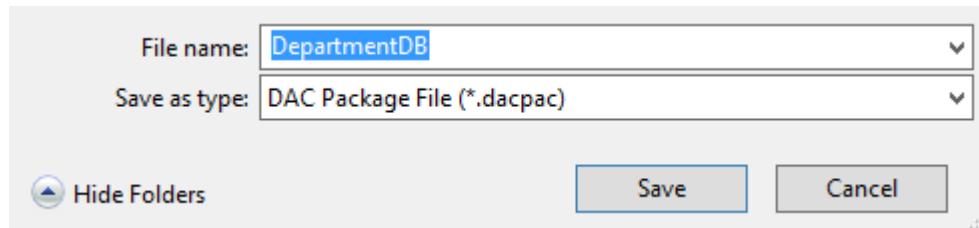
(8 row(s) affected)



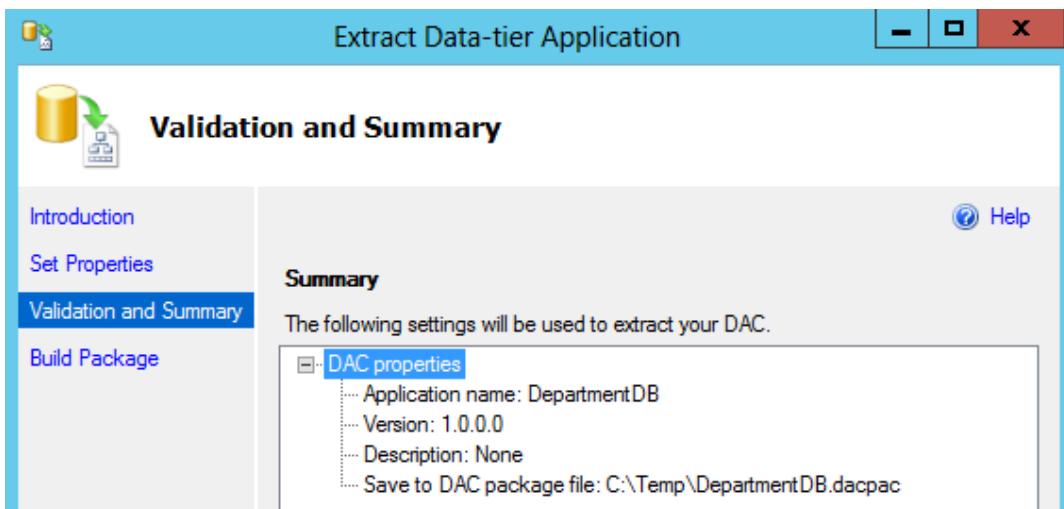
Check with participants to ensure that they have added rows into the Employees and Departments table.

5. Create a data-tier application.

- In the **Object Explorer** pane, right-click the **DepartmentDB** database and select **Tasks→Extract Data-tier Application**.
- In the **Extract Data-tier Application** wizard, on the **Introduction** page, select **Next**.
- On the **Set Properties** page, verify that the **Application name** text box displays **DepartmentDB**.
- Select **Browse** to the right of the **Save to DAC package file** text box.
- In the **Save As** dialog box, navigate to the **C:\Temp** folder.
- In the **Filename** text box, verify that **DepartmentDB** is displayed and select **Save**.

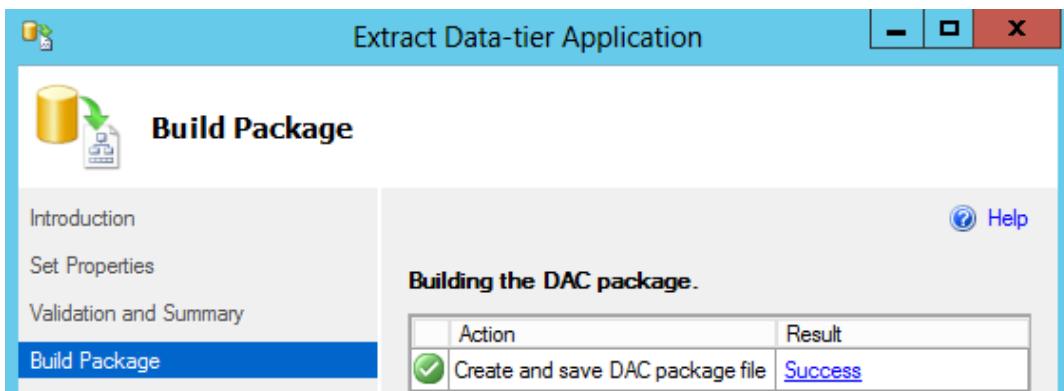


- On the **Set Properties** page, select **Next**.
- On the **Validation and Summary** page, observe the summary of the settings you specified and select **Next**.



**Note:** Once you select **Next**, SQL Server starts building the DAC package. You may have to wait for a few minutes until this process completes.

- i) On the **Build Package** page, verify that the package has been built successfully and select **Finish**.



- j) Close the **Query Editor** panes without saving the queries.

# ACTIVITY 13-2

## Deploying a Data-Tier Application Package

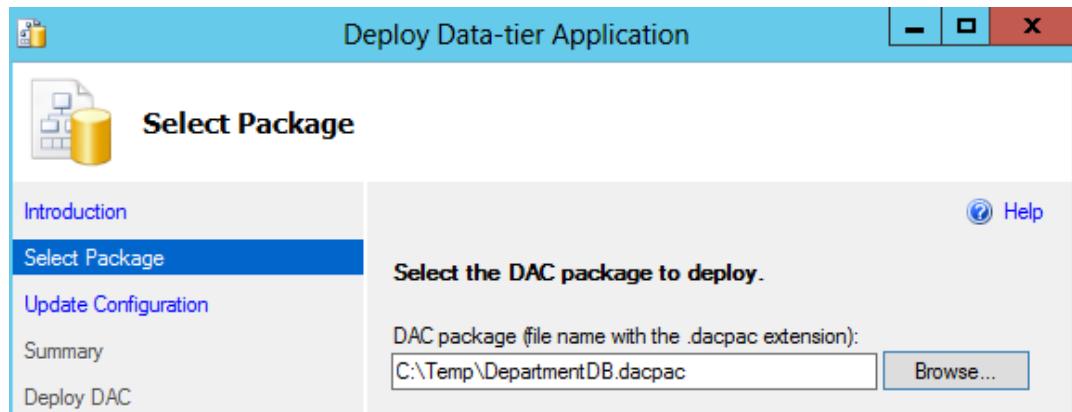
### Scenario

The HR team of Adventure Works Cycles wants to develop an application to track employee performance. To do this, you want to set up the DepartmentDB database on the SQL01\INSTANCE01 instance so that the application developer working on that instance can start using the database. Because you have already created a data-tier application for the DepartmentDB database, you want to deploy it on another instance of SQL Server.

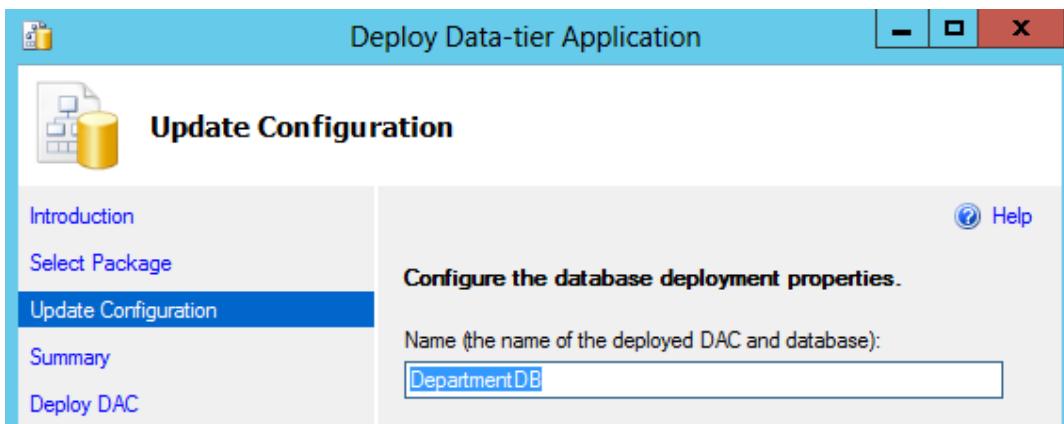
1. Launch the Deploy Data-tier Application wizard.
  - a) In the **Object Explorer** pane, right-click **SQL01** and select **Disconnect**.
  - b) In the **Object Explorer** pane, select **Connect** and from the menu that appears, select **Database Engine**.
  - c) In the **Connect to Server** dialog box, from the **Server name** drop-down list, select **SQL01\INSTANCE01** and then select **Connect**.
  - d) In the **Object Explorer** pane, in the **SQL01\INSTANCE01** server instance, expand **Databases**.
  - e) Right-click the **Databases** folder and select **Deploy Data-tier Application**.
2. Deploy the DAC package.
  - a) In the **Deploy Data-tier Application** wizard, on the **Introduction** page, select **Next**.
  - b) On the **Select Package** page, select **Browse**.
  - c) In the **Open** dialog box, navigate to the **C:\Temp** folder, select the **DepartmentDB.dacpac** file and select **Open**.



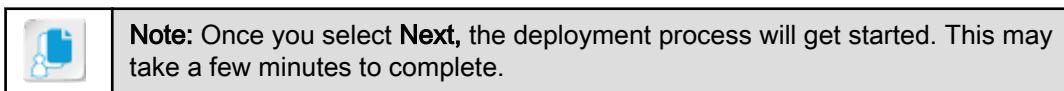
Check with participants to ensure that they have selected the **Databases** on the **SQL01\INSTANCE01** server instance.



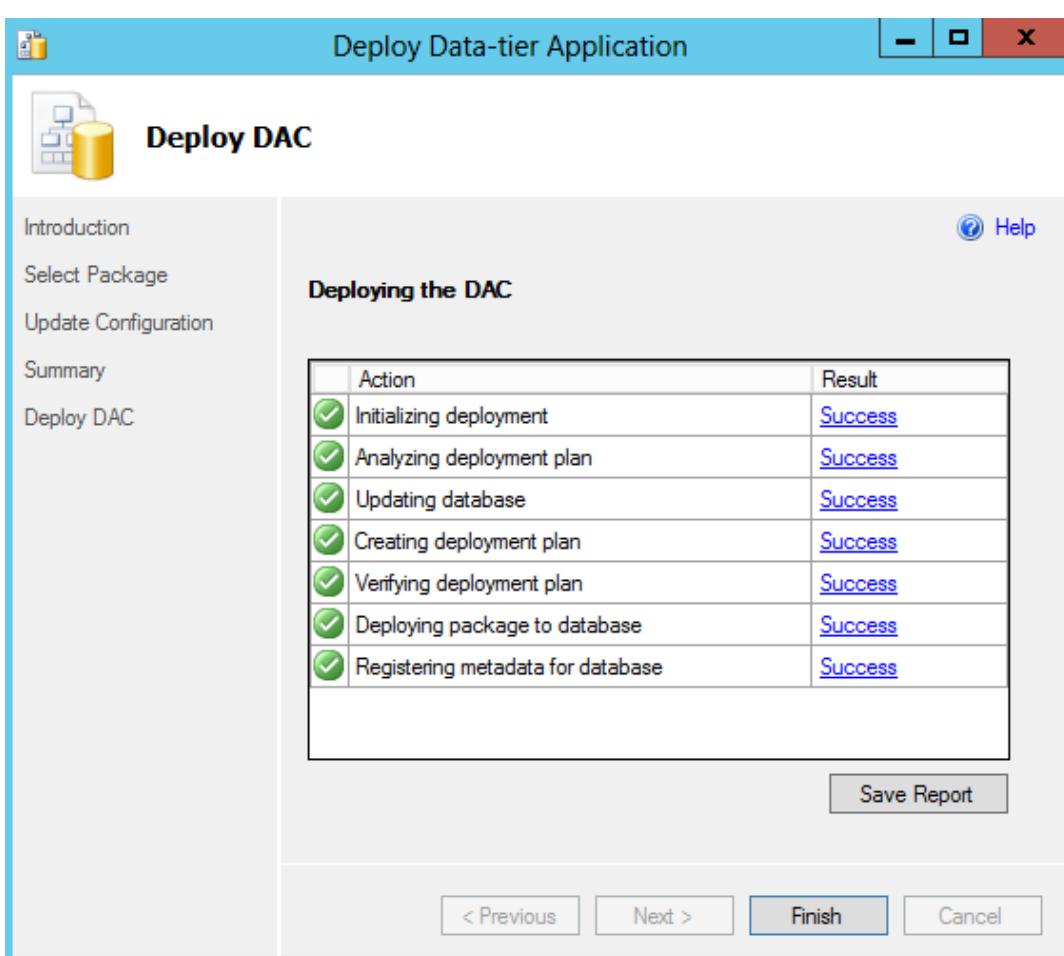
- d) On the **Select Package** page, select **Next**.
- e) On the **Update Configuration** page, in the **Name** text box, verify that the name of the DAC package to be deployed is **DepartmentDB** and select **Next**.



- f) On the **Summary** page, verify the summary of the settings you specified and select **Next**.

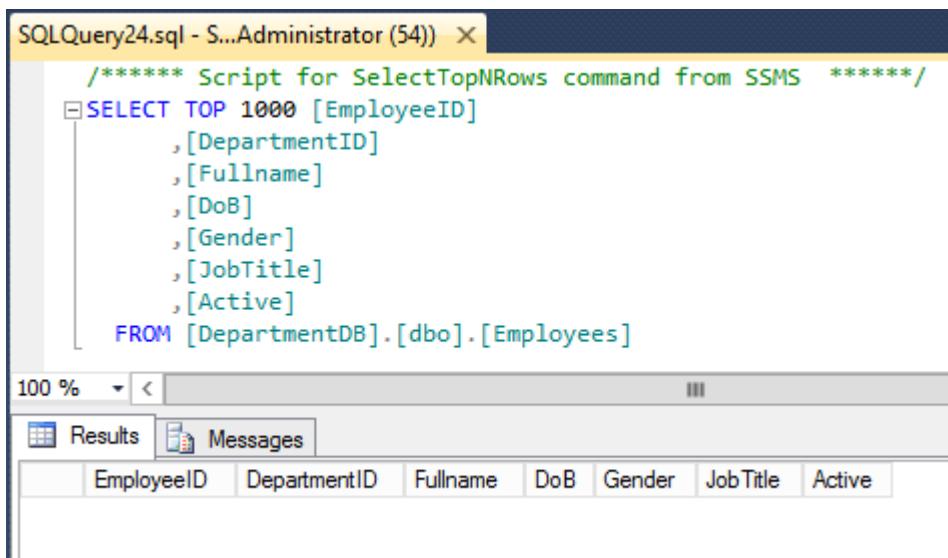


- g) On the **Deploy DAC** page, verify that the deployment has completed successfully and select **Finish**.



3. Verify that the database has been deployed correctly.
  - a) Right-click **Databases** and select **Refresh**.
  - b) Verify that the **DepartmentDB** database is present and expand the **Tables** folder.

- c) Right-click the **dbo.Employees** table and select the **Select Top 1000** rows menu option. Observe that the table does not contain any records and close the **Query Editor** pane without saving.



The screenshot shows the SSMS Query Editor window titled "SQLQuery24.sql - S...Administrator (54)". The query script is as follows:

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [EmployeeID]
 ,[DepartmentID]
 ,[Fullname]
 ,[DoB]
 ,[Gender]
 ,[JobTitle]
 ,[Active]
FROM [DepartmentDB].[dbo].[Employees]
```

The Results tab is selected, displaying a header row for the columns: EmployeeID, DepartmentID, Fullname, DoB, Gender, JobTitle, and Active. The data grid below the header is empty, indicating no results were returned.



Remind participants that while extracting a data-tier application, only the structure of the database is packaged. Therefore, while deploying the package, only the database structure is available and the records are not copied from the source database.

# ACTIVITY 13-3

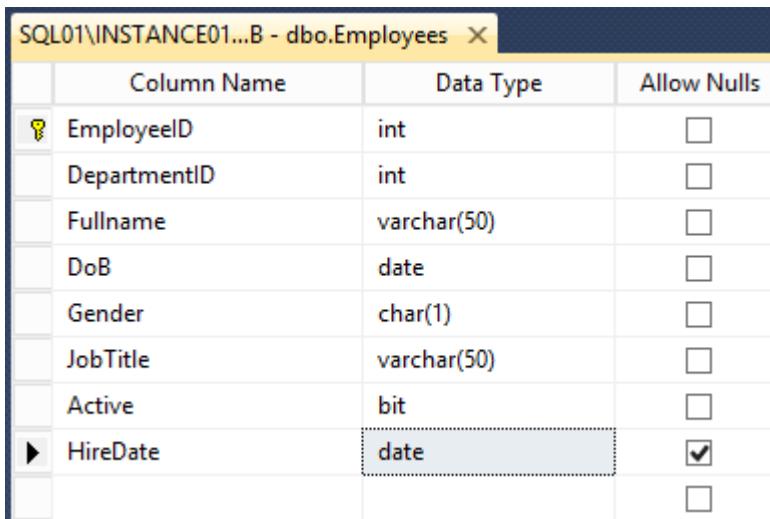
## Upgrading a Data-Tier Application

### Scenario

While creating the application for tracking employee performance, the HR team identifies the need to add the date of hire of each employee. Also, they want to store the details of the projects that employee is associated with. As database administrator at Adventure Works Cycles, you are assigned the task of modifying the structure of the DepartmentDB database to perform these additions. To be able to synchronize the changes with the other server instances on which you deployed the data-tier application, you want to upgrade the DAC package.

#### 1. Modify the structure of the Employees table.

- In the **Object Explorer** pane, in the **SQL01\INSTANCE01** server instance, right-click the **Employees** table and select **Design**.
- In the table design, select the **ColumnName** cell in the last row and type **HireDate**.
- In the **Data Type** column, select **date**.
- In the **Allow Nulls** column, ensure that the check box is checked to allow null values.



The screenshot shows the 'SQL01\INSTANCE01...B - dbo.Employees' table design. The table has eight columns: EmployeeID, DepartmentID, Fullname, DoB, Gender, JobTitle, Active, and HireDate. The 'HireDate' column is highlighted with a dotted border. The 'Allow Nulls' column for 'HireDate' contains a checked checkbox, indicating it allows null values.

|              | Column Name | Data Type                           | Allow Nulls |
|--------------|-------------|-------------------------------------|-------------|
| EmployeeID   | int         | <input type="checkbox"/>            |             |
| DepartmentID | int         | <input type="checkbox"/>            |             |
| Fullname     | varchar(50) | <input type="checkbox"/>            |             |
| DoB          | date        | <input type="checkbox"/>            |             |
| Gender       | char(1)     | <input type="checkbox"/>            |             |
| JobTitle     | varchar(50) | <input type="checkbox"/>            |             |
| Active       | bit         | <input type="checkbox"/>            |             |
| HireDate     | date        | <input checked="" type="checkbox"/> |             |

- On the **Standard** toolbar, select **Save** to save the changes.
- Close the table design pane.

#### 2. Create a new table named Projects.

- Open a **Query Editor** pane and type the CREATE TABLE statement followed by the table name Projects.

```
CREATE TABLE Projects
```

- Enter the following columns or fields that need to be created along with the table with the specified data types.

- ProjectID column with integer data type, IDENTITY value of (1,1) and NOT NULL constraint.
- Project Name column with varchar data type with a maximum character length of 50 characters and NOT NULL constraint.
- Project Owner column with integer data type and NOT NULL constraint.



Check with participants to ensure that they have added the HireDate column to the Employees table

```
CREATE TABLE Projects
(
 ProjectID int IDENTITY(1,1) NOT NULL,
 ProjectName varchar(50) NOT NULL,
 ProjectOwner int NOT NULL,
```

- c) Enter the CONSTRAINT keyword followed by the constraint name PK\_Projects along with the PRIMARY KEY CLUSTERED keyword.

```
"...
ProjectName varchar(50) NOT NULL,
ProjectOwner int NOT NULL,
CONSTRAINT PK_Projects PRIMARY KEY CLUSTERED
```

- d) Enter the ProjectID column with the ASC keyword to arrange the data in ascending order.

```
"...
ProjectName varchar(50) NOT NULL
ProjectOwner int NOT NULL,
CONSTRAINT PK_Projects PRIMARY KEY CLUSTERED
(ProjectID ASC));
```

- e) Execute the query to create the table.

- f) Close the **Query Editor** pane without saving.



Check with participants to ensure that they were able to create the Projects table without any errors.

3. Alter the Projects table to add a foreign key constraint that relates it to the Employees table.

- a) Open a **Query Editor** pane and type the ALTER TABLE statement followed by the Projects table name.

```
ALTER TABLE Projects
```

- b) Enter the WITH CHECK ADD CONSTRAINT keywords to add a constraint to the Projects table.

```
ALTER TABLE Projects WITH CHECK ADD CONSTRAINT
```

- c) Type the constraint name FK\_Projects along with the FOREIGN KEY keyword and the ProjectOwner column name to create a foreign key on the ProjectOwner column.

```
ALTER TABLE Projects WITH CHECK ADD CONSTRAINT
FK_Projects_Employees FOREIGN KEY(ProjectOwner)
```

- d) Enter the REFERENCES keyword followed by the Employees table name and the EmployeeID column to reference the EmployeeID column in the Employees table.

```
ALTER TABLE Projects WITH CHECK ADD CONSTRAINT
FK_Projects_Employees FOREIGN KEY(ProjectOwner)
REFERENCES Employees (EmployeeID);
```

- e) Enter the ALTER TABLE statement followed by the Projects table name and the CHECK CONSTRAINT statement to check the constraint added to the Projects table.

```
"...
FK_Projects_Employees FOREIGN KEY(ProjectOwner)
REFERENCES Employees (EmployeeID);
ALTER TABLE Projects CHECK CONSTRAINT
```

- f) Type FK\_Projects\_Employees to specify the name of the constraint to check the foreign key constraint on the Projects table.

```
"...
REFERENCES Employees (EmployeeID)
ALTER TABLE Projects CHECK CONSTRAINT
FK_Projects_Employees;
```

- g) Execute the query.

- h) Close the **Query Editor** pane without saving the query.

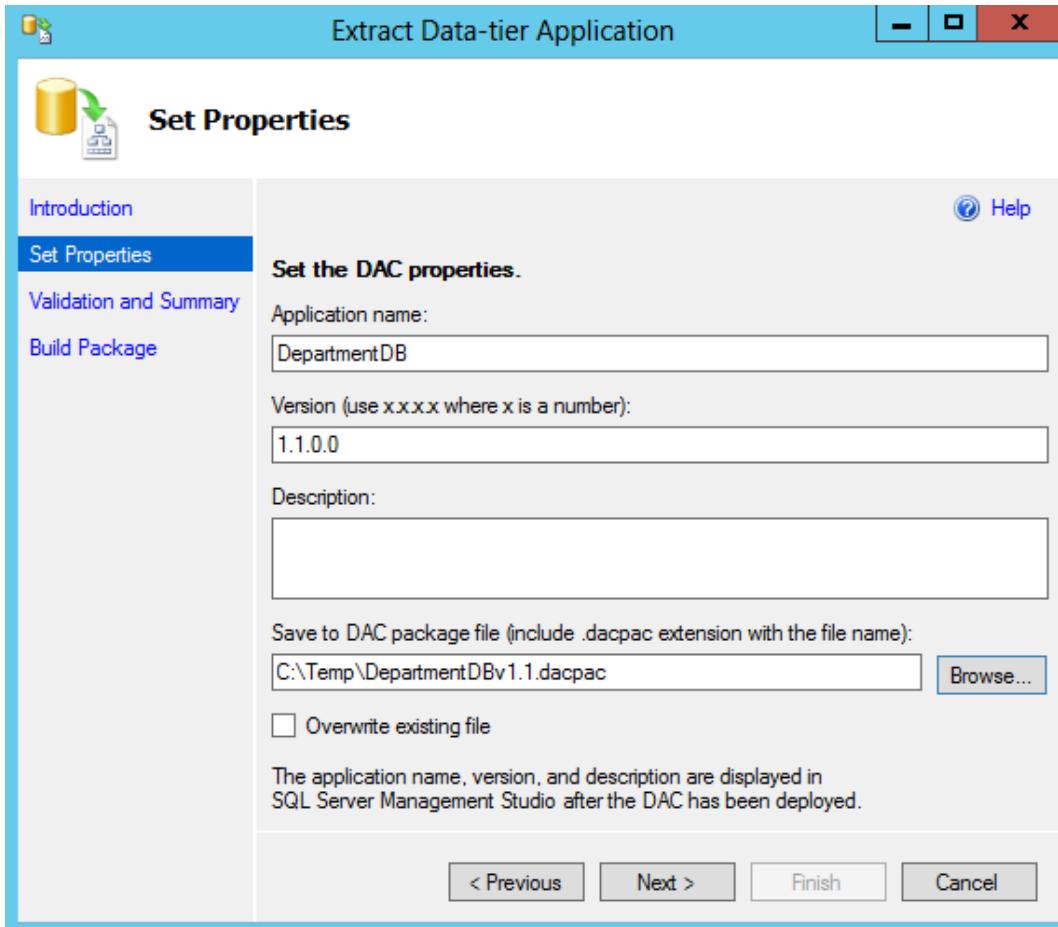


Check with participants to ensure that they were able to create the foreign key without any errors.

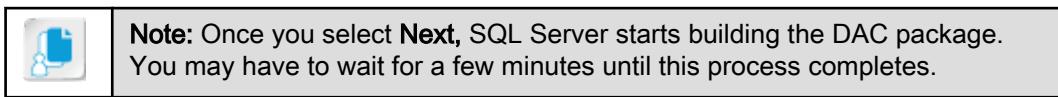
4. Create another version of the data-tier application.

- a) In the **Object Explorer** pane, right-click the **DepartmentDB** database and select **Tasks→Extract Data-tier Application**.

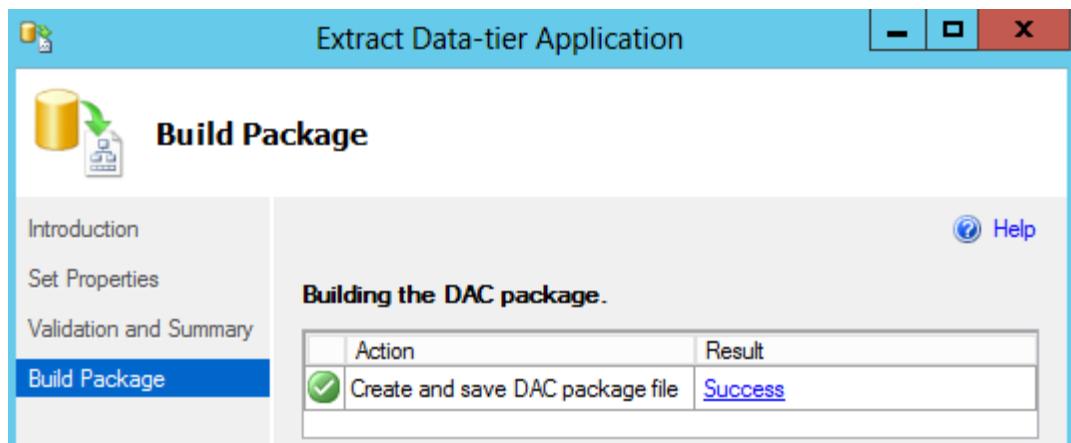
- b) In the Extract Data-tier Application wizard, on the **Introduction** page, select **Next**.
- c) On the **Set Properties** page, verify that the **Application name** text box displays **DepartmentDB**.
- d) In the **Version** text box, type **1.1.0.0**.
- e) Select **Browse** to the right of the **Save to DAC package file** text box.
- f) In the **Save As** dialog box, in the **File name** text box, type **DepartmentDBv1.1** and select **Save**.
- g) On the **Set Properties** page, select **Next**.



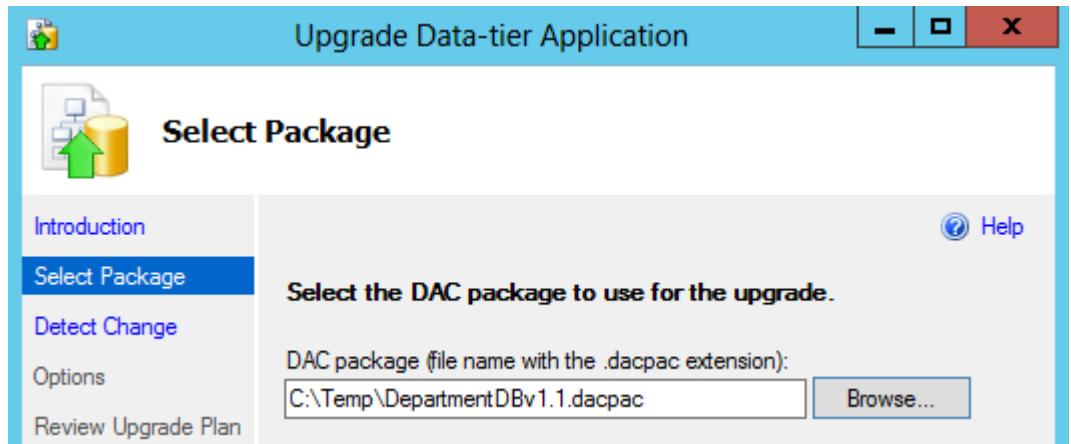
- h) On the **Validation and Summary** page, observe the summary of the settings you specified and select **Next**.



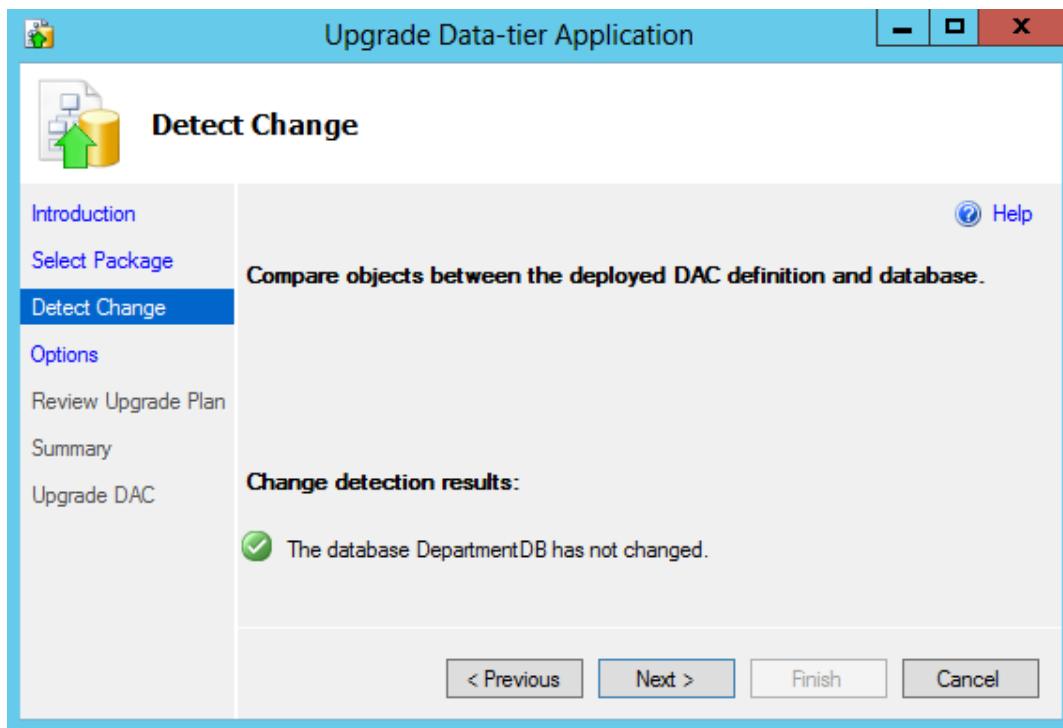
- i) On the **Build Package** page, verify that the package has been built successfully and select **Finish**.



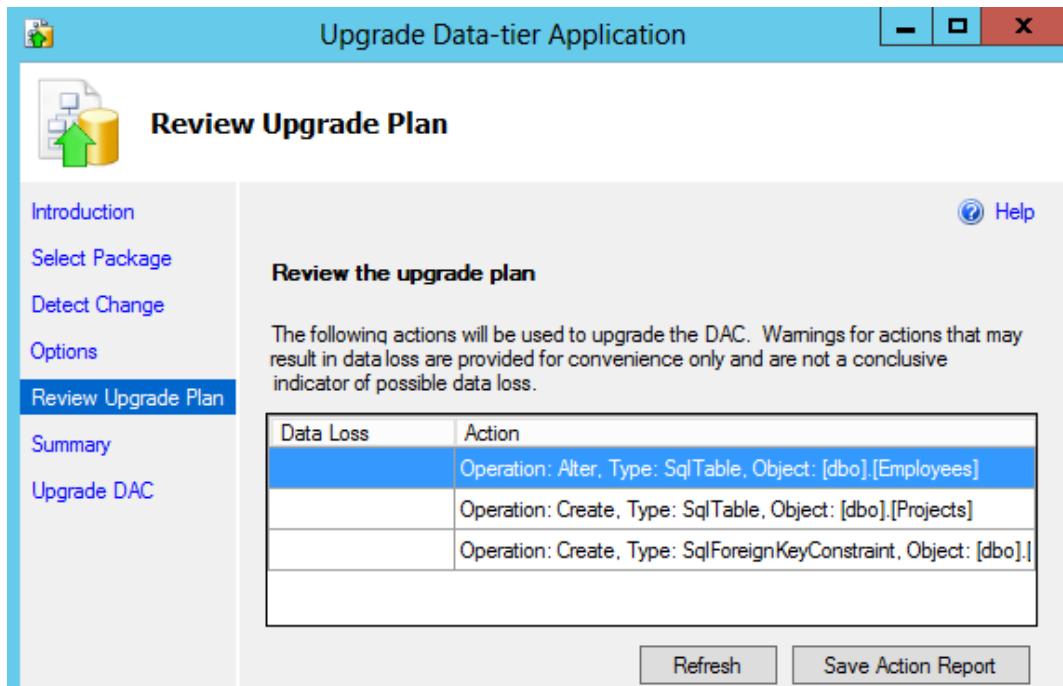
- j) In the **Object Explorer** pane, right-click **SQL01\INSTANCE01** and select **Disconnect**.
5. Register the new data-tier application in the SQL01 default server instance.
- In the **Object Explorer** pane, select **Connect→Database Engine**.
  - In the **Connect to Server** dialog box, in the **Server Name** text box, select **SQL01** and select **Connect**.
  - In the **Object Explorer** pane, in the **SQL01** default server instance, expand **Databases**.
  - Right-click **DepartmentDB** and select **Tasks→Register as Data-tier Application**.
  - In the **Register Data-tier Application** wizard, on the **Introduction** page, select **Next**.
  - On the **Set Properties** page, observe the default values and select **Next**.
  - On the **Validation and Summary** page, verify the summary of settings you specified and select **Next**.
  - On the **Registering DAC** page, observe that the registration has completed successfully and select **Finish**.
6. Launch the process to upgrade the data-tier application.
- Right-click **DepartmentDB** and select **Tasks→Upgrade Data-tier Application**.
  - In the **Upgrade Data-tier Application** wizard, on the **Introduction** page, select **Next**.
  - On the **Select Package** page, select **Browse**.
  - In the **Open** dialog box, navigate to the **C:\Temp** folder.
  - Select the **DepartmentDBv1.1** file and select **Open**.
  - On the **Select Package** page, select **Next**.



- g) On the **Detect Change** page, verify that no changes were detected in the **DepartmentDB** database and select **Next**.



- h) On the **Options** page, ensure that the **Rollback on failure** check box is unchecked and select **Next**.
- i) On the **Review Upgrade Plan** page, observe that the changes you performed on the tables in the database are listed and select **Next**.

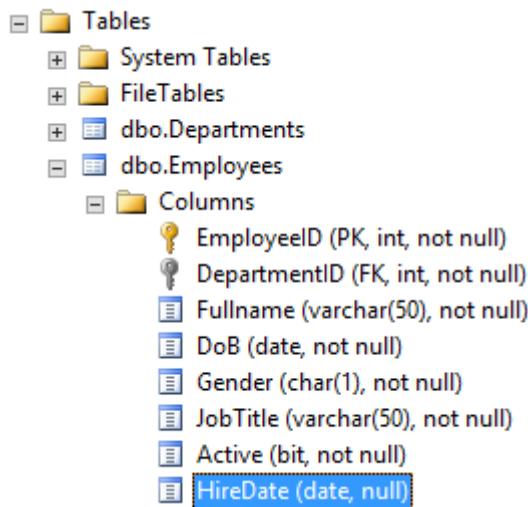


- j) On the **Summary** page, verify the settings you specified and select **Next**.
- k) On the **Upgrade DAC** page, verify that the data-tier application has been upgraded successfully and select **Finish**.

**7.** Verify the upgraded data-tier application.

- a) In the **Object Explorer** pane, expand the **DepartmentDB** database and expand **Tables**.

- b) Expand the **Employees** table and then expand **Columns**.
- c) Verify that the new column **HireDate** is displayed.



- d) Collapse the **Columns** folder.
-

## ACTIVITY 13–4

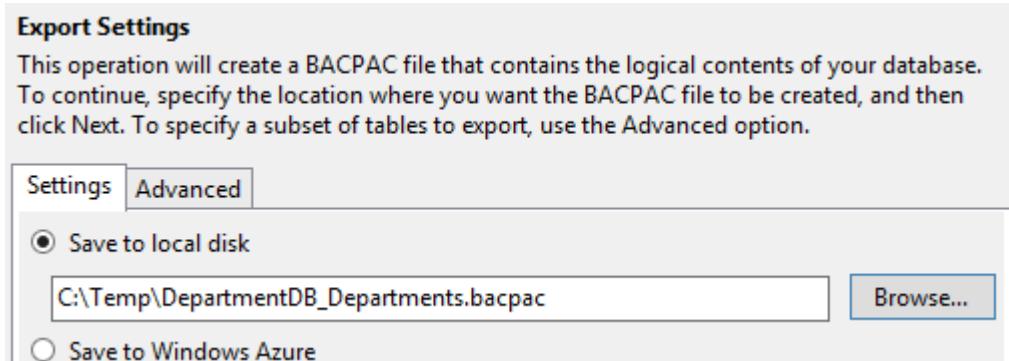
### Exporting Data from a Data-Tier application

#### Scenario

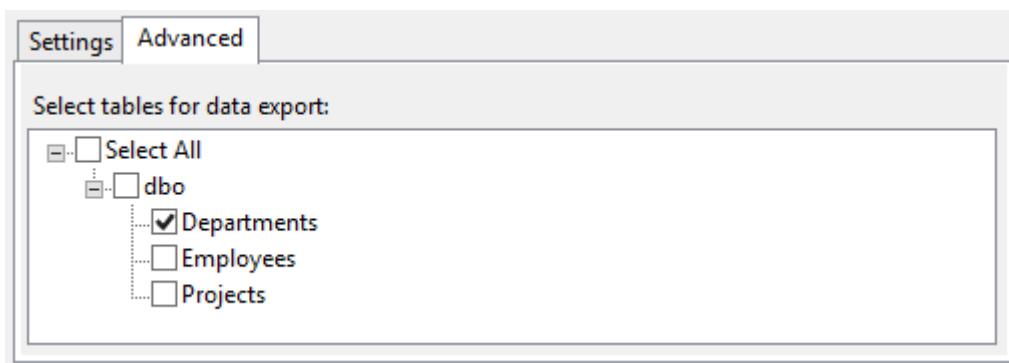
The application developers at Adventure Works Cycles are creating applications using the structure of the DepartmentDB database. To complete the development, they need the details of the departments in the organization. The Departments table in the DepartmentDB database that contains information about the departments in Adventure Works Cycles. Because the application developers require this table to develop code, you want to package the data in this table along with the data structure of the DepartmentDB database.

#### 1. Specify the settings to export a data-tier application.

- In the **Object Explorer** pane, in the **SQL01** default instance, right-click the **DepartmentDB** database and select **Tasks→Export Data-tier Application**.
- In the **Export Data-tier Application** wizard, on the **Introduction** page, select **Next**.
- On the **Export Settings** page, in the **Settings** tab, select **Browse**.
- In the **Save As** dialog box, navigate to the **C:\Temp** folder.
- In the **File name** text box, type **DepartmentDB.bacpac** and select **Save**.



- Select the **Advanced** tab.
- In the **Select tables for data export** list, uncheck the **Select All** check box.
- Expand **Select All→dbo** and check the **Departments** check box and select **Next**.



#### 2. Verify the settings and complete the export operation.



Inform participants that selecting a table here copies the contents of the table. When the BACKPAC file is imported, the contents of the exported table are copied to the destination database.

- a) On the **Summary** page, verify that the settings you specified are listed and select **Finish**.
  - b) On the **Results** page, verify that the export operation has completed successfully and select **Close**.
  - c) In Windows Explorer, navigate to the **C:\Temp** folder.
  - d) Observe that the exported data-tier application is saved as the **DepartmentDB.bacpac** file in this folder.
-

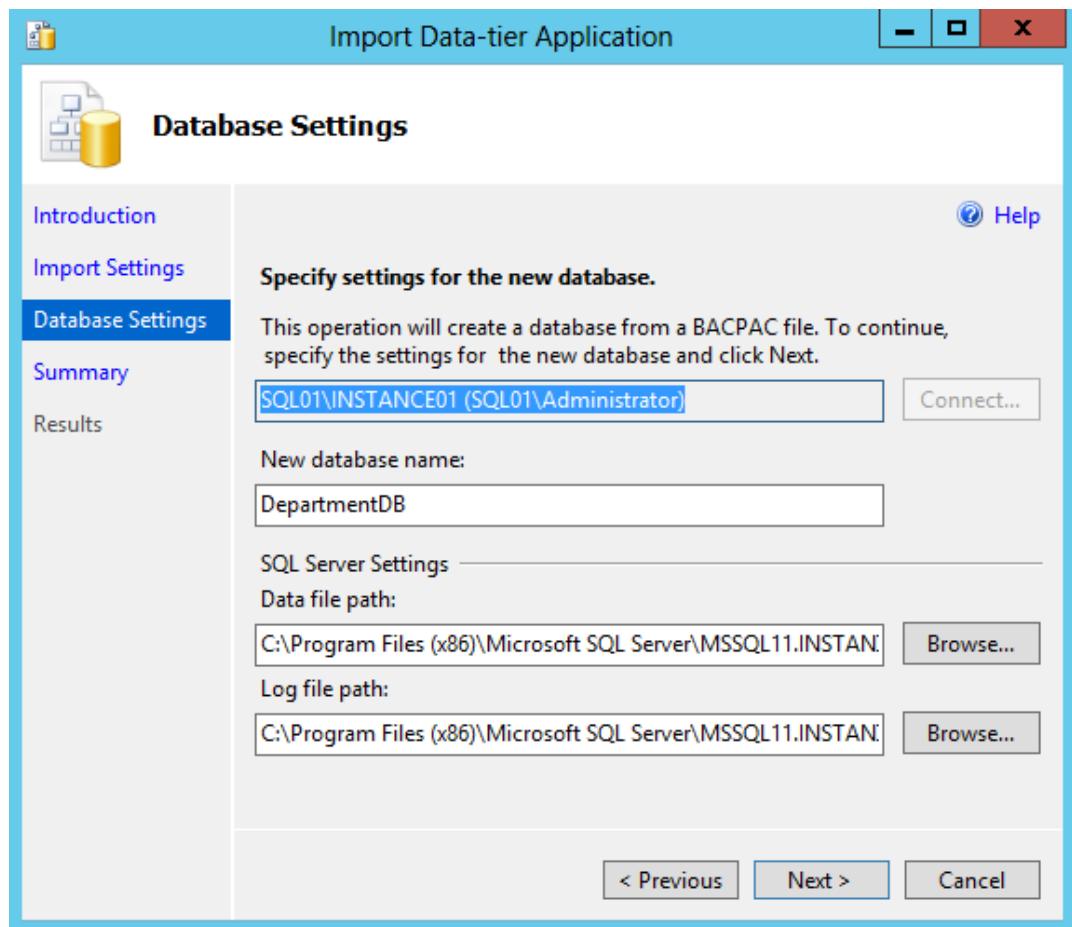
# ACTIVITY 13–5

## Importing an Exported Data-Tier Application

### Scenario

As database administrator at Adventure Works Cycles, you created a .bacpac file that stores the database structure along with the Departments table. To enable the application developer access the required data structure and data, you want to import the .BAC package on another instance of SQL Server.

1. Select the .BAC package to be imported.
  - a) In the **Object Explorer** pane, right-click **SQL01** and select **Disconnect**.
  - b) In the **Object Explorer** pane, select **Connect→Database Engine**.
  - c) In the **Connect to Server** dialog box, in the **Server Name** text box, select **SQL01\INSTANCE01** and select **Connect**.
  - d) In the **Object Explorer** pane, in the **SQL01\INSTANCE01** instance, right-click **Databases** folder and select **Import Data-tier Application**.
  - e) In the **Import Data-tier Application** wizard, on the **Introduction** page, select **Next**.
  - f) On the **Import Settings** page, verify that the **Import from local disk** option is selected and select **Browse**.
  - g) In the **Open** dialog box, navigate to the **C:\Temp** folder.
  - h) Select the **DepartmentDB.bacpac** file and select **Open**.
2. Import the select .BAC package.
  - a) On the **Import Settings** page, select **Next**.
  - b) On the **Database Settings** page, in the **New database name** text box, verify that the name of the database to be imported is **DepartmentDB** and select **Next**.



3. Verify that the package has been imported correctly.
  - c) On the **Summary** page, verify that the settings you specified are listed, and select **Finish**.
  - d) On the **Complete** page, select **Close**.

The screenshot shows the SQL Server Management Studio (SSMS) interface. A query window titled "SQLQuery1.sql - SQ...Administrator (56)\*" is open, displaying three SELECT statements:

```
SELECT * FROM Departments;
SELECT * FROM Employees;
SELECT * FROM Projects;
```

The "Results" tab is selected, showing the output for the first query (Departments). The data is presented in a table:

|   | DepartmentID | Name            |
|---|--------------|-----------------|
| 1 | 1            | Human Resources |
| 2 | 2            | IT and Support  |
| 3 | 3            | Development     |
| 4 | 4            | Marketing       |
| 5 | 5            | Operations      |

Below this, two more tables are shown, each with a single row of data:

|  | EmployeeID | DepartmentID | Fullname | DoB |
|--|------------|--------------|----------|-----|
|  |            |              |          |     |

|  | ProjectID | ProjectName | ProjectOwner |
|--|-----------|-------------|--------------|
|  |           |             |              |

- f) Close the **Query Editor** pane without saving.

# TOPIC B

## Configure SSIS

You have managed the database infrastructure to ensure that the environment is up-to-date and delivers good performance. As part of the advanced management tasks that you perform on the databases in your network, you need to obtain information from different data sources and populate them into the databases. In this topic, you will configure SSIS.

### SSIS

*SQL Server Integration Services (SSIS)* is a component of SQL Server that allows you to perform Extract, Transform, and Load (ETL) processes on data. SSIS can connect to different data sources, extract data from them, transform them into a format of your choice and load them into a different database. SSIS includes the Import and Export wizard that enables you to import databases from and export databases to different data sources.

### SSIS Security

SSIS works by the concept of packages. These packages can be stored at the file system or within a database. You can use NTFS security or database to protect access to these packages.

These packages also have a built-in option to be able to encrypt the contents. This is called the Package Protection Level and can be configured to use one of the values listed in the table.

| <b>Package Protection Level Value</b> | <b>When This Value Is Selected</b>                                                                                                                                                                                                                                                                     |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Don't Save Sensitive                  | The package is unprotected but does not save any sensitive data (for instance, the username and password for connection strings). Each time the package is executed, the user will need to supply the values for sensitive information.                                                                |
| Encrypt Sensitive with Password       | Sensitive information such as usernames and passwords are encrypted using a common password. For opening or executing the package, the password must be supplied otherwise the user will be prompted for values for the sensitive information.                                                         |
| Encrypt Sensitive with User Key       | Sensitive information is encrypted using the current user's credentials. Therefore only the current user can open or execute the package with the values within the sensitive fields. When a different user tries to open or execute the package, he will be prompted for new values for these fields. |
| Encrypt All with Password             | The entire package is encrypted using a password. Opening or running the package requires supplying the password. If not supplied, the package will simply fail to open or execute.                                                                                                                    |
| Encrypt All with User Key             | The entire package is encrypted using the current user's credentials and only he can open or run it.                                                                                                                                                                                                   |
| Server Storage                        | SQL Server database roles are enabled to protect the contents of the package. However, the roles are enabled only if the package is stored within SQL Server and not in the file system.                                                                                                               |

## The Data Collector

The *data collector* is a component that retrieves data from different systems on either a constant or pre-defined schedule. This data is then directed into a database termed as the Management Data Warehouse. This data can be used to process the state of any database or server in your network.

The data is collected for a number of parameters under disk activity, query activity and server activity headings as well as miscellaneous ones. These are collected locally in a temp folder and then uploaded in batches into the data warehouse. DBAs can then analyze the data warehouse information to check the statistics of the server's activities.

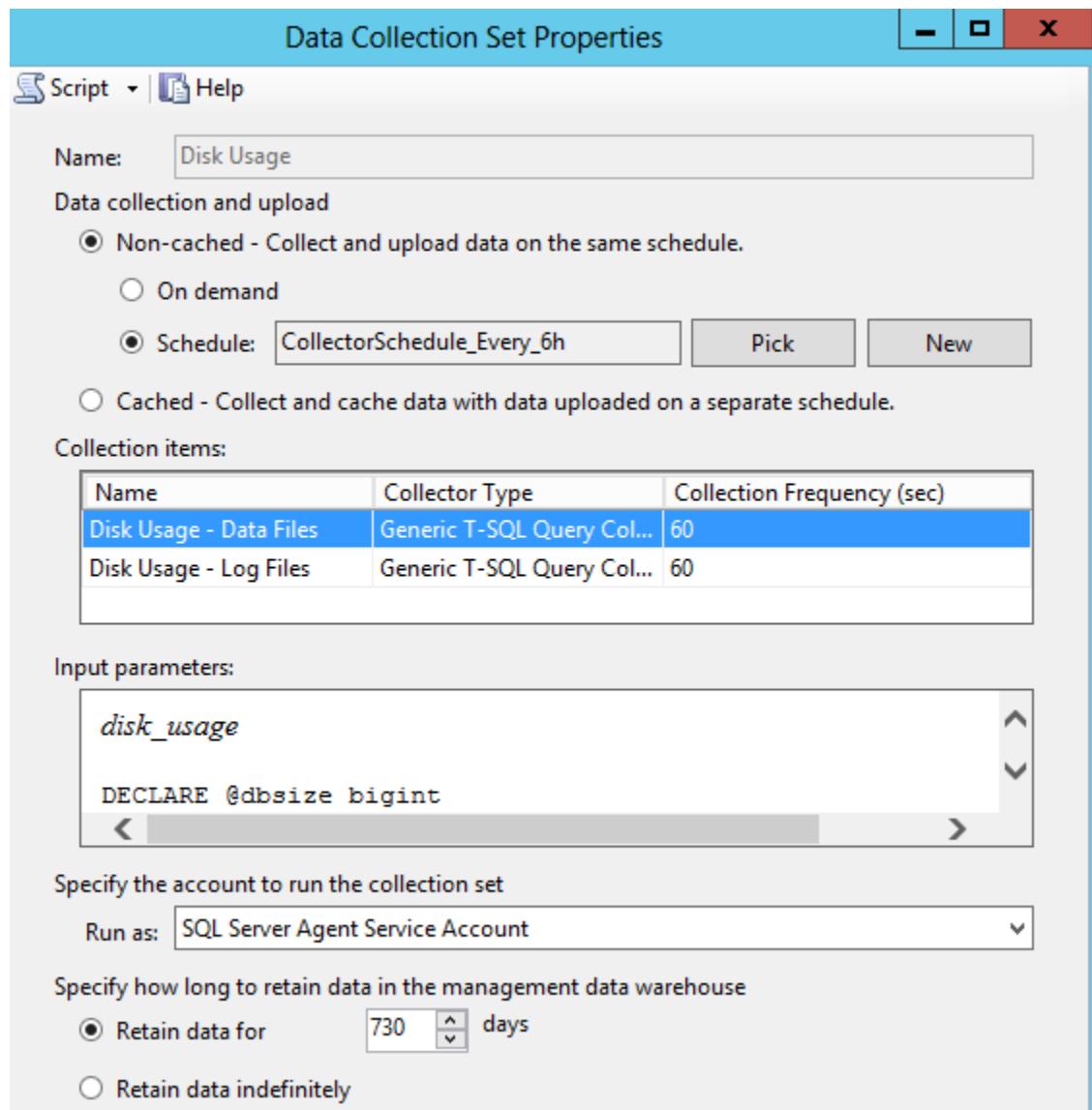


Figure 13–6: The Data Collection Set Properties window is used to collect database information.

## Multi-TB Databases

*Multiple terabyte (Multi-TB) databases* are very large SQL Server databases that store terabytes of data. However, managing these databases is difficult if not planned properly. Some of the guidelines to manage multi-TB databases include.

- Using files and filegroups to ensure different files are placed across different disks and I/O controllers for maximum performance.

- Partitioning tables and placing them in appropriate file groups.
- Turning off Auto-shrink and Auto-grow.
- Creating indexes correctly.
- Storing data on high speed disks or SANs with adequate network bandwidth.
- Using disk mirroring and replication to maintain sets of data at local locations for quick querying.
- Visualizing, predicting, and creating the database system to grow into a larger set.

SQL Server 2012 also supports Big Data which enables you to work with extremely large and complex sets of data such as Internet search results, biological genetic data, meteorological predictions, and data in particle physics.



Access the Checklist tile on your LogicalCHOICE course screen for reference information and job aids on **How to Configure SSIS**

# ACTIVITY 13–6

## Configuring SSIS Security

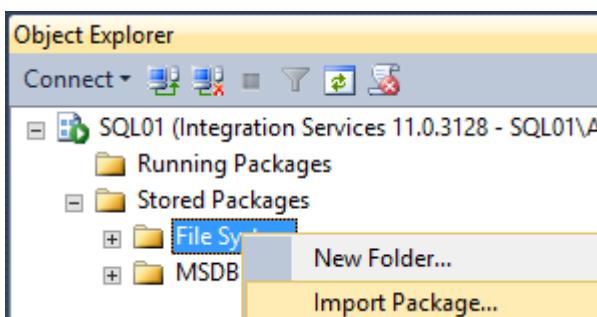
### Before You Begin

1. From the C:\093108Data\Performing Advanced Database Management Tasks folder, copy and paste the zip file named - **SQL2012.Integration\_Services.Create\_Simple\_ETL\_Tutorial.Sample.zip** to C:\Temp folder.
2. Unzip the package and store the files in the **Temp** folder.
3. Disconnect from **INSTANCE01**.

### Scenario

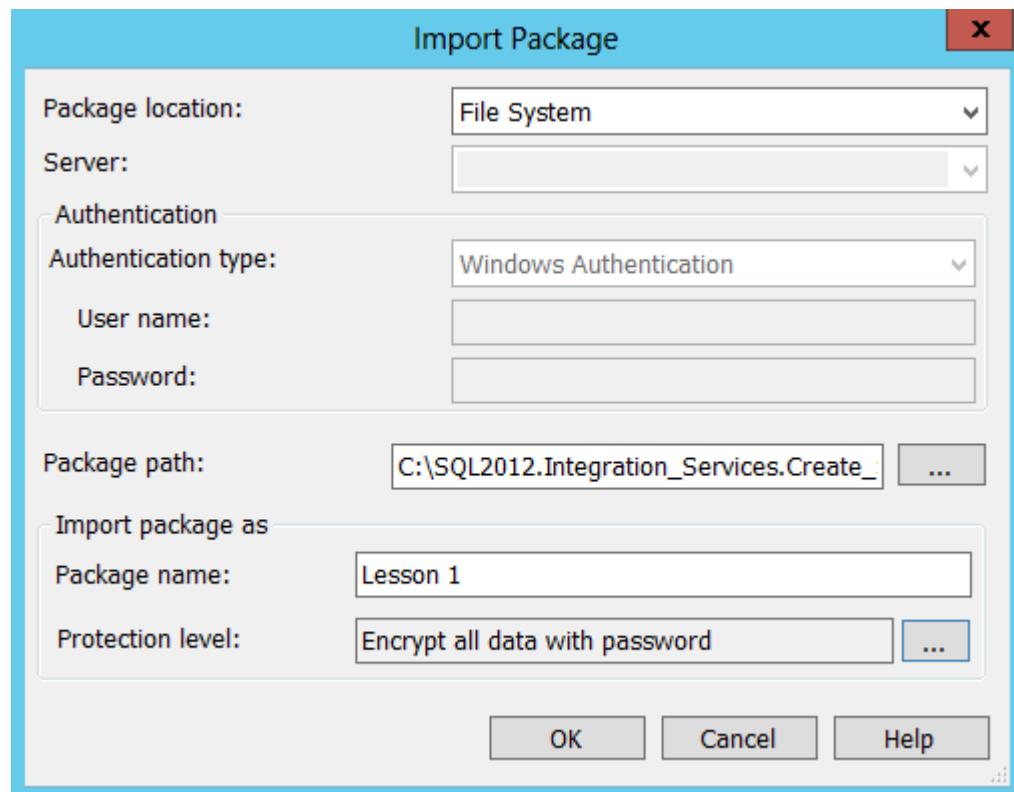
Software developers of the IT department have given you an SSIS package that needs to be deployed on the server. Because the package contains some sensitive information, you are asked to ensure that the package is encrypted. You decide to configure the required security settings to the SSIS package.

1. Import a package on the Integration Services.
    - a) In the **Object Explorer** pane, select the **Connect** button and from the drop-down list, select **Integration Services**.
    - b) In the **Connect to Server** dialog box, in the **Server type** drop-down list, select **Integration Services**, and in the **Server name** drop-down list, verify that the **SQL01** server is selected.
    - c) In the **Authentication** drop-down list, verify that **Windows Authentication** is selected and select **Connect** to connect to the server.
-  **Note:** The user name and login name can be searched in the search dialog box by selecting the **Search** button.
- d) In the **Integration Services** folder, expand **Stored Packages** folder.
  - e) Right-click the **File System** folder and from the fly-out pane, select **Import Package**.

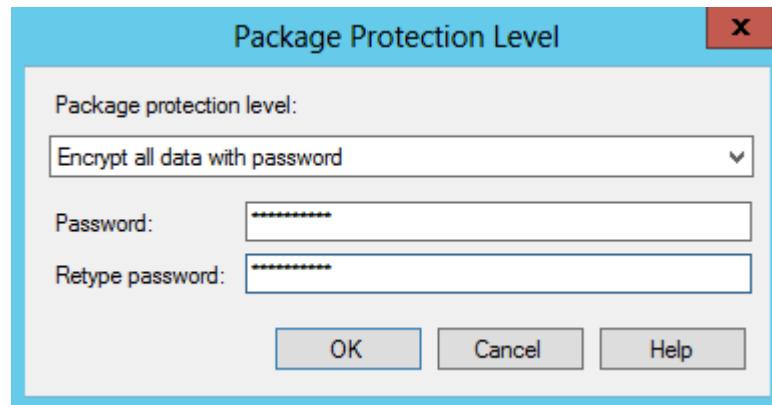


- f) In the **Import Package** dialog box, select the **ellipsis (...)** button next to the **Package path** text box.
- g) In the **Load Package** dialog box, navigate to the C:\Temp\SQL2012.Integration\_Services.Create\_Simple\_ETL\_Tutorial.Sample folder, select **Lesson1.dtsx**, and select **Open**.

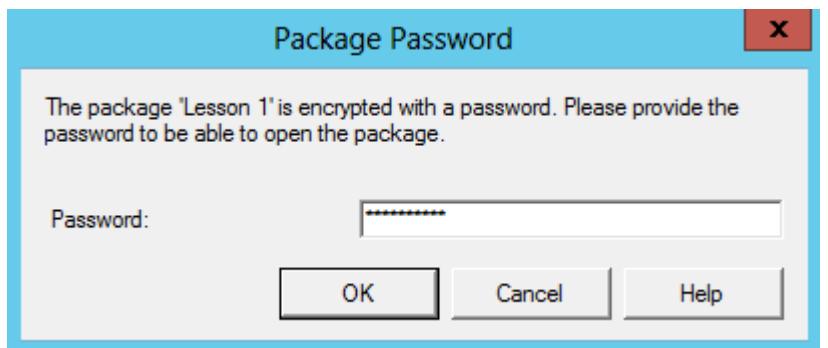




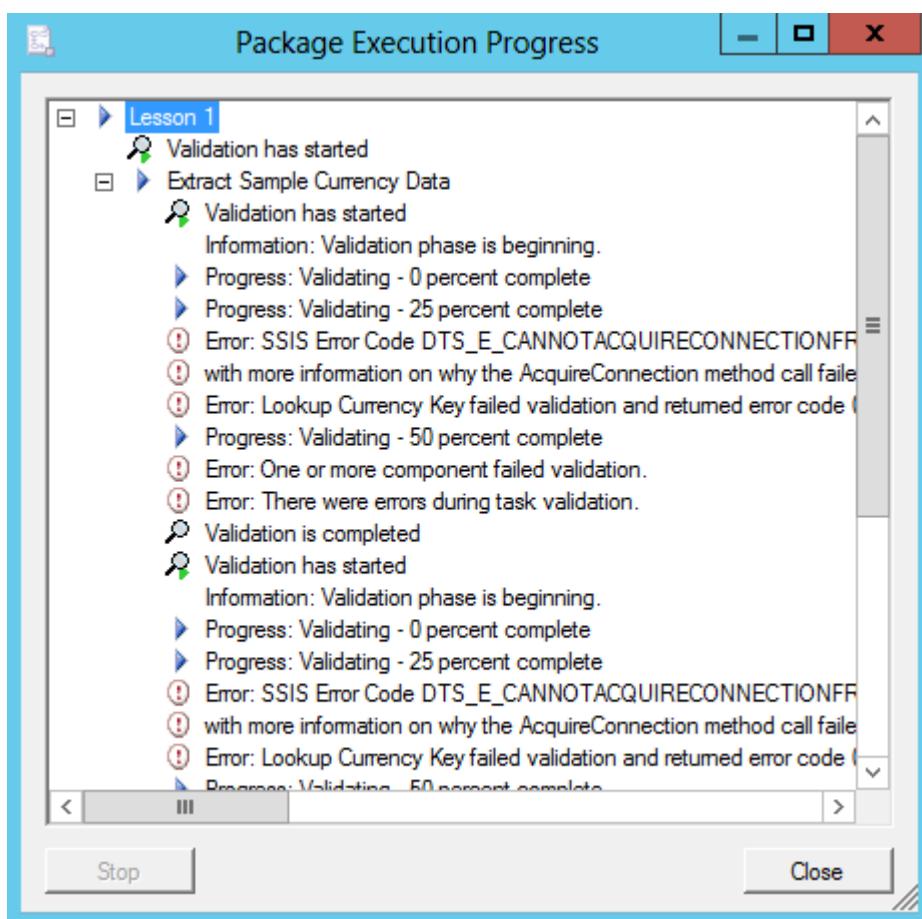
- h) In the **Import Package** dialog box, select the **ellipses (...)** button next to the **Protection level** text box.
- i) In the **Package Protection Level** dialog box, from the **Package protection level** drop-down list, select **Encrypt all data with password** level.
- j) In the **Password** text box, type “**pass@word1**”, and in the **Retype password** text box, type the password again. Select **OK** to set the password.



- k) In the **Import Package** dialog box, select **OK** to import the package with encryption.
2. Run the package that was imported.
  - a) On the **Object Explorer** pane, expand the **File System** folder, right-click the **Lesson1** package and from the fly-out pane, select **Run Package**.
  - b) In the Execute Package Utility window, in the **Package source** drop-down list, ensure that **SSIS Package Store** is selected and select **Execute**.
  - c) In the **Package Password** dialog box, you will be prompted to type the password for Lesson1 package, in the **Password** text box, type a wrong password and select **OK**.



- d) The **Package Password** dialog box will throw an error message and asks you to type the password again. In the **Password** text box, type **pass@word1** and select **OK**.
- e) In the Package Execution Progress window, observe that the package now executes correctly. Select **Close** to close the window.



- f) In the Execute Package Utility window, select **Close** to close the window.
- g) In the **Object Explorer** pane, right-click the **Integration Services** folder and from the fly-out pane, select **Disconnect**.

# ACTIVITY 13-7

## Configuring Data Collector

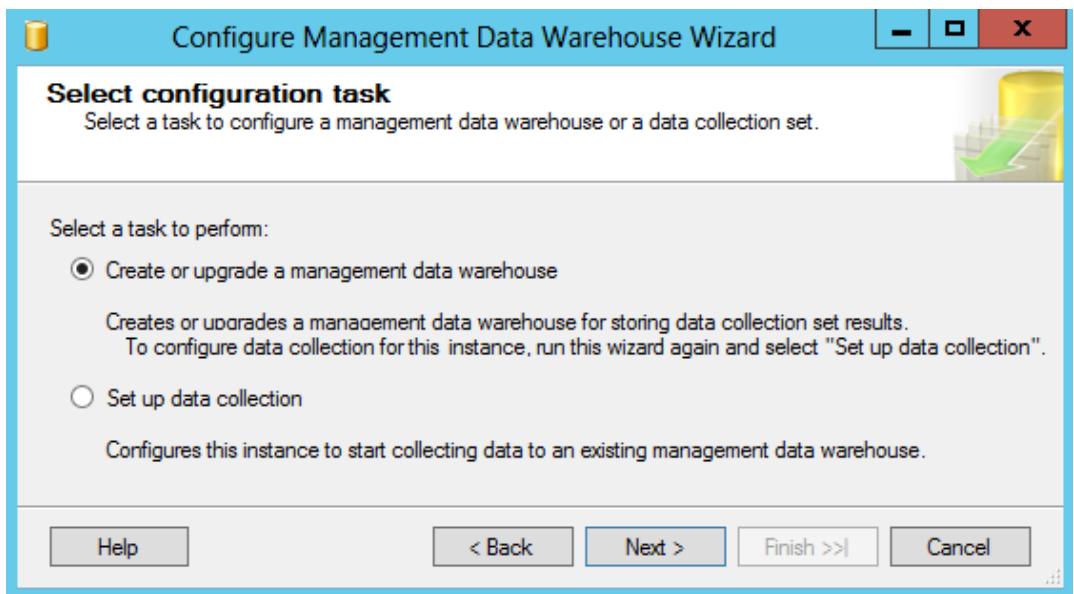
### Before You Begin

1. Connect to SQL01.
2. Ensure that SQL Server Agent is running.

### Scenario

Management of Adventure Works Cycles has decided to conduct an inspection next week. As database administrator of Adventure Works Cycles, you are assigned the task of collecting detailed information and presenting it. You are asked to collect performance-based data from the data warehouse. You decide to configure the data collector in order to perform this task.

1. Setup Configure Management Data Warehouse to create a management data warehouse.
  - a) In the **Object Explorer** pane, right-click **SQL01** and select **Refresh**.
  - b) In the **Object Explorer** pane, expand the **Management** folder.
  - c) Right-click the **Data Collection** folder and from the fly-out pane, select **Configure Management Data Warehouse**.
  - d) In the **Configure Management Data Warehouse Wizard**, select **Next** to continue with the configuration.
  - e) On the **Select configuration task page**, in the **Select a task to perform** section, ensure that **Create or upgrade a management data warehouse** option is selected. Select **Next** to continue with the configuration.

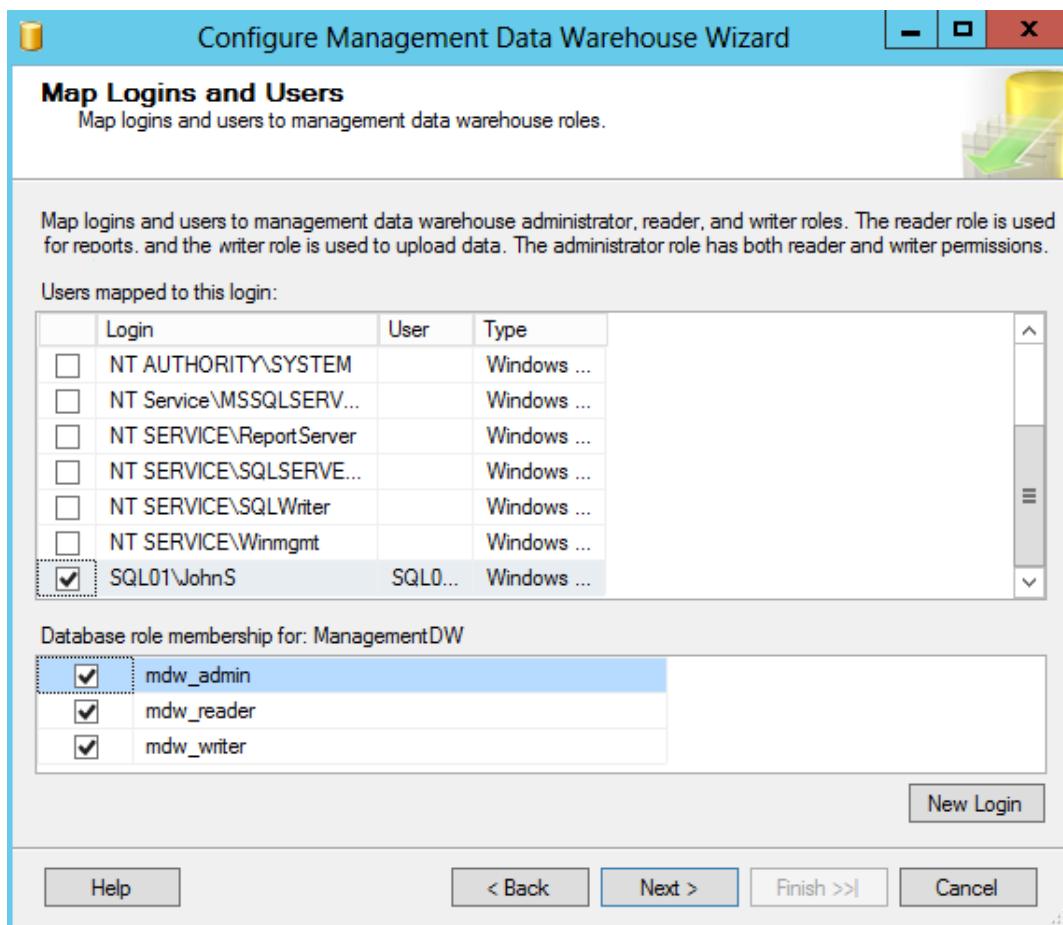


- f) On the **Configure Management Data Warehouse Storage** page, in the **Select a server and database to host your management data warehouse** section, in the **Server name** text box, ensure that **SQL01** is displayed.
- g) In the **Database name** section list, select the **New** button to create a new management data warehouse.
- h) In the New Database window, in the **Database name** text box, type **ManagementDW** and select **OK**.

- i) On the **Configure Management Data Warehouse Storage** page, ensure that in the **Database name** text box **ManagementDW** is displayed. Select **Next** to continue with the configuration.



- j) On the **Map Logins and Users** page, in the **User mapped to this login** section select **SQL01\JohnS**.  
 k) In the **Database role membership for: ManagementDW** section, check the **mdw\_admin** check box and ensure that the **mdw\_reader**, and **mdw\_writer** check boxes are checked in order to give rights to **SQL01\JohnS**. Select **Next** to continue with the configuration.



- i) In the **Complete the Wizard** page, observe that **SQL01** is the **Server** that is selected and **ManagementDW** is the name of the data warehouse selected. Select **Finish** to complete the **Configure Management Warehouse** wizard.



**Note:** The configuration might take a few minutes.

- m) On the **Configure Data Collection Wizard Progress** page, ensure that the configuration is completed successfully. Select **Close** to exit the configuration wizard.

2. Configuring Management Data Warehouse to set up data collection.
  - a) In the **Object Explorer** pane, right-click the **Data Collection** folder and from the fly-out pane, select **Configure Management Data Warehouse**.
  - b) In the **Configure Management Data Warehouse Wizard**, select **Next** to continue with the configuration.
  - c) On the **Select configuration task** page, in the **Select a task to perform** section select the **Set up data collection** option. Select **Next** to continue with the configuration.
  - d) On the **Configure Management Data Warehouse Storage** page, in the **Select a server and database that is the host for your management data warehouse** section, in the **Server name** text box, select the **ellipses (...)** button.
  - e) In the **Connect to Server** dialog box, in the server name drop-down list, ensure that **SQL01** is selected and select **Connect**.
  - f) In the **Select a server and database that is the host for your management data warehouse** section, in the **Server name** text box, observe that the server name **SQL01** is displayed.
  - g) In the **Database name** drop-down list, observe that **ManagementDW** data warehouse name is selected, and if it is not listed, select it from the drop-down list. Select **Next** to continue with the configuration.

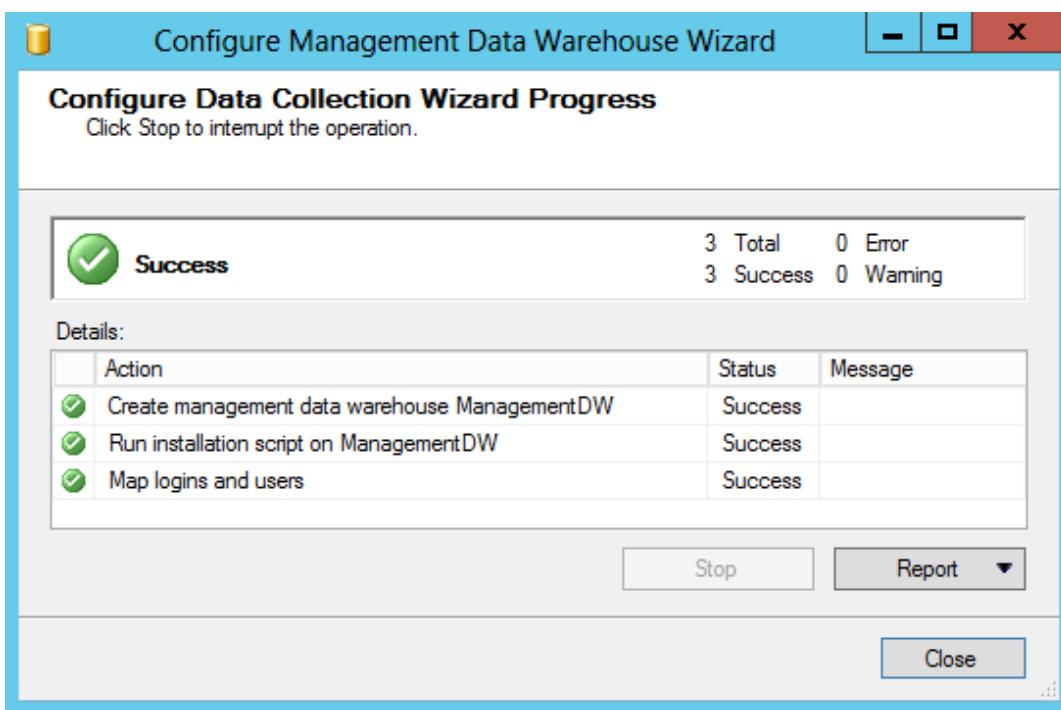


**Note:** In the **Configure Management Data Warehouse Storage** page, in the **Cache directory** text box leave it blank. The blank value will use the Temp directory for the collector process.

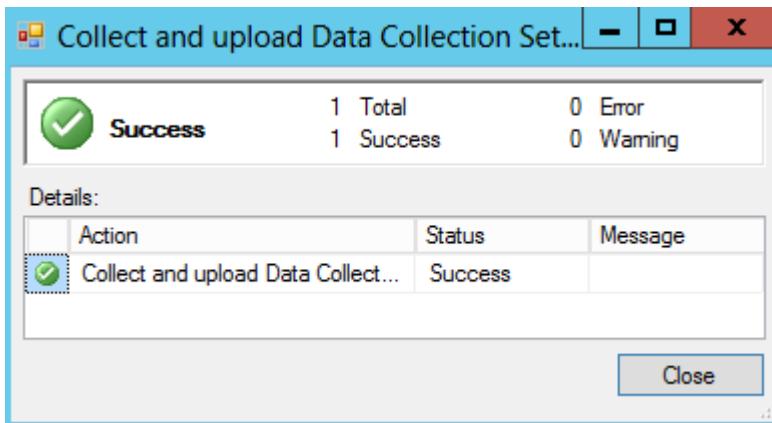
- h) In **Complete the Wizard** page, observe that **SQL01** is the **Server** that is selected and **ManagementDW** is the **Name** of the data warehouse selected. Select **Finish** to complete the **Configure Management Warehouse** wizard.



**Note:** The configuration might take a few minutes.



- i) On the **Configure Data Collection Wizard Progress** page, ensure that the configuration is completed successfully. Select **Close** to exit the configuration wizard.
3. Generate reports for the Configured Data Collection Sets.
  - a) In the Object Explorer pane, expand Data Collection→System Data Collection Sets.
  - b) Right-click the Disk Usage node and from the fly-out pane, select Collect and Upload Now.
  - c) In the Collect and upload Data Collection Sets dialog box, ensure that upload was successful. Select Close.



- d) In the Object Explorer pane, right-click the **Query Statistics** node and from the fly-out pane, select Collect and Upload Now.
- e) In the Collect and upload Data Collection Sets dialog box, ensure that upload was successful. Select Close.
- f) In the Object Explorer pane, right-click the **Server Activity** node and from the fly-out pane, select Collect and Upload Now.
- g) In the Collect and upload Data Collection Sets dialog box, ensure that upload was successful. Select Close.
- h) In the Object Explorer pane, right-click the **Disk Usage** node and from the fly-out pane, select Properties.



**Note:** The Properties window can be opened for any of these three Data Collectors.

- i) In the **Data Collection Set Properties** dialog box, in the **Select a page** section, select the **Uploads**, **General** and **Description** pages and explore the options in them. Select **Cancel** to close the properties window.
-

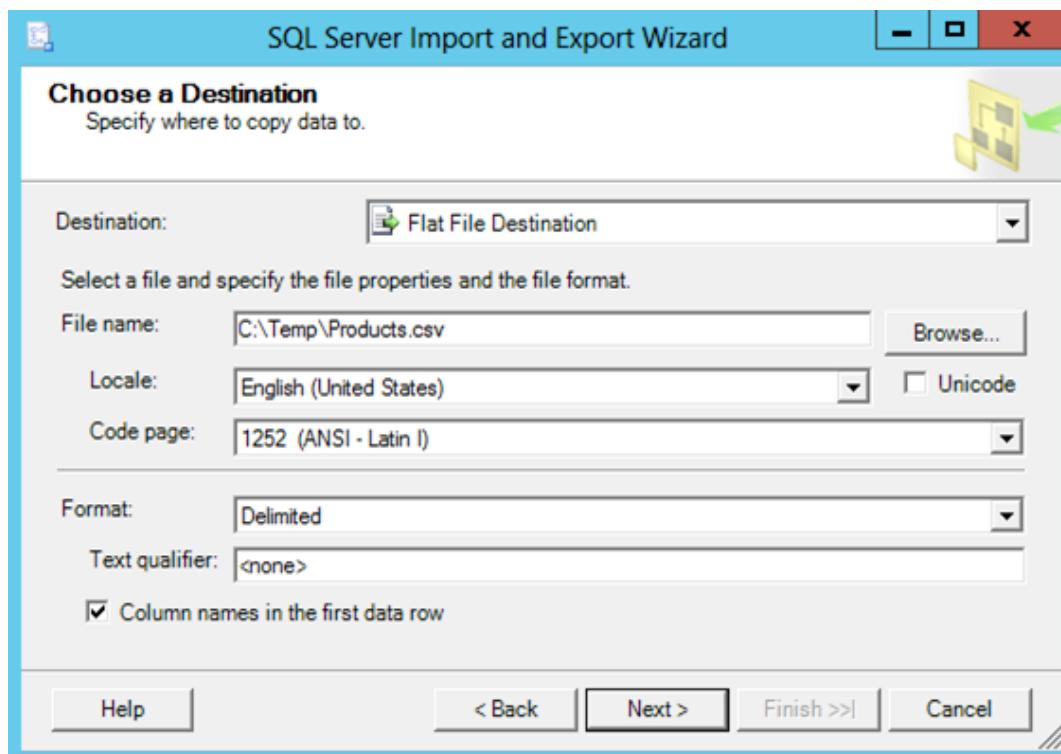
# ACTIVITY 13-8

## Exporting Packages to an SSIS Project

### Scenario

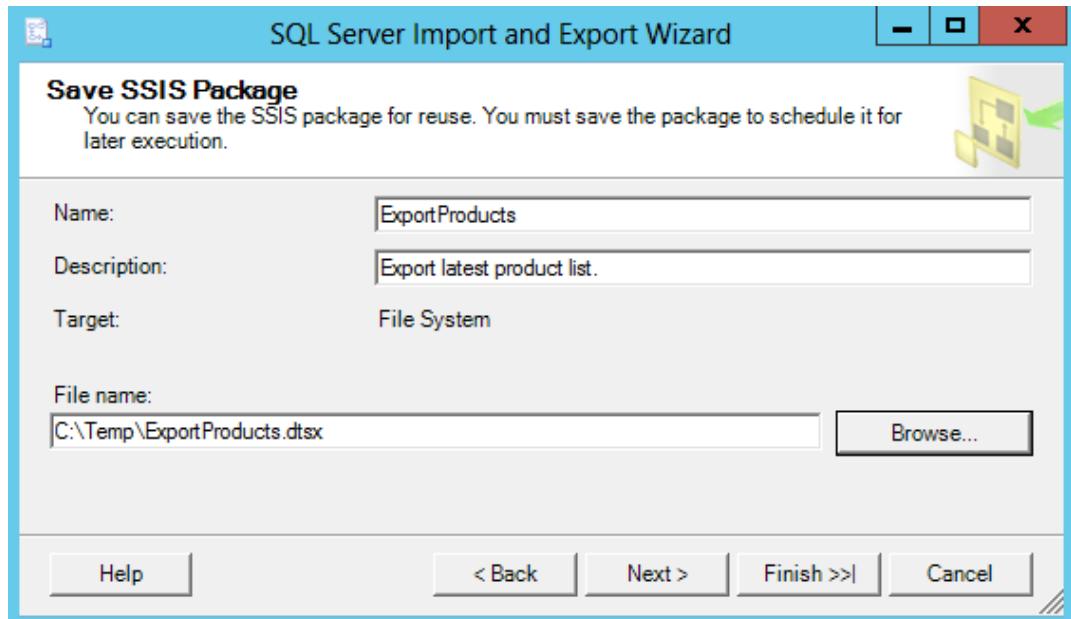
Adventure Works Cycles has several sales outlets that serve the client requirements. These outlets use different database systems to store and access data. As a database administrator, you wish to add the Export of data from a database table as an operation in your project. For this, you plan to import the SSIS package that the SQL Server Import/Export Wizard builds automatically.

1. Export data in the AdventureWorksCycles2012.Production.Product table to an SSIS project.
  - a) In the **Object Explorer** pane, in the **SQL01** default server instance, expand **Databases** folder, right-click the **AdventureWorks2012** database and select **Tasks→Export Data**.
  - b) On the **SQL Server Import and Export** wizard, select **Next** to continue with the setup.
  - c) On the **Choose a Data Source** page, in the **Database** section, ensure that **AdventureWorks2012** database is selected and select **Next** to continue.
  - d) On the **Choose a Destination** page, from the **Destination** drop-down list, select **Flat File Destination**.
  - e) In the **File name** text box, type **C:\Temp\Products.csv**.
  - f) From the **Format** drop-down list, ensure that **Delimited** is selected and ensure that the **Column names in the first data row** check box is checked. Select **Next** to continue.



- g) On the **Specify Table Copy or Query** page, ensure that **Copy data from one or more tables or views** is selected and select **Next** to continue.
- h) On the **Configure Flat File Destination** page, from the **Source table or view** drop-down list, select **[Production].[Product]** and select **Next** to continue.
- i) On the **Save and Run Package** page, uncheck the **Run immediately** check box, verify that the **Save SSIS Package** check box is checked and select the **File system** option.

- j) From the **Package protection level** drop-down list, select **Encrypt sensitive data with password**.
- k) In the **Password and Retype password** text boxes, type **pass@word1** and select **Next** to continue.
- l) On the **Save SSIS Package** page, in the **Name** text box, type **ExportProducts** and in the **Description** text box, type **Exports latest product list**.
- m) In the **File name** text box, type **C:\Temp\ExportProducts.dtsx** and select **Next** to continue.



- n) On the **Complete the Wizard** page, verify the options selected and select **Finish**.
  - o) On the **Results** page, observe that the export was successful and select **Close**.
2. Which among the following Package Protection Level options allows using SQL Server database roles to protect the contents of the package?
    - Encrypt Sensitive with Password
    - Encrypt Sensitive with User Key
    - Encrypt All with User Key
    - Encrypt All with Password
    - Server Storage
  3. Which of the following statements are true with regard to working with multi-TB databases? (Choose three).
    - Tables should use partitioning to place them in appropriate file groups.
    - Auto-shrink and Auto-grow should be turned on.
    - I/O should be placed on high speed disks or SANs with enough network bandwidth.
    - Disk mirroring can help keep sets of data at “database” locations for querying.
    - Indexing should be done correctly.

# Summary

In this lesson, you used the advanced management features available in SQL Server 2012 to manage database infrastructure and configured SSIS. This will enable you to configure your server properties depending on your requirements and improve the performance, scalability, and reliability of your database system.

**In what ways do you think is having a central management data warehouse for data collection advantageous over the local installations?**

A: Answers will vary, but may include: to have a single point for performance analysis and to improve the storage from production systems.



Use the review questions provided to generate discussion among the participants.

**In what circumstances would you use the data-tier applications?**

A: Answers will vary, but may include: to create .DACPAC and .BACPAC files from an existing database, to pass to developers to enable further development using data-tier applications.



**Note:** Check your LogicalCHOICE Course screen for opportunities to interact with your classmates, peers, and the larger LogicalCHOICE online community about the topics covered in this course or other topics you are interested in. From the Course screen you can also access available resources for a more continuous learning experience.



Encourage students to use the social networking tools provided on the LogicalCHOICE Course screen to follow up with their peers after the course is completed for further discussion and resources to support continued learning.



# Course Follow-Up

Congratulations! You have completed the *Microsoft® SQL Server® 2012: Database Administration (Exam 70-462)* course. In this course, you planned and deployed an SQL Server 2012 setup and administered databases. You implemented a SQL Server setup and configured databases on it. You also configured multiple server instances, managed the databases in them, controlled access to the databases, automated SQL Server tasks, and managed the data in the databases. To ensure that the databases in your setup function without any hassles you implemented high availability, optimized performance, addressed database issues, and performed advanced management tasks.

With this hands-on knowledge, as database administrator, you are well equipped to manage and implement SQL Server 2012 databases at the enterprise level.

## What's Next?

This is the second course in the Microsoft® SQL Server® 2012 series. You can take up the following Logical Operation's course to enhance your knowledge and prepare for Microsoft Certified Solutions Associate (MCSA): SQL Server 2012 certification:

- *Microsoft® SQL Server® 2012: Data Warehouse Implementation (Exam 70-463)*

You are also encouraged to explore SQL Server 2012 further by actively participating in any of the social media forums set up by your instructor or training administrator through the **Social Media** tile on the LogicalCHOICE Course screen.



# A

# Exam 70-462: Administering Microsoft SQL Server 2012 Databases

The following table lists the exam objectives of Exam 70-462: Administering Microsoft® SQL Server® 2012 Databases and where they are covered in the course.

| <i>Exam 70-462 Mapping Objectives</i>                                                                      | <i>Covered In</i> |
|------------------------------------------------------------------------------------------------------------|-------------------|
| <b>Section 1: Install and Configure (19%)</b>                                                              |                   |
| <b>Plan installation.</b>                                                                                  |                   |
| Evaluate installation requirements                                                                         | Topic 1-A         |
| Design the installation of SQL Server and its components (drives, service accounts etc.)                   | Topic 1-B, 2-A    |
| Plan scale up vs. scale out basics                                                                         | Topic 1-A         |
| Plan for capacity                                                                                          | Topic 1-A         |
| Including if/when to shrink                                                                                | Topic 4-B         |
| Grow, autogrow, and monitor growth                                                                         | Topic 8-C         |
| Manage the technologies that influence SQL architecture (e.g., service broker, full text, scale out, etc.) | Topic 1-A         |
| Design the storage for new databases (drives, filegroups, partitioning)                                    | Topic 1-B         |
| Design database infrastructure                                                                             | Topic 1-B, 2-A    |
| Configure a SQL Server standby database for reporting purposes                                             | Topic 13-A        |
| Windows-level security and service level security                                                          | Topic 2-A         |
| Core mode installation                                                                                     | Topic 2-A         |
| Benchmark a server before using it in a production environment (SQLIO, Tests on SQL Instance)              | Topic 2-A, 11-A   |
| Choose the right hardware                                                                                  | Topic 1-A         |
| <b>Install SQL Server and related services.</b>                                                            |                   |

| <b>Exam 70-462 Mapping Objectives</b>                                                                                                | <b>Covered In</b> |
|--------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Test connectivity                                                                                                                    | Topic 12-A        |
| Enable and disable features                                                                                                          | Topic 2-A         |
| Install SQL Server database engine and SSIS (not SSRS and SSAS)                                                                      | Topic 2-A, 13-B   |
| Configure an OS disk                                                                                                                 | Topic 2-A         |
| <b>Implement a migration strategy.</b>                                                                                               |                   |
| Restore vs detach/attach                                                                                                             | Topic 7-A, 9-B    |
| Migrate security                                                                                                                     | Topic 5-C         |
| Migrate from a previous version                                                                                                      | Topic 2-B         |
| Migrate to new hardware                                                                                                              | Topic 2-B         |
| Migrate systems and data from other sources                                                                                          | Topic 2-B         |
| <b>Configure additional SQL Server components.</b>                                                                                   |                   |
| Setup and configure all SQL Server components (Engine, AS, RS and SharePoint integration) in a complex and highly secure environment | Topic 2-A         |
| Configure full-text indexing                                                                                                         | Topic 8-A         |
| SSIS security                                                                                                                        | Topic 13-B        |
| filestream                                                                                                                           | Topic 8-B         |
| filetable                                                                                                                            | Topic 8-B         |
| <b>Manage SQL Server Agent.</b>                                                                                                      |                   |
| Create, maintain, and monitor jobs                                                                                                   | Topic 9-D         |
| Administer jobs and alerts                                                                                                           | Topic 9-D         |
| Automate (setup, maintenance, monitoring) across multiple databases and multiple instances                                           | Topic 13-A        |
| Send to "Manage SQL Server Agent jobs"                                                                                               | Topic 9-D         |
| <b>Section 2: Maintain Instances and Databases (17%)</b>                                                                             |                   |
| <b>Manage and configure databases.</b>                                                                                               |                   |
| Design multiple file groups                                                                                                          | Topic 1-B         |
| Database configuration and standardization: autoclose, autoshrink, recovery models                                                   | Topic 4-B         |
| Manage file space, including adding new filegroups and moving objects from one filegroup to another                                  | Topic 4-B         |
| Implement and configure contained databases                                                                                          | Topic 4-A         |
| Data compression; configure TDE                                                                                                      | Topic 7-B, 7-A    |
| Partitioning; manage log file growth                                                                                                 | Topic 1-B, 8-C    |
| DBCC                                                                                                                                 | Topic 12-A        |
| <b>Configure SQL Server instances.</b>                                                                                               |                   |
| Configure and standardize a database: autoclose, autoshrink, recovery models                                                         | Topic 4-A         |

| <b>Exam 70-462 Mapping Objectives</b>                                              | <b>Covered In</b>     |
|------------------------------------------------------------------------------------|-----------------------|
| Install default and named instances                                                | Topic 2-A             |
| Configure SQL to use only certain CPUs (affinity masks, etc.)                      | Topic 3-B             |
| Configure server level settings                                                    | Topic 3-A             |
| Configure many databases/instance, many instances/server, virtualization           | Topic 2-A, 4-A, 13-A  |
| Configure clustered instances including MSDTC                                      | Topic 10-A            |
| Memory allocation                                                                  | Topic 3-B             |
| Database mail                                                                      | Topic 3-C             |
| Configure SQL Server engine: memory, fillfactor, sp_configure, default options     | Topic 3-B, 3-A        |
| <b>Implement a SQL Server clustered instance.</b>                                  |                       |
| Install a cluster                                                                  | Topic 10-A            |
| Manage multiple instances on a cluster                                             | Topic 10-A            |
| Set up subnet clustering                                                           | Topic 10-A            |
| Recover from a failed cluster node                                                 | Topic 10-A            |
| <b>Manage SQL Server instances.</b>                                                |                       |
| Install an instance                                                                | Topic 2-A             |
| Manage interaction of instances                                                    | Topic 11-A            |
| SQL patch management                                                               | Topic 13-A            |
| Install additional instances                                                       | Topic 2-A             |
| Manage resource utilization by using Resource Governor                             | Topic 11-A            |
| Cycle error logs                                                                   | Topic 8-C             |
| <b>Section 3: Optimize and Troubleshoot (14%)</b>                                  |                       |
| <b>Identify and resolve concurrency problems.</b>                                  |                       |
| Examine deadlocking issues using the SQL server logs using trace flags             | Topic 8-C, 12-A       |
| Design reporting database infrastructure (replicated databases)                    | Topic 13-A            |
| Monitor via DMV or other MS product                                                | Topic 11-A            |
| Diagnose blocking, live locking and deadlocking                                    | Topic 11-A, 12-A, 8-C |
| Diagnose waits                                                                     | Topic 11-A            |
| Performance detection with built in DMVs                                           | Topic 11-A            |
| Know what affects performance                                                      | Topic 11-A            |
| Locate and if necessary kill processes that are blocking or claiming all resources | Topic 11-A            |
| <b>Collect and analyze troubleshooting data.</b>                                   |                       |

| <b>Exam 70-462 Mapping Objectives</b>                                                                                                                                  | <b>Covered In</b>      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| Monitor using Profiler, collect performance data by using System Monitor, Collect trace data by using SQL Server Profiler, identify transactional replication problems | Topic 11-A, 12-A, 10-B |
| Identify and troubleshoot data access problems; gather performance metrics                                                                                             | Topic 12-B, 11-A       |
| Identify potential problems before they cause service interruptions                                                                                                    | Topic 11-A             |
| Identify performance problems, use Xevents and dmvs                                                                                                                    | Topic 12-A, 11-A       |
| Create alerts on critical server condition                                                                                                                             | Topic 9-D              |
| Monitor data and server access by creating audit and other controls                                                                                                    | Topic 6-B              |
| Identify IO vs. memory vs. CPU bottlenecks                                                                                                                             | Topic 11-A             |
| Use the Data Collector tool                                                                                                                                            | Topic 13-B             |
| <b>Audit SQL Server instances.</b>                                                                                                                                     |                        |
| Implement a security strategy for auditing and controlling the instance                                                                                                | Topic 6-B              |
| Configure an audit                                                                                                                                                     | Topic 6-B              |
| Configure server audits                                                                                                                                                | Topic 6-B              |
| Track who modified an object                                                                                                                                           | Topic 6-B              |
| Monitor elevated privileges as well as unsolicited attempts to connect                                                                                                 | Topic 6-B              |
| Policy-based management                                                                                                                                                | Topic 6-B              |
| <b>Section 4: Manage Data (19%)</b>                                                                                                                                    |                        |
| <b>Configure and maintain a back up strategy.</b>                                                                                                                      |                        |
| Manage different backup models, including point in time recovery                                                                                                       | Topic 9-A, 9-B         |
| Protect customer data even if backup media is lost                                                                                                                     | Topic 9-A              |
| Perform backup/restore based on proper strategies including backup redundancy                                                                                          | Topic 9-A              |
| Recover from a corrupted drive                                                                                                                                         | Topic 12-B             |
| Manage a multi-TB database                                                                                                                                             | Topic 13-B             |
| Implement and test a database implementation and a backup strategy (multiple files for user database and tempdb, spreading database files, backup/restore)             | Topic 9-A, 9-B, 1-B    |
| Backup a SQL Server environment                                                                                                                                        | Topic 9-A              |
| Back up system databases                                                                                                                                               | Topic 9-A              |
| <b>Restore databases.</b>                                                                                                                                              |                        |
| Restore a database secured with TDE                                                                                                                                    | Topic 7-A              |
| Recover data from a damaged DB (several errors in DBCC checkdb)                                                                                                        | Topic 12-B             |

| <b>Exam 70-462 Mapping Objectives</b>                                           | <b>Covered In</b> |
|---------------------------------------------------------------------------------|-------------------|
| Restore to a point in time                                                      | Topic 9-B         |
| File group restore                                                              | Topic 9-B         |
| Page level restore                                                              | Topic 9-B         |
| <b>Implement and maintain indexes.</b>                                          |                   |
| Inspect physical characteristics of indexes and perform index maintenance       | Topic 11-B        |
| Identify fragmented indexes                                                     | Topic 11-B        |
| Identify unused indexes; implement indexes                                      | Topic 11-B        |
| Defrag/rebuild indexes                                                          | Topic 11-B        |
| Set up a maintenance strategy for indexes and statistics                        | Topic 11-B        |
| Optimize indexes (full, filter index)                                           | Topic 11-B        |
| Statistics (full, filter) force or fix queue                                    | Topic 11-B        |
| When to rebuild vs. reorg and index                                             | Topic 11-B        |
| Full text indexes                                                               | Topic 8-A         |
| Column store indexes                                                            | Topic 8-B         |
| <b>Import and export data.</b>                                                  |                   |
| Transfer data                                                                   | Topic 9-B         |
| Bulk copy                                                                       | Topic 9-B         |
| Bulk insert                                                                     | Topic 9-B         |
| <b>Section 5: Implement Security (18%)</b>                                      |                   |
| <b>Manage logins and server roles.</b>                                          |                   |
| Configure server security                                                       | Topic 5-A         |
| Secure the SQL Server using Windows Account / SQL Server accounts, server roles | Topic 5-A, 5-B    |
| Create log in accounts                                                          | Topic 5-A         |
| Manage access to the server, SQL Server instance, and databases                 | Topic 5-B         |
| Create and maintain user-defined server roles                                   | Topic 5-B         |
| Manage certificate logins                                                       | Topic 6-A         |
| <b>Manage database permissions.</b>                                             |                   |
| Configure database security                                                     | Topic 5-C         |
| Database level, permissions                                                     | Topic 5-C         |
| Protect objects from being modified                                             | Topic 5-C         |
| <b>Manage users and database roles.</b>                                         |                   |
| Create access to server / database with least privilege                         | Topic 5-C         |
| Manage security roles for users and administrators                              | Topic 5-C         |
| Create database user accounts                                                   | Topic 5-A         |

| <b>Exam 70-462 Mapping Objectives</b>         | <b>Covered In</b> |
|-----------------------------------------------|-------------------|
| Contained logins                              | Topic 4-A         |
| <b>Troubleshoot security.</b>                 |                   |
| Manage certificates and keys                  | Topic 6-A         |
| Endpoints                                     | Topic 6-A         |
| <b>Implement High Availability (12%)</b>      |                   |
| <b>Implement AlwaysOn.</b>                    |                   |
| Implement a mirroring solution using AlwaysOn | Topic 10-C        |
| Failover                                      | Topic 10-A        |
| <b>Implement database mirroring.</b>          |                   |
| Set up mirroring                              | Topic 10-C        |
| Monitor the performance of database mirroring | Topic 10-C        |
| <b>Implement replication.</b>                 |                   |
| Troubleshoot replication problems             | Topic 10-B        |
| Identify appropriate replication strategy     | Topic 10-B        |

# Lesson Labs

Lesson labs are provided for certain lessons as additional learning resources for this course. Lesson labs are developed for selected lessons within a course in cases when they seem most instructionally useful as well as technically feasible. In general, labs are supplemental, optional unguided practice and may or may not be performed as part of the classroom activities. Your instructor will consider setup requirements, classroom timing, and instructional needs to determine which labs are appropriate for you to perform, and at what point during the class. If you do not perform the labs in class, your instructor can tell you if you can perform them independently as self-study, and if there are any special setup requirements.

# Lesson Lab 1–1

## Creating Filegroups and Partitions in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

1. Ensure that you have Administrative privileges on a working instance of SQL Server 2012.
2. Attach AdventureWorks2012 database to the working instance of SQL Server 2012.

### Scenario

As database administrator of Adventure Works Cycles, a bicycle manufacturing company, you have been asked to migrate the existing databases to a new server, which uses state-of-art technology and has advanced features. Before performing the actual migration, you decide to set up a test environment to analyze the existing server and database configurations so that the migration process is smoother. For this purpose, you decide to use a new database named PurchasesDB. In addition, due to sustained advertisement campaigns during the last three months, the business has improved and more new orders are being received. You have been asked to capture the details of the purchase orders, which have so far been maintained in a spreadsheet, in a new table named Purchase Order.

Finally, because you expect this database to grow due to the anticipated huge volumes of transactions, you need to improve the query performance. In order to achieve this, you decide to distribute query load across multiple files and drives for certain parts of the database by using filegroups and partitions to track the year-wise data.

- 
1. Launch the SQL Server Management Studio and with Windows Authentication selected as the Authentication mode, create a database named PurchasesDB and add it to the PRIMARY filegroup.
  2. Create two folders named Drive E and Drive F under C: drive to simulate different storage locations.
  3. Create a filegroup named Transactions for the PurchasesDB database and define two new pages named Trans01 and Trans02 to use C:\Drive E and C:\Drive F as their physical path.
  4. Create a new filegroup named OlderPurchaseOrders with a logical name, Old\_Purch\_Orders to select this new filegroup. Save the new page as Old\_Purch\_Orders and save it the C:\Drive F folder.
  5. Create a partition function named, Purch\_Ord\_Part\_Func(DATE) as RANGE with the values ranging from 20120101 to 20130101.
  6. Create a partition scheme named Purch\_Ord\_Year\_PartScheme as PARTITION on the filegroups, PRIMARY, Old\_Purch\_Orders, and Transactions.
  7. Create a table named PurchaseOrders with the following columns, indexes, and partition.
    - a) PurchaseOrderID of int data type and IDENTITY (1,1) with NOT NULL constraint.

- b) PurchaseOrderDate of Date Type with NOT NULL constraint.
  - c) ProductID of int type with NOT NULL constraint.
  - d) ListPrice of float data type with NOT NULL constraint.
  - e) OrderQty of int data type with NOT NULL constraint.
  - f) TotalPurchaseOrderAmount of float data type with NOT NULL constraint.
  - g) Add a Non-Clustered primary key named PK\_PurchaseOrders based on PurchaseOrderID and PurchaseOrderDate columns.
  - h) Create the table on the Purch\_Ord\_Year\_PartScheme partition based on the values in the PurchaseOrderDate column.
-

# Lesson Lab 2-1

## Deploying SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

Ensure that you have placed the SQL Server 2012 64-bit installation DVD in your DVD drive or mount the .ISO image of SQL Server 2012 64-bit from either the local hard disk or network drive.

### Scenario

As part of testing the migrating process, you want to carry out certain checks on the SQL Server without affecting the existing SQL Server instances. For this purpose, you decide to create a new instance named “Test\_Instance” so that you can switch to this instance whenever required. Further, you need only Database Engine Services and Reporting Services—Native feature on this instance. Also, as you intend to carry out replication tasks later on the server, you decide to create another instance named Test\_Instance1 with Database Engine Services and SQL Replication features.

- 
1. Launch the Installation wizard and install a new SQL instance named TEST\_INSTANCE with the Database Engine Services and Reporting Services - Native features selected.
  2. Launch the Installation wizard and install a new SQL instance named TEST\_INSTANCE1 with the Database Engine Services and SQL Replication features selected.
-

# Lesson Lab 3-1

## Configuring SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

1. You have launched SSMS and logged in as administrator to an SQL Server instance.
2. You have selected a Query Editor pane.

### Scenario

As the database administrator of Adventure Works Cycles, you are asked to configure the server memory to improve the overall performance of queries. You feel that increasing the minimum server memory would help improve the query performance. You decide to configure the minimum server memory using the sp\_configure procedure. Further, you need to ensure that the index pages are organized better by allocating more free space. You decide to increase the fill factor to achieve this.

- 
1. Run the sp\_configure stored procedure and view the advanced configuration options of the server.



**Note:** If you are trying out this lab outside of the classroom setup, where you have performed the activity of Lesson 3, you need to run the sp\_configure stored procedure with two parameters, "show advanced options" and "1" followed by the RECONFIGURE command to display the advanced server configuration options.

2. Run the sp\_configure stored procedure with two parameters, "min server memory (MB)" and "2" followed by the RECONFIGURE command to set the minimum server memory as 2 MB.
  3. Set the minimum server memory as 2 MB using the required sp\_configure stored procedure parameters followed by the RECONFIGURE command.
  4. Run the sp\_configure stored procedure again to verify that the minimum server memory has been set correctly in the config\_value column.
  5. Modify the properties of the server to set the fill factor value of Database Settings properties to 75%.
-

# Lesson Lab 4-1

## Managing Databases in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

You have launched SSMS and logged in as administrator to the SQL Server instance that contains the SocialEventsDB database.

### Scenario

Management of Adventure Works Cycles has decided to sponsor social events periodically to make their brand more visible to the general public. For this purpose, they decide to create a new department named SocialEvents. You are asked to create a database named SocialEventsDB to stores the details of the events that are sponsored by the organization.

You need to ensure database performance by using partial containment type and full recovery model for the database. Management also wants to ensure that the SocialEventsDB is isolated from other databases in their network. You decide to migrate the SocialEventsDB database as a contained database, so that it will remain isolated from other databases. Further, you need to optimize the performance of the database using the auto close and auto shrink features.

- 
1. Modify the properties of the SocialEvents database to enable the Contained Databases option.
  2. Change the SocialEventsDB database containment type to partial and the database recovery model to Full.
  3. Enable the Auto Close and Auto Shrink options for the SocialEvents DB database.
-

# Lesson Lab 5-1

## Creating Server Account in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

1. You have logged in as Administrator of Windows Server and you have administrative privileges on SQL Server 2012.
2. You have enabled mixed mode authentication on SQL Server 2012 and attached AdventureWorks2012 database.

### Scenario

The Sales team of Adventure Works Cycles wants you to create a new user account for JohnS on Windows Server and SQL Server by performing the following tasks:

1. You have to create windows account for a user, John Smith, with the user name JohnS and password as pass@word1.
2. Grant database access to JohnS on the Windows Server.
3. Login as JohnS on Windows Server, and attempt to access SSMS and connect to SQL Server database as JohnS.
4. As Administrator, create a new login account for JohnS on SQL Server with Windows Authentication and PurchasesDB as the default database.
5. Add JohnS as a new database user for the PurchasesDB database.
6. As JohnS, verify that the changes have taken effect.

- 
1. In the Local Users and Groups tool, create a new user account for John Smith using the credentials indicated in the scenario.
  2. Assign permissions to make JohnS as member of the Windows Administrators group.
  3. Login to Windows as JohnS and connect to the TEST\_INSTANCE using the Windows Administrator credentials.
  4. Login to Windows as Administrator and create a new login account for JohnS to use Windows Authentication and connect by default to PurchasesDB database.
  5. Create a new user account, JohnS, of type Windows user, to the PurchasesDB database.
  6. Login as JohnS and verify that database connection can be established and the PurchasesDB database is accessible.
-

# Lesson Lab 6-1

## Creating Certificate Based Logins and Server Auditing SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

1. You have launched SSMS and logged in as administrator to an SQL Server instance that contains the AdventureWorks2012 database.
2. You have selected a Query Editor pane.

### Scenario

The management team of Adventure Works Cycles wants you to create certificate-based logins and server audit as part of implementing advanced security measures. You have to perform the tasks mentioned below in order to implement advanced security measures.

1. Create a new certificate named MyCertificate for JohnS that is valid until 01/01/2015.
2. Create a new login for JohnS that maps to MyCertificate.
3. Backup the certificate and the private key to the C:\Temp\JohnSCert.cer and C:\Temp\JohnSPvtKey.key files, respectively. Encrypt with a password Pass@word123.
4. Create a new server audit for the current month named in the format ServerAudit<Month><Year> to audit Fail operation. Save the file in C:\Temp.



**Note:** Replace <Month><Year> in the audit name with the current month and year.

5. Create a new server audit specification for the current month named in the format ServerAudit<Month><Year>\_LoginFailures and select ServerAudit<Month><Year> as the server audit object and FAILED\_LOGIN\_GROUP as the audit action type.



**Note:** Replace <Month><Year> in the audit name and audit specification with the current month and year.

6. Enable the server audit and server audit specification created and attempt to log in to the database with an incorrect login or password to populate the server log and view the audit log for the related entry.

# Lesson Lab 7-1

## Backing Up and Compressing a Database in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

You have two running instances of SQL Server 2012 on the physical system and AdventureWorks2012 database is contained in only one of the server instance.

### Scenario

As database administrator, you have been assigned the following database optimization tasks:

1. The Production team wants the PurchasesDB database to be backed up using Full Backup option on the current server.
  2. After backup, the database has to be restored on another instance on the same physical server.
  3. Compress the Purchasing.PurchaseOrderDetail table in the AdventureWorks2012 database based on ROW using a stored procedure and schema name.
  4. Rebuild the partitions and verify that the compression was successfully applied on the Purchasing.PurchaseOrderDetail table.
- 
1. Take a full backup of the PurchasesDB to C:\Temp folder. Name the backup file as PurchasesDB.bak.
  2. Restore PurchasesDB database from TEST\_INSTANCE to TEST\_INSTANCE1 server.
  3. In the AdventureWorks2012 database, compress the PurchaseOrderDetail table based on ROW using sp\_estimate\_data\_compression\_savings and Purchasing schema.
  4. Rebuild all the partitions of the Purchasing.PurchaseOrderDetail table with the ROW level data compression and verify the changes in the properties dialog box of the PurchaseOrderDetail table.

# Lesson Lab 8-1

## Working with Indexes in SQL Server 2012

**Activity Time:** 20 minutes

### Before You Begin

1. You have launched SSMS and logged in as administrator to an SQL Server instance that contains the AdventureWorks2012 database.
2. You have selected a Query Editor pane.

### Scenario

You have been asked to manage indexes by performing the following tasks:

1. The Production team of Adventure Works Cycles wants their team to be able to search the product models by using any word in name column. You need to create a Full Text index named NewFTC\_Product based on the primary key of the Product Model table.
2. After setting up the Full Text Catalog index, you need to query for product models that contain specific words in their names.
3. Include the Product table into the newly created NewFTC\_Productindex and rebuild the full text catalog.
4. Copy all records from the Purchasing.PurchaseOrderDetail table into a new table named Purchasing.PurchaseOrderDetailForAnalysis.
5. Modify the properties of the Purchasing.PurchaseOrderDetailForAnalysis table to set the combination of PurchaseOrderID and PurchaseOrderDetailID columns as the primary key.
6. Create a column store index named PurchasePriceAndQty on the OrderQty and UnitPrice columns.
7. Execute the SELECT statement to find the performance of the PurchaseOrderDetailForAnalysis and PurchaseOrderDetail tables.

1. Setup a Full-Text Index on the Production.ProductModel table in the AdventureWorks2012 database on the primary key index PK\_ProductModel\_ProductModelID. Select the Name column and select English as the Language for Word Breaker option. Retain the default change tracking option and specify NewFTC\_Product as the new catalog name.
2. Enter the query to locate product models that contain the words “Road” or “Tour” in their names.
3. Modify the properties of the NewFTC\_Product index and include the Production.ProductModel table. Select the Name column of the ProductModel table as the eligible column. Rebuild the catalog.
4. Use the SELECT \* INTO statement to create the PurchaseOrderDetailForAnalysis table with the records from the PurchaseOrderDetail table.
5. Modify the table design suitably to set the composite primary key.
6. Create the new column store index using the required columns from the PurchaseOrderDetailForAnalysis table.

7. Create two SELECT queries to select Sum(OrderQty) from both the tables and execute the query with Actual Execution plan.
-

## Lesson Lab 9-1

### Backing up/Restoring and Exporting Database in SQL Server 2012

**Activity Time:** 15 minutes

#### Before You Begin

You have launched SSMS and logged in as administrator to an SQL Server instance that contains the AdventureWorks2012 database.

#### Scenario

You have been assigned certain database administration tasks.

1. Take a full compressed backup of PurchasesDB database of the c:\Temp folder after setting the reliability options for verifying backup and performing checksum.
  2. Restore the PurchasesDB database from the backup set to another instance of SQL Server.
  3. Export the PurchaseOrders table from PurchasesDB to C:\Temp
- 
1. Backup the PurchasesDB database using the full backup and compressed option. Also, set the reliability options mentioned in the scenario. Save the backup as PurchasesDB.bak in the C:\Temp folder.
  2. Restore the PurchasesDB database to the Test\_Instance server using the PurchasesDB.bak file.
  3. Export the PurchaseOrders table to Purch\_Orders.csv in C:\Temp.
-

# Lesson Lab 10-1

## Replicating and Database Mirroring in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

1. Apart from the default server instance, you have another working instance of SQL Server.
2. The default instance contains both PurchasesDB and SalesDB databases.

### Scenario

You are asked to replicate PurchasesDB database so that it will be available on more than one server instance. Also, you need to mirror SalesDB database such that the same updates are available on multiple server instances.

- 
1. Verify that the default instance of SQL Server is configured as the distributor by invoking the Configure Distribution Wizard and publish PurchasesDB for replication. Configure the security settings using the Administrator credentials.
  2. From Test\_Instance1 subscribe to the SQL01 publication. Configure the publication settings for push subscription. Configure distribution agent security settings using Administrator credentials. Initialize the subscription.
  3. Create a mirror instance of the SalesDB database between SQL01 and Test\_Instance1 server instances after configuring SQL01 as the principal server and Test\_Instance1 as the mirror server.
-

# Lesson Lab 11-1

## Optimizing Indexes in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

You have launched SSMS and logged in as administrator to an SQL Server instance that contains the AdventureWorks2012 database.

### Scenario

You need to create a new index on the Size column in the Production.Product table to address a ticket reporting performance issues. You decide to create a non-clustered index on the size column and modify the fill value and modify the index statistics settings.

---

1. Add a non-clustered index named AK\_Size based on the size column.
  2. Set the fill factor of the index as 85% and rebuild the index.
  3. Modify the statistics page of the AK\_Size index to enable the statistics update for the selected columns.
-

# Lesson Lab 12-1

## Checking Database and Creating a Trace in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

1. You have launched SSMS and logged in as administrator to an SQL Server instance that contains the AdventureWorks2012 database. You have open a Query Editor pane.
2. Launch SQL Profiler and connect to the same SQL Server instance.

### Scenario

The Production department has an important upcoming event for which they have asked you to ensure that PurchasesDB is error-free. Further, you have been asked to create a new trace that will accommodate a larger trace output on the Test\_Instance server.

- 
1. Run the DBCC Check command on the PurchasesDB database.
  2. Create a new trace named Longer\_Trace of Size 25 MB on the Test\_Instance server. Verify the trace contents by connecting to Test\_Instance as Administrator from SSMS and disconnecting from the server.
-

# Lesson Lab 13-1

## Managing Database in SQL Server 2012

**Activity Time:** 15 minutes

### Before You Begin

You have launched SSMS and logged in as administrator to an SQL Server instance that contains the AdventureWorks2012 database. Ensure that there is another running instance of SQL Server on the physical system.

### Scenario

The Production department wants you to backup PurchaseDB as a package from one server and create an additional copy in another server to ensure better availability of the database. Further, for analysis purpose, they want you to export the contents of the Production.ProductSubCategory table as a delimited text file.

- 
1. Export the PurchaseDB database from Test\_Instance as Purchases.Bacpac to C:\Temp.
  2. From Test\_Instance1 server, import the C:\Temp\Purchases.Bacpac file to create an additional copy of the database.
  3. Export the contents of Production.ProductSubCategory table as a delimited text file named Prod\_Category.csv and save it in the C:\Temp folder.
-

# Glossary

**active directory**

The directory that monitors all changes and updates of all the accounts and passwords within an organization.

**administrator**

A user who has full access across all accounts.

**agent job**

A series of steps that execute a task.

**alert**

A message sent to user when an event occurs on the server.

**AlwaysOn**

A new feature introduced in SQL Server 2012 that provides high availability and disaster recovery capabilities at the enterprise level.

**asymmetric encryption**

A type of encryption that uses two different keys for encrypting and decrypting data.

**asymmetric key**

A key of a pair of keys used in asymmetric encryption.

**audit action**

An action for which the audit information is logged.

**auto statistic**

The statistic created automatically for all non-indexed columns in a query when a query that uses a column index is fired.

**availability group**

A group of SQL Server nodes that work together.

**availability mode**

A property of an availability group that determines whether the primary availability group waits for a secondary availability group to commit its transactions on the disk.

**backup**

A copy of the server data.

**backup redundancy**

A backup strategy to ensure that a backup is also backed up.

**BCP**

(BULK COPY Program) A command line tool that allows you to export and import data.

**benchmark**

The standard tests that are run on systems to study their performance.

**bottlenecks**

The external and internal factors, that can affect system performance.

**BULK INSERT**

A T-SQL command that allows you to insert a large number of rows at a time into a table.

**CA**

(Certificate Authority) A universally trusted source that issues certificates to prove the

identity of objects such as users, databases, servers, or services.

### **capacity planning**

A task that database administrators perform to estimate the expected load on a database or server and plan a server deployment according to the estimate.

### **certificate**

A digital identity that is assigned to objects such as users, databases, servers or services by a universally trusted source called the Certification Authority (CA).

### **certificate login**

A login associated with a certificate to ensure that the user is a valid user.

### **cluster**

A group of independent servers that provide a set of services as a single unit.

### **Clustered Instance**

An instance of SQL Server that is installed on a node in a cluster.

### **CMS**

(Central Management Server) A server that helps administrators to centrally manage multiple registered servers as a single group.

### **column store index**

A property that is used to store column data for the columns mentioned in the index, instead of storing rows of data in the index.

### **contained login**

A login that exists only within the contained database without having any reference in the SQL Server instance.

### **copy-only backup**

A backup that is not included in the backup sequence.

### **credentials**

A record in SQL Server that stores authentication information for an external resource.

### **custom server role**

A role in which an administrator can create and assign any permission to.

### **cycling log file**

A process that is done when you want to use multiple log files on each database.

### **data collector**

A component that retrieves data from different systems on either a constant or pre-defined schedule.

### **data compression**

A process of reducing the size of your database data.

### **data-tier application**

A self-contained SQL Server entity that defines the database objects and instance objects associated with a database.

### **database audit specification**

An object that tracks parameters at database level.

### **database backup**

A copy of the database in a server.

### **database level role**

A role that is created at the database level.

### **database mirroring**

A high availability technique in which two copies of a database are maintained on different SQL Server instances.

### **database permissions**

The permissions that are given to a database role or a single user on the database.

### **database recovery**

The process of recovering the data from failed, corrupted, or inaccessible databases.

### **DBCC**

(Database Console Commands) A set of commands that are executed in the database engine and SQL Server to return the results of the execution back to the user.

**DBCC SHRINKFILE**

A command used to shrink a log file.

**deadlock**

A special error condition in SQL Server where two different applications or connections hold locks on two separate objects and are waiting for the object held by the other to be released.

**default instance**

The default installation of the database engine that is set up when SQL Server is installed.

**DENY statement**

The statement that allows you to take away permissions that were given to the user logins.

**DMK**

(Database Master Key) A symmetric key used to protect asymmetric keys and private keys in the database.

**DMV**

(Dynamic Management Views) A special system views that allows you to query the system state of the server and monitor the different server parameters, check health, diagnose issues, and tune up the performance.

**drive**

A physical storage device on which data in SQL Server databases is stored.

**endpoint**

The object that is used to connect SQL Server using protocols such as SOAP, XML Web Services, and WCF.

**facet**

An object that contains the properties to govern a policy.

**failover**

The process in which when a node in a cluster fails, the services provided by that node are switched to another available node or availability group.

**failover mode**

A property of an availability group that determines whether failover occurs automatically or manually.

**FIFO**

(First In First Out) A backup redundancy strategy that uses multiple media sets and rotates them over a period of time.

**file space**

The space occupied by a database file on the disk.

**filegroups**

A collection of files that is organized based on storage allocation and administration considerations to help database administrators manage a set of files of a database on different drives.

**FILESTREAM**

A file object that is stored on the file system but referenced within a special VARBINARY(MAX) column with the FILESTREAM option.

**FileTable**

A feature that allows administrators to point to a standard file system folder such as the hard disk or Universal Naming Convention (UNC) path.

**fixed database role**

A role that is created at the database level.

**fixed server role**

A pre-defined role within SQL Server, which cannot be deleted or modified in any manner other than adding accounts to it.

**fragmented index**

An index that contains empty space in some pages or has split pages, which cause the logical order of the index to differ from its physical location in the database.

**FTI**

(Full Text Indexing) A property that allows you to search for a particular word or phrases in a data table.

**Full Text Catalog**

A grouped set created by one or more catalogs in a database and adding Full Text Indexes in it.

**Full Text Search**

An SQL Server 2012 feature that enables users to execute complex text-based queries on character-based data in tables.

**Grandfather–Father–Son**

A backup redundancy strategy that uses three sets of backups and graduates the latest backup of one set to the other set.

**GRANT statement**

The statement that allows you to give certain permissions to a login on the specified resources by authorizing the login to have the permission to grant other users with the permissions as well.

**horizontal scaling**

A scalability method that involves adding hardware to distribute load and increase the capabilities of each server in the setup.

**hypervisor**

An application that manages multiple virtual machines on a single physical machine.

**index**

A property that allows a database to find files in the table without searching the entire table.

**job history**

A log maintained for each job defined in the system.

**least privilege access**

The rule applied to determine the access of an individual login for each object.

**live lock**

A situation where a request for an exclusive lock on a resource is continuously denied to a process since overlapping shared locks keep getting access.

**log file**

A recording of the process replaying back and forth in the server.

**login account**

An account created for every user within the network to monitor the activities performed within a database.

**LSN**

(Log Sequence Number) A unique id that maintains the position of the last current transaction that can be committed or rolled back.

**Microsoft WSUS**

A component that manages the download and installation of updates and patches on Windows operating systems.

**migrating logins**

The practice of moving a database from one server to another, along with the logins in the database.

**MSFC**

(Multi Subnet Failover Clustering) An SQL Server 2012 feature that enables you to setup a cluster in such a way that each node in the cluster is connected to a different IP subnet or some nodes are in a single subnet and the other nodes are in other subnets.

**Multi–TB**

(multiple terabyte) A database that stores terabytes of data.

**named instance**

An additional instance of the database engine that is set up on the same system on which the default instance is installed.

**node**

A server in a cluster that accesses a shared storage

**noise words**

The common set of words that really don't need to be ever searched or used.

**notification**

An alert that is based on some system events to database operators.

**page-level compression**

A database compression method used to compress the storage of data on a page.

**partition**

The basic unit of organization in the physical implementation of tables and indexes that enable you to manage subsets of data.

**patch**

An update to an application that resolve bugs or issues identified in the application after its release.

**PBM**

(Policy Based Management) A feature of SQL Server that enables administrators to ensure that different component configurations of the server follow the prescribed standards.

**PerfMon**

(System Performance Monitor) A built-in tool of Windows that helps benchmarking and monitoring system performance.

**point in time snapshot**

A mirror copy of the database for a particular point in time.

**primary availability group**

The current availability group in a cluster that services requests.

**private key**

An asymmetric key in a certificate that is known only to the user of the certificate.

**proxy account**

An account that has permissions to perform job steps in a subsystem.

**public key**

An asymmetric key in a certificate that is shared with anyone who requests for the certificate.

**publisher**

A SQL Server instance that publishes data and database objects to be replicated.

**replication**

A SQL Server feature that copies and distributes database objects and data from one database server to another.

**Resource Governor**

A feature of SQL Server that allows you to manage the SQL Server's workload by specifying limits on CPU and memory that are available for client requests.

**resource group**

A group of resources that are shared.

**restore**

A process of using a database backup to recover a failed database.

**round-robin**

An approach to balance the log files in a database.

**row-level compression**

A database compression method used to compress the storage of data within rows.

**scalability**

A factor that determines the performance of applications when their capacity is increased by increasing the number of resources the server uses.

**secondary availability group**

An availability groups in a cluster that is not currently active.

**security model**

A model that is based on granting and denying permissions on logins.

**server audit**

An object that enables auditing at server instance and database levels.

**server audit specification**

An object that audits events at the server instance level.

**server role**

A role that groups multiple users under a single name so that the settings applied to the particular group apply to all the users within that group as well.

**service broker**

An SQL Server 2012 component that provides queuing and messaging capabilities to the database and enables client applications to send and receive asynchronous messages.

**SID**

A structure used to identify users distinctively.

**slipstreaming**

A patch management technique that integrates updates, patches, and service packs into the base installation files stored in the installation media and performs a single step installation of the updated software.

**SMK**

(Service Master Key) An auto generated symmetric key to encrypt other keys, linked server passwords, and credentials in SQL Server.

**SQL profiler**

A built-in tool of SQL Server that allows you to create and manage traces.

**SQL Server account**

A login account that exists only within the server instance and cannot login to Windows itself.

**SQL Server Agent**

The SQL Server component that is used to setup administrative tasks.

**SSD**

(Solid State Drives) A storage device that uses flash memory to store data.

**SSIS**

(SQL Server Integration Services) A component of SQL Server used to perform Extract, Transform, Load (ETL) processes on data.

**statistics**

The information stored about the distribution of values in columns of a table based on which indexes are created. The query optimizer uses statistics for estimating the number of output rows which in turn forms the basis of the query plan of the SQL Server.

**stop words**

A set of discarded strings that help the FTI from being over loaded with the unnecessary word searches.

**subscriber**

A SQL Server instance that receives the data and database objects published by the publisher.

**subsystem**

A listing of different types of job steps.

**tail-log backup**

A backup taken when a database is non-functional and there are transactions after the last backup.

**TDE**

(Transparent Data Encryption) A feature of SQL Server 2012 that allows data to be encrypted and decrypted on the fly by SQL Server.

**trace data**

A collection of events and data returned by the database engine.

**trace flag**

A value used for tracing certain issues that might occur in SQL Server.

**UNC**

A string that allows administrators to identify the location on a standard file system.

**user defined server role**

A built-in role that makes it easier to grant or revoke access to a server at any point in time.

**vertical scaling**

A scalability method that involves adding resources such as memory, processors, and storage, to a server to increase the capacity at which it runs.

**virtualization**

A technology that is used to create and manage processes in virtual machines.

**VLF**

(Virtual Log Files) A set of equal size “blocks” within the real log file which are allocated and reused as and when transactions are added and truncated.

**Windows user**

The actions that users can perform in Windows.

**workload**

A uniform set of queries that limit the amount of CPU and memory allocated.

## **WSFC**

(Windows Server Failover Clustering) A Windows Server technology that enables multiple servers to work together as if they were a single server.

## **XEvents**

(Extended Events) A system feature that lets you query the SQL Server engine to gather as much or as little information as you require to troubleshoot any issues.



# Index

## A

Active Directory [96](#)  
administrator [96](#)  
affinity masks  
    64-bit [46](#)  
agent jobs [259](#)  
alerts [148](#)  
AlwaysOn [307](#)  
asymmetric encryption [132](#)  
asymmetric keys [132](#)  
Attach and Detach  
    comparison [240](#)  
audit actions [142](#)  
audit action types  
    database level [143](#)  
    server level [143](#)  
authentication modes [23](#)  
Auto statistics [352](#)  
availability modes [281](#)

## B

backup redundancy  
    First In First Out [232](#)  
    Grandfather-Father-Son [233](#)  
backups [230](#)  
backup strategy selection  
    guidelines [233](#)  
benchmarks [326](#)  
bottlenecks [330](#)

## C

capacity planning [5](#)  
Central Management Server [319](#)  
certificate logins [134](#)

certificates  
    exporting [135](#)  
clustered instances  
    benefits [282](#)  
    failover in [282](#)  
clusters [278](#)  
CMS, *See* Central Management Server  
column store indexes [199](#)  
commands  
    BULK COPY [248](#)  
    BULK INSERT [248](#)  
    DBCC SHRINKFILE [215](#)  
    sp\_configure [40](#)  
connectivity issues  
    guidelines to test [367](#)  
contained databases [75](#)  
credentials [262](#)  
cycle error logs  
    need [216](#)  
cycling log files [216](#)

## D

data access problems  
    guidelines to identify and troubleshoot  
        [376](#)  
database audit specification [145](#)  
database backup [230](#)  
Database Console Commands [362](#)  
database corruption  
    possible reasons [376](#)  
Database Engine  
    service account [23](#)  
database level roles [116](#)  
Database Mail [59](#)  
database mail configuration [60](#)

database mirroring  
 disadvantages 307  
 database permissions 117  
 database recovery 376  
 data collector 413  
 data compression 180  
 data from a corrupted hard drive  
 guidelines to recover 376

Data-tier Application  
 need for 390  
 operations 389

DBCC, *See* Database Console Commands

DBCC commands  
 categories 362  
 informational 363  
 maintenance 363  
 miscellaneous 364  
 validation 363  
 deadlock issues 218  
 default instances 5  
 default server options 42  
 DMK, *See* Database Master Key  
 DMVs, *See* Dynamic Management Views  
 drives 9  
 Dynamic Management Views 328

## E

encryption levels 169  
 endpoints 136  
 evaluation modes 148  
 Extended Events 365

## F

facets 146  
 failover  
 types 280  
 failover mode 281  
 FIFO, *See* First In First Out  
 file group restores 241  
 filegroups  
 advantages 11  
 data movement 87  
 files 9  
 file space 85  
 FileTable 201  
 fill factor 50  
 fixed database roles 117  
 fragmented indexes 353  
 FTI  
 features 189

full database recovery model process 83  
 Full Text Catalog 189  
 Full Text Indexing 189  
 full text querying 189  
 Full Text Search 3  
 functions  
 FILEPROPERTY 86

## H

hypervisor 385

## I

implementing Windows and Service Level Security  
 guidelines 22  
 Import and Export Wizard 247  
 indexes 188  
 installation methods 27  
 instances 5

## J

job history 260

## L

least privilege access 119  
 log file growth properties 216  
 log files 215  
 Login accounts 97  
 Log Sequence Number 215  
 LSN, *See* Log Sequence Number

## M

mail accounts and profiles 59  
 mail profiles 59  
 memory allocation 49  
 methods  
 DBCC SHOWCONTIG 354  
 DMV 355  
 Microsoft WSUS 387  
 migrating databases  
 guidelines 34  
 migrating hardware  
 guidelines 34  
 migrating logins 120  
 min memory per query 42  
 models  
 bulk logged recovery 84  
 full recovery 83

simple recovery 83

MSFC, *See* Multi Subnet Failover Clustering

Multiple terabyte databases 413

Multi Subnet Failover Clustering 282

multi-TB databases, *See* Multiple terabyte databases

## N

Named instances 5

noise words 189

notifications 61

NUMA 49

## O

object modification

    protection against 118

OLTP, *See* Online Transaction Processing

System

Online Transaction Processing System 388

options

    Auto Close 82

    Auto Shrink 82

    data compression 181

    FILESTREAM 200

## P

page-level compression 180

page level restores 241

partitions 11

patches 385

patch management

    guidelines to implement 387

PBM

    Policy Based Management 146

PerfMon, *See* System Performance Monitor

physical structure of databases 74

platform components 2

Point in Time Snapshots 241

primary availability groups 279

Private Key 132

proxy accounts 263

Public Key 132

publisher 295

## R

recovery models

    using 84

remote access 26

replication

types 296

replication process 295

replication strategy selection

    guidelines 296

reporting database infrastructure configuration process 388

resource and availability groups 278

Resource Governor 327

restore 240

round-robin 216

row-level compression 180

## S

scalability methods 4

secondary availability groups 279

security models 97

security strategy

    guidelines 149

select hardware

    guidelines 6

server audit 141

server audit specification 144

Server Core mode

    installation 25

server-level settings 41

server memory options 49

Server performance

    factors that affect 329

server roles

    custom 110

    fixed 109

    user-defined 111

service broker 2

SMK, *See* Service Master Key

SMTP 59

*See also* Simple Mail Transfer Protocol

snapshot restrictions 241

solid state drives 9

SQLIO tools 26

SQL Profiler 329

SQL Server accounts 96

SQL Server affinity 46

SQL Server Agent 259

SQL Server agent security 261

SQL Server Database Engine Instance

Configuration 24

SQL Server Import and Export wizard 34

SQL Server installation

    guidelines 5, 22

SQL Server installation requirements 4

SQL Server Integration Services 412

SQL Server patch management [386](#)  
 SSIS  
*See also* SQL Server Integration Services  
 security [412](#)  
*See also* SQL Server Integration Services  
 statements  
   ALTER SERVER CONFIGURATION  
[48](#)  
   CREATE PARTITION FUNCTION [12](#)  
   CREATE PARTITION SCHEME [13](#)  
   DENY [98](#)  
   GRANT [97](#)  
   statistics [351](#)  
   Statistics Object [352](#)  
   stop words [189](#)  
   stored procedures  
     sp\_send\_dbmail [60](#)  
   strategies to protect data [234](#)  
   subscriber [295](#)  
   symmetric keys [133](#)  
   system databases [74](#)  
   System Performance Monitor [326](#)  
   system stored procedures  
     sp\_estimate\_data\_compression\_savings  
[180](#)

**T**

Tail-Log backup [240](#)  
 TDE  
   Transparent Data Encryption [168](#)  
 trace data [364](#)  
 Trace Flags [217](#)  
 type of storage  
   guidelines [6](#)

**U**

UNC, *See* Universal Naming Convention  
 Universal Naming Convention [201](#)  
 user databases [75](#)  
 user login accounts  
   types [23](#)

**V**

virtualization [384](#)  
 Virtual Log Files [215](#)  
 VLFs, *See* Virtual Log Files

**W**

Windows accounts [96](#)

Windows Server Failover Clustering [278](#)  
 Windows users [96](#)  
 workloads [327](#)  
 WSFC, *See* Windows Server Failover Clustering

**X**

XEvents, *See* Extended Events



0931081 rev 1.1  
ISBN-13 978-1-4246-2151-4  
ISBN-10 1-4246-2151-8

A standard linear barcode is positioned horizontally. To its right, the number "9 0000" is printed vertically.

9 781424 621514