Microsoft®

OFFICIAL MICROSOFT LEARNING PRODUCT

6234A

Implementing and Maintaining Microsoft® SQL Server® 2008 Analysis Services

Be sure to access the extended learning content on your Course Companion CD enclosed on the back cover of the book.

Product Number: 6234A

Part Number: X15-01560

Released: 11/2008

MICROSOFT LICENSE TERMS
OFFICIAL MICROSOFT LEARNING PRODUCTS - TRAINER EDITION – Pre-Release and Final Release Versions

These license terms are an agreement between Microsoft Corporation and you. Please read them. They apply to the Licensed Content named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,
- supplements,
- Internet-based services, and
- support services

for this Licensed Content, unless other terms accompany those items. If so, those terms apply.

By using the Licensed Content, you accept these terms. If you do not accept them, do not use the Licensed Content.

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

1. **DEFINITIONS.**

   a. **"Academic Materials"** means the printed or electronic documentation such as manuals, workbooks, white papers, press releases, datasheets, and FAQs which may be included in the Licensed Content.

   b. **"Authorized Learning Center(s)"** means a Microsoft Certified Partner for Learning Solutions location, an IT Academy location, or such other entity as Microsoft may designate from time to time.

   c. **"Authorized Training Session(s)"** means those training sessions authorized by Microsoft and conducted at or through Authorized Learning Centers by a Trainer providing training to Students solely on Official Microsoft Learning Products (formerly known as Microsoft Official Curriculum or "MOC") and Microsoft Dynamics Learning Products (formerly know as Microsoft Business Solutions Courseware). Each Authorized Training Session will provide training on the subject matter of one (1) Course.

   d. "**Course**" means one of the courses using Licensed Content offered by an Authorized Learning Center during an Authorized Training Session, each of which provides training on a particular Microsoft technology subject matter.

   e. **"Device(s)"** means a single computer, device, workstation, terminal, or other digital electronic or analog device.

   f. **"Licensed Content"** means the materials accompanying these license terms. The Licensed Content may include, but is not limited to, the following elements: (i) Trainer Content, (ii) Student Content, (iii) classroom setup guide, and (iv) Software. There are different and separate components of the Licensed Content for each Course.

   g. **"Software"** means the Virtual Machines and Virtual Hard Disks, or other software applications that may be included with the Licensed Content.

   h. **"Student(s)"** means a student duly enrolled for an Authorized Training Session at your location.

i. **"Student Content"** means the learning materials accompanying these license terms that are for use by Students and Trainers during an Authorized Training Session. Student Content may include labs, simulations, and courseware files for a Course.

j. **"Trainer(s)"** means a) a person who is duly certified by Microsoft as a Microsoft Certified Trainer and b) such other individual as authorized in writing by Microsoft and has been engaged by an Authorized Learning Center to teach or instruct an Authorized Training Session to Students on its behalf.

k. **"Trainer Content"** means the materials accompanying these license terms that are for use by Trainers and Students, as applicable, solely during an Authorized Training Session. Trainer Content may include Virtual Machines, Virtual Hard Disks, Microsoft PowerPoint files, instructor notes, and demonstration guides and script files for a Course.

l. **"Virtual Hard Disks"** means Microsoft Software that is comprised of virtualized hard disks (such as a base virtual hard disk or differencing disks) for a Virtual Machine that can be loaded onto a single computer or other device in order to allow end-users to run multiple operating systems concurrently. For the purposes of these license terms, Virtual Hard Disks will be considered "Trainer Content".

m. **"Virtual Machine"** means a virtualized computing experience, created and accessed using Microsoft® Virtual PC or Microsoft® Virtual Server software that consists of a virtualized hardware environment, one or more Virtual Hard Disks, and a configuration file setting the parameters of the virtualized hardware environment (e.g., RAM). For the purposes of these license terms, Virtual Hard Disks will be considered "Trainer Content".

n. **"you"** means the Authorized Learning Center or Trainer, as applicable, that has agreed to these license terms.

2. **OVERVIEW.**

Licensed Content. The Licensed Content includes Software, Academic Materials (online and electronic), Trainer Content, Student Content, classroom setup guide, and associated media.

License Model. The Licensed Content is licensed on a per copy per Authorized Learning Center location or per Trainer basis.

3. **INSTALLATION AND USE RIGHTS.**

a. **Authorized Learning Centers and Trainers: For each Authorized Training Session, you may:**

i. either install individual copies of the relevant Licensed Content on classroom Devices only for use by Students enrolled in and the Trainer delivering the Authorized Training Session, provided that the number of copies in use does not exceed the number of Students enrolled in and the Trainer delivering the Authorized Training Session, **OR**

ii. install one copy of the relevant Licensed Content on a network server only for access by classroom Devices and only for use by Students enrolled in and the Trainer delivering the Authorized Training Session, provided that the number of Devices accessing the Licensed Content on such server does not exceed the number of Students enrolled in and the Trainer delivering the Authorized Training Session.

iii. and allow the Students enrolled in and the Trainer delivering the Authorized Training Session to use the Licensed Content that you install in accordance with (ii) or (ii) above during such Authorized Training Session in accordance with these license terms.

i. Separation of Components. The components of the Licensed Content are licensed as a single unit. You may not separate the components and install them on different Devices.

ii. Third Party Programs. The Licensed Content may contain third party programs. These license terms will apply to the use of those third party programs, unless other terms accompany those programs.

b. **Trainers:**

i. Trainers may Use the Licensed Content that you install or that is installed by an Authorized Learning Center on a classroom Device to deliver an Authorized Training Session.

ii. Trainers may also Use a copy of the Licensed Content as follows:

A. Licensed Device. The licensed Device is the Device on which you Use the Licensed Content. You may install and Use one copy of the Licensed Content on the licensed Device solely for your own personal training Use and for preparation of an Authorized Training Session.

B. Portable Device. You may install another copy on a portable device solely for your own personal training Use and for preparation of an Authorized Training Session.

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

a. **Pre-Release Licensed Content.** This Licensed Content is a pre-release version. It may not contain the same information and/or work the way a final version of the Licensed Content will. We may change it for the final, commercial version. We also may not release a commercial version. You will clearly and conspicuously inform any Students who participate in each Authorized Training Session of the foregoing; and, that you or Microsoft are under no obligation to provide them with any further content, including but not limited to the final released version of the Licensed Content for the Course.

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

c. **Confidential Information.** The Licensed Content, including any viewer, user interface, features and documentation that may be included with the Licensed Content, is confidential and proprietary to Microsoft and its suppliers.

i. **Use.** For five years after installation of the Licensed Content or its commercial release, whichever is first, you may not disclose confidential information to third parties. You may disclose confidential information only to your employees and consultants who need to know the information. You must have written agreements with them that protect the confidential information at least as much as this agreement.

ii. **Survival.** Your duty to protect confidential information survives this agreement.

iii. **Exclusions**. You may disclose confidential information in response to a judicial or governmental order. You must first give written notice to Microsoft to allow it to seek a protective order or otherwise protect the information. Confidential information does not include information that

- becomes publicly known through no wrongful act;
- you received from a third party who did not breach confidentiality obligations to Microsoft or its suppliers; or
- you developed independently.

d. **Term**. The term of this agreement for pre-release versions is (i) the date which Microsoft informs you is the end date for using the beta version, or (ii) the commercial release of the final release version of the Licensed Content, whichever is first ("beta term").

e. **Use**. You will cease using all copies of the beta version upon expiration or termination of the beta term, and will destroy all copies of same in the possession or under your control and/or in the possession or under the control of any Trainers who have received copies of the pre-released version.

f. **Copies**. Microsoft will inform Authorized Learning Centers if they may make copies of the beta version (in either print and/or CD version) and distribute such copies to Students and/or Trainers. If Microsoft allows such distribution, you will follow any additional terms that Microsoft provides to you for such copies and distribution.

5. **ADDITIONAL LICENSING REQUIREMENTS AND/OR USE RIGHTS.**

a. Authorized Learning Centers and Trainers:

i. Software.

ii. **Virtual Hard Disks.** The Licensed Content may contain versions of Microsoft XP, Microsoft Windows Vista, Windows Server 2003, Windows Server 2008, and Windows 2000 Advanced Server and/or other Microsoft products which are provided in Virtual Hard Disks.

A. If the Virtual Hard Disks and the labs are launched through the Microsoft Learning Lab Launcher, then these terms apply:

Time-Sensitive Software**.** If the Software is not reset, it will stop running based upon the time indicated on the install of the Virtual Machines (between 30 and 500 days after you install it). You will not receive notice before it stops running. You may not be able to access data used or information saved with the Virtual Machines when it stops running and may be forced to reset these Virtual Machines to their original state. You must remove the Software from the Devices at the end of each Authorized Training Session and reinstall and launch it prior to the beginning of the next Authorized Training Session.

B. If the Virtual Hard Disks require a product key to launch, then these terms apply:

Microsoft will deactivate the operating system associated with each Virtual Hard Disk. Before installing any Virtual Hard Disks on classroom Devices for use during an Authorized Training Session, you will obtain from Microsoft a product key for the operating system software for the Virtual Hard Disks and will activate such Software with Microsoft using such product key.

C. These terms apply to all Virtual Machines and Virtual Hard Disks:

You may only use the Virtual Machines and Virtual Hard Disks if you comply with the terms and conditions of this agreement and the following security requirements:

- You may not install Virtual Machines and Virtual Hard Disks on portable Devices or Devices that are accessible to other networks.
- You must remove Virtual Machines and Virtual Hard Disks from all classroom Devices at the end of each Authorized Training Session, except those held at Microsoft Certified Partners for Learning Solutions locations.
- You must remove the differencing drive portions of the Virtual Hard Disks from all classroom Devices at the end of each Authorized Training Session at Microsoft Certified Partners for Learning Solutions locations.
- You will ensure that the Virtual Machines and Virtual Hard Disks are not copied or downloaded from Devices on which you installed them.
- You will strictly comply with all Microsoft instructions relating to installation, use, activation and deactivation, and security of Virtual Machines and Virtual Hard Disks.
- You may not modify the Virtual Machines and Virtual Hard Disks or any contents thereof.
- You may not reproduce or redistribute the Virtual Machines or Virtual Hard Disks.

ii. **Classroom Setup Guide**. You will assure any Licensed Content installed for use during an Authorized Training Session will be done in accordance with the classroom set-up guide for the Course.

iii. **Media Elements and Templates.** You may allow Trainers and Students to use images, clip art, animations, sounds, music, shapes, video clips and templates provided with the Licensed Content solely in an Authorized Training Session. If Trainers have their own copy of the Licensed Content, they may use Media Elements for their personal training use.

iv. **iv Evaluation Software**. Any Software that is included in the Student Content designated as "Evaluation Software" may be used by Students solely for their personal training outside of the Authorized Training Session.

b. **Trainers Only:**

i. **Use of PowerPoint Slide Deck Templates***.* The Trainer Content may include Microsoft PowerPoint slide decks. Trainers may use, copy and modify the PowerPoint slide decks only for providing an Authorized Training Session. If you elect to exercise the foregoing, you will agree or ensure Trainer agrees: (a) that modification of the slide decks will not constitute creation of obscene or scandalous works, as defined by federal law at the time the work is created; and (b) to comply with all other terms and conditions of this agreement.

ii. **Use of Instructional Components in Trainer Content*.*** For each Authorized Training Session, Trainers may customize and reproduce, in accordance with the MCT Agreement, those portions of the Licensed Content that are logically associated with instruction of the Authorized Training Session. If you elect to exercise the foregoing rights, you agree or ensure the Trainer agrees: (a) that any of these customizations or reproductions will only be used for providing an Authorized Training Session and (b) to comply with all other terms and conditions of this agreement.

iii. **Academic Materials.** If the Licensed Content contains Academic Materials, you may copy and use the Academic Materials. You may not make any modifications to the Academic Materials and you may not print any book (either electronic or print version) in its entirety. If you reproduce any Academic Materials, you agree that:

- The use of the Academic Materials will be only for your personal reference or training use
- You will not republish or post the Academic Materials on any network computer or broadcast in any media;
- You will include the Academic Material's original copyright notice, or a copyright notice to Microsoft's benefit in the format provided below:

  Form of Notice:

  © 2008 Reprinted for personal reference use only with permission by Microsoft Corporation. All rights reserved.

  Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the US and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

6. **INTERNET-BASED SERVICES.** Microsoft may provide Internet-based services with the Licensed Content. It may change or cancel them at any time. You may not use these services in any way that could harm them or impair anyone else's use of them. You may not use the services to try to gain unauthorized access to any service, data, account or network by any means.

7. **SCOPE OF LICENSE.** The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allow you to use it in certain ways. You may not

- install more copies of the Licensed Content on classroom Devices than the number of Students and the Trainer in the Authorized Training Session;
- allow more classroom Devices to access the server than the number of Students enrolled in and the Trainer delivering the Authorized Training Session if the Licensed Content is installed on a network server;
- copy or reproduce the Licensed Content to any server or location for further reproduction or distribution;
- disclose the results of any benchmark tests of the Licensed Content to any third party without Microsoft's prior written approval;
- work around any technical limitations in the Licensed Content;
- reverse engineer, decompile or disassemble the Licensed Content, except and only to the extent that applicable law expressly permits, despite this limitation;
- make more copies of the Licensed Content than specified in this agreement or allowed by applicable law, despite this limitation;
- publish the Licensed Content for others to copy;

- transfer the Licensed Content, in whole or in part, to a third party;
- access or use any Licensed Content for which you (i) are not providing a Course and/or (ii) have not been authorized by Microsoft to access and use;
- rent, lease or lend the Licensed Content; or
- use the Licensed Content for commercial hosting services or general business purposes.
- Rights to access the server software that may be included with the Licensed Content, including the Virtual Hard Disks does not give you any right to implement Microsoft patents or other Microsoft intellectual property in software or devices that may access the server.

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

9. **NOT FOR RESALE SOFTWARE/LICENSED CONTENT.** You may not sell software or Licensed Content marked as "NFR" or "Not for Resale."

10. **ACADEMIC EDITION.** You must be a "Qualified Educational User" to use Licensed Content marked as "Academic Edition" or "AE." If you do not know whether you are a Qualified Educational User, visit www.microsoft.com/education or contact the Microsoft affiliate serving your country.

11. **TERMINATION**. Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of these license terms. In the event your status as an Authorized Learning Center or Trainer a) expires, b) is voluntarily terminated by you, and/or c) is terminated by Microsoft, this agreement shall automatically terminate. Upon any termination of this agreement, you must destroy all copies of the Licensed Content and all of its component parts.

12. **ENTIRE AGREEMENT. This agreement, and the terms for supplements, updates, Internet-based services and support services that you use, are the entire agreement for the Licensed Content and support services.**

13. **APPLICABLE LAW.**

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

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

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

15. **DISCLAIMER OF WARRANTY. The Licensed Content is licensed "as-is." You bear the risk of using it. Microsoft gives no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this agreement cannot change. To the extent permitted under your local laws, Microsoft excludes the implied warranties of merchantability, fitness for a particular purpose and non-infringement.**

16. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO U.S. $5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.**

    This limitation applies to

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

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

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

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

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

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

Cette limitation concerne:

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

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

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

Acknowledgement

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

Peter Lammers – Lead Developer

Peter Lammers joined Aeshen in 2002 as a Product Analyst, and he has been a Lead Product Analyst since 2005, working on Microsoft TechNet Content, Webcasts, White Papers, and Microsoft Learning Courses. Prior to that he has been a computer programmer and network technician with a 14-year background in troubleshooting, training, modifying and supporting a software application; network administration, troubleshooting, and server, desktop, and firewall support.

Sean Masters – Content Developer

Mr. Masters joined Aeshen in 2007. He has worked in SMB technical operations for nearly 10 years including 4 years as manager of information technology at a property management firm and 4 years as a private consultant to various legal and financial firms in the New England area.

Sunni Brock – Content Developer

Sunni has been working with Aeshen as a content developer since 2006. In her 20-year career, she spent 15 years at Microsoft as a Lead Program Manager in the Windows Product Group and as a Technical Support lead replicating customer configuration scenarios. Prior to joining Aeshen, she served as a Technical Account Manager for Sonic Solutions/Roxio acting as liaison to Adobe, Microsoft, Sony and other technology leaders.

Karl Middlebrooks - Subject Matter Expert

Mr. Middlebrooks is a Product Analyst with Aeshen, and joined in 2004. He has over 20 years experience in IT and Operations management, network administration, and database administration.

Contents

About This Course

This section provides you with a brief description of the course, audience, suggested prerequisites, and course objectives.

Course Description

This three-day instructor-led course teaches students how to implement an Analysis Services solution in an organization. The course discusses how to use the Analysis Services development tools to create an Analysis Services database and an OLAP cube, and how to use the Analysis Services management and administrative tools to manage an Analysis Services solution.

Audience

The primary audience for this course is individuals who design and maintain business intelligence solutions for their organization. These individuals work in environments where databases play a key role in their primary job and may perform database administration and maintenance as part of their primary job responsibilities.

The secondary audience for this course is individuals who develop applications that deliver content from Microsoft® SQL Server® Analysis Services to the organization.

Student Prerequisites

This course requires that you meet the following prerequisites:

- Conceptual understanding of OLAP solutions.
- Experience navigating the Microsoft Windows Server® environment.
- Experience with Windows services (starting and stopping).
- Experience creating service accounts and permissions.
- Experience with Microsoft SQL Server, including:
  - SQL Server Agent.
  - SQL Server query language (SELECT, UPDATE, INSERT, and DELETE).
  - SQL Server system tables.
  - SQL Server accounts (users and permissions).

In addition, it is recommended, but not required, that students have completed:

- Course 6231: Maintaining a Microsoft SQL Server 2008 Database.
- Course 6232: Implementing a Microsoft SQL Server 2008 Database.

Course Objectives

After completing this course, students will be able to:

- Describe how SQL Server Analysis Services can be used to implement analytical solutions.
- Create multidimensional analysis solutions with SQL Server Analysis Services.
- Implement dimensions and cubes in an Analysis Services solution.
- Implement measures and measure groups in an Analysis Services solution.
- Query a multidimensional Analysis Services solution.
- Customize an Analysis Services cube.
- Deploy and secure an Analysis Services database.
- Maintain a multidimensional Analysis Services solution.
- Implement a Data Mining solution.

Course Outline

This section provides an outline of the course:

Module 1: Introduction to Microsoft SQL Server Analysis Services

This module introduces common analysis scenarios and describes how Analysis Services provides a powerful platform for multidimensional OLAP solutions and data mining solutions. The module then describes the main considerations for installing Analysis Services.

Module 2: Creating Multidimensional Analysis Solutions

This module introduces the development tools you can use to create an Analysis Services multidimensional analysis solution, and describes how to create data sources, data source views, and cubes.

Module 3: Working with Cubes and Dimensions

This module describes how to edit dimensions and to configure dimensions, attributes, and hierarchies.

Module 4: Working with Measures and Measure Groups

This module explains how to edit and configure measures and measure groups.

Module 5: Querying Multidimensional Analysis Solutions

This module introduces multidimensional expressions (MDX) and describes how to implement calculated members and named sets in an Analysis Services cube.

Module 6: Customizing Cube Functionality

This module explains how to customize a cube by implementing key performance indicators (KPIs), actions, perspectives, and translations.

Module 7: Deploying and Securing an Analysis Services Database

This module describes how to deploy an Analysis Services database to a production server, and how to implement security in an Analysis Services multidimensional solution.

Module 8: Maintaining a Multidimensional Solution

This module discusses the maintenance tasks associated with an Analysis Services solution, and describes how administrators can use the Analysis Services management tools to perform them.

Module 9: Introduction to Data Mining

This module introduces data mining, and describes how to implement data mining structures and models. It then explains how to validate data model accuracy.

Course Materials

The following materials are included with your kit:

- *Course Handbook*. A succinct classroom learning guide that provides all the critical technical information in a crisp, tightly-focused format, which is just right for an effective in-class learning experience.
  - Lessons: Guide you through the learning objectives and provide the key points that are critical to the success of the in-class learning experience.
  - Labs: Provide a real-world, hands-on platform for you to apply the knowledge and skills learned in the module.
  - Module Reviews and Takeaways: Provide improved on-the-job reference material to boost knowledge and skills retention.
  - Lab Answer Keys: Provide step-by-step lab solution guidance at your finger tips when it's needed.
- *Course Companion CD*. Searchable, easy-to-navigate digital content with integrated premium on-line resources designed to supplement the Course Handbook.
  - Lessons: Include detailed information for each topic, expanding on the content in the Course Handbook.
  - Labs: Include complete lab exercise information and answer keys in digital form to use during lab time.
  - Resources: Include well-categorized additional resources that give you immediate access to the most up-to-date premium content on TechNet, MSDN®, and Microsoft Press®.
  - Student Course Files: Include the Allfiles.exe, a self-extracting executable file that contains all the files required for the labs and demonstrations.

Note: To access the full course content, insert the Course Companion CD into the CD-ROM drive, and then in the root directory of the CD, double-click StartCD.exe.

- *Course evaluation.* At the end of the course, you will have the opportunity to complete an online evaluation to provide feedback on the course, training facility, and instructor.

To provide additional comments or feedback on the course, send e-mail to support@mscourseware.com. To inquire about the Microsoft Certification Program, send e-mail to mcphelp@microsoft.com.

Virtual Machine Environment

This section provides the information for setting up the classroom environment to support the business scenario of the course.

Virtual Machine Configuration

In this course, you will use Microsoft Virtual Server 2005 R2 with SP1 to perform the labs.

> **Important:** At the end of each lab, you must close the virtual machine and must not save any changes. To close a virtual machine without saving the changes, perform the following steps:
> 1. On the virtual machine, on the **Action** menu, click **Close**.
> 2. In the **Close** dialog box, in the **What do you want the virtual machine to do?** list, click **Turn off and delete changes**, and then click **OK**.

The following table shows the role of each virtual machine used in this course:

Virtual machine	Role
6234A-NY-SQL-01	Windows Server 2008 with SQL Server 2008
6234A-NY-SQL-02	Server used to install SQL Server 2008

Software Configuration

The following software is installed on each VM:

- Windows Server 2008 Enterprise Edition
- SQL Server 2008

Course Files

There are files associated with the labs in this course. The lab files are located in the folder E:\Labfiles on the student computers.

Classroom Setup

Each classroom computer will have the same virtual machine configured in the same way.

Course Hardware Level

To ensure a satisfactory student experience, Microsoft Learning requires a minimum equipment configuration for trainer and student computers in all Microsoft Certified Partner for Learning Solutions (CPLS) classrooms in which Official Microsoft Learning Product courseware are taught.

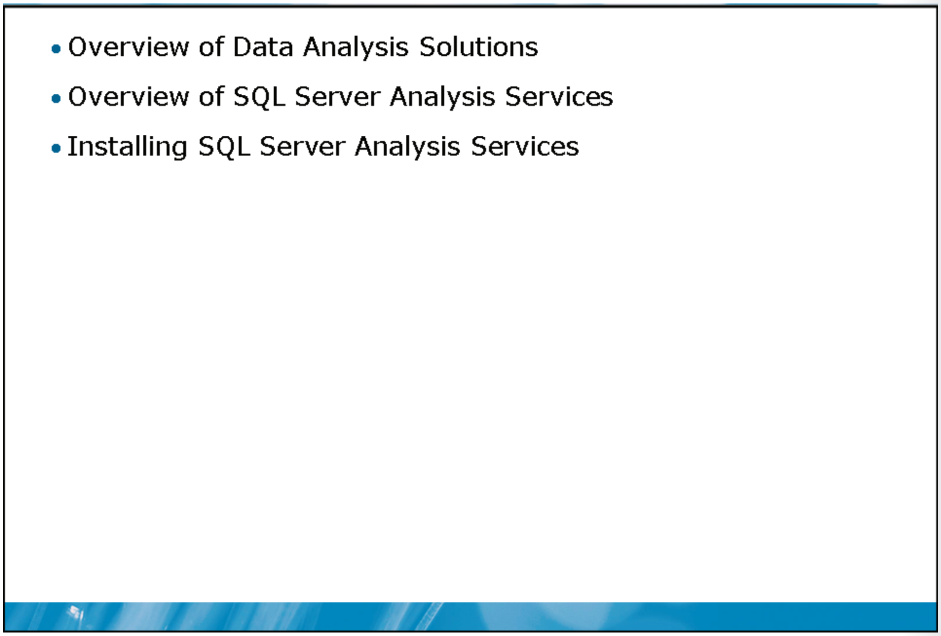
This course requires that you have a computer that meets or exceeds hardware level 5.5, which specifies a 2.4–gigahertz (GHz) (minimum) Pentium 4 or equivalent CPU, at least 2 gigabytes (GB) of RAM, 16 megabytes (MB) of video RAM, and two 7200 RPM 40-GB hard disks.

Module 1

Introduction to Microsoft SQL Server Analysis Services

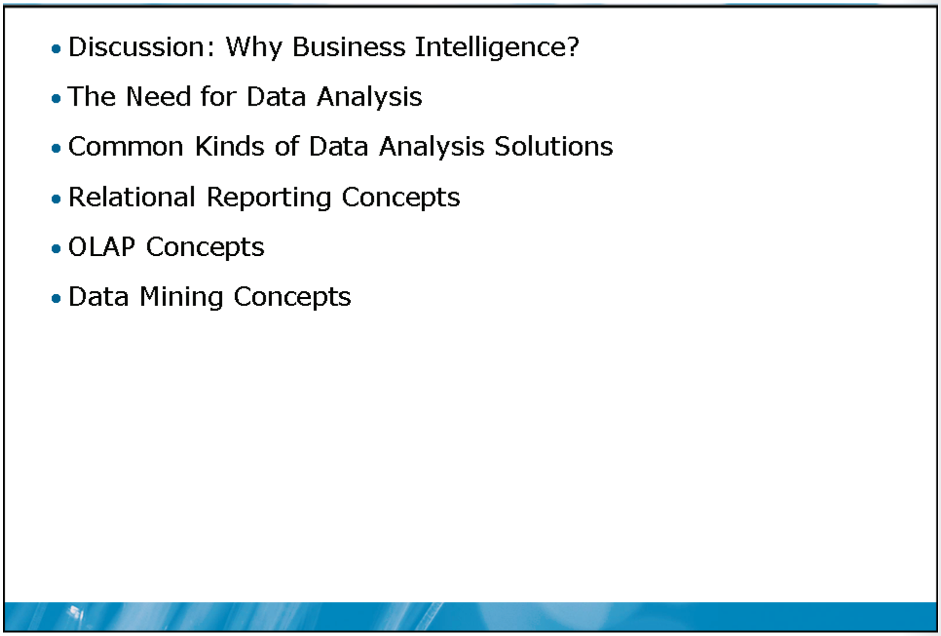
Contents:

Module Overview



Databases throughout an organization contain large amounts of data that can be useful to a variety of people and provide important insight for making business decisions. In order to bring data from databases to people in a meaningful way, you need to have ways to analyze and present it. Microsoft® SQL Server® Analysis Services helps your organization make sense of the data stored in your databases.

Lesson 1
Overview of Data Analysis Solutions



Business intelligence (BI) allows you to bring together information from different sources to perform analysis and reporting. The goal of business intelligence is to allow you to make better business decisions.

Discussion: Why Business Intelligence?

- Continuous Business Improvement
- Better, More Informed, Faster Business Decisions

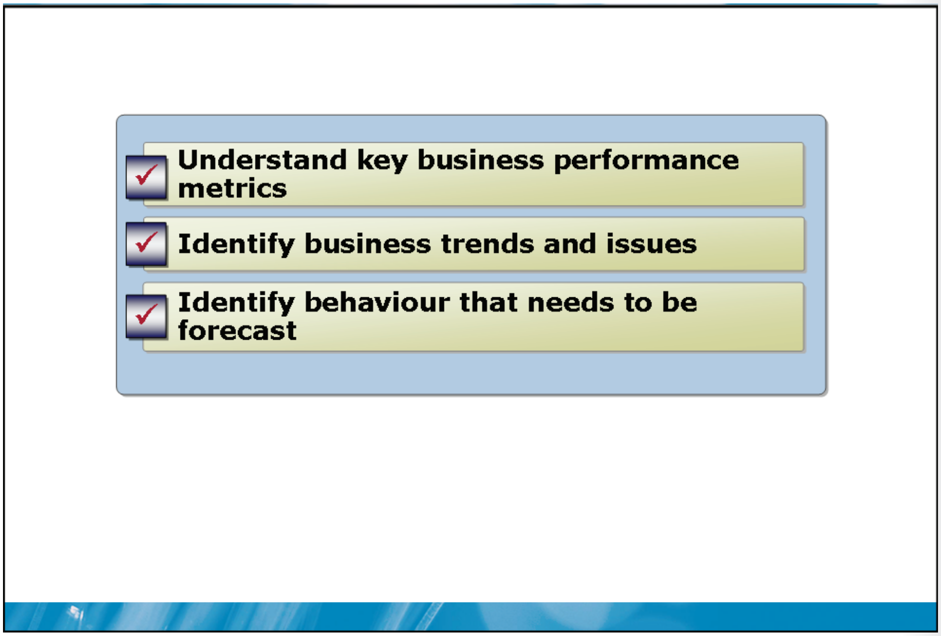
Key Points

The goal of business intelligence (BI) is allow continuous business improvement and to enable better, more informed, and faster decisions - to get the right information at the right time and in the right format.

Many times a combination of up-to-date and historical information is necessary. For example, while it may be useful for the store manager to know the number of sales of a particular item so far this morning, it is more useful if he can put that information in the context of the average number sales of that same item on the same day of the week over the last year.

Question: What would business intelligence mean in your organization?

The Need for Data Analysis



Key Points

Organizations generate a tremendous amount of data. Data is often stored in a number of databases, and can provide useful information to people throughout the organization. In order to make best use of database data, it must be analyzed and presented in meaningful ways.

Organizations typically need data analysis to:

- Understand key business performance metrics.
- Identify business trends and issues.
- Identify behavior that needs to be forecast.

Question: What kinds of data do you think would benefit from data analysis?

Common Kinds of Data Analysis Solutions

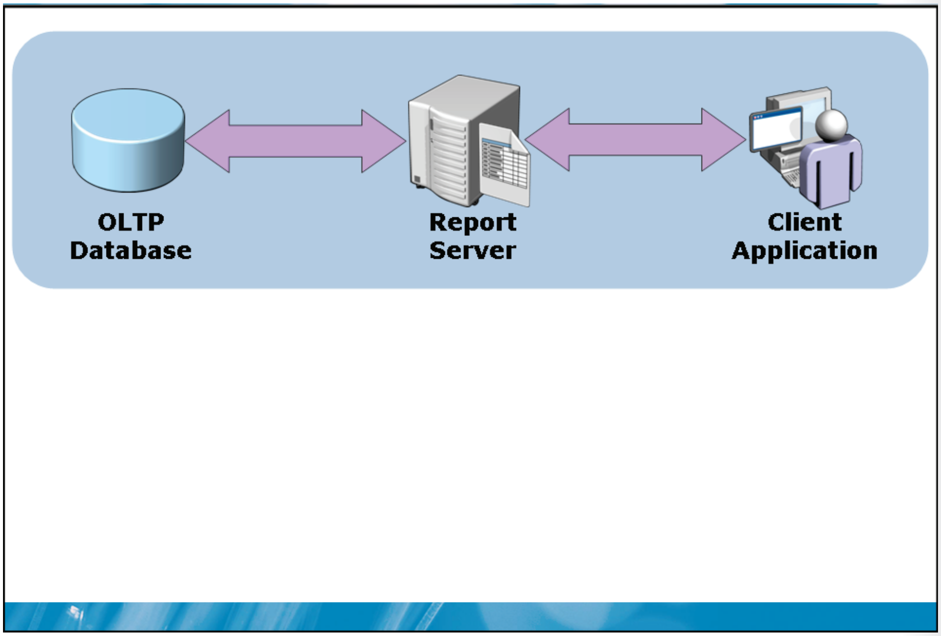
Key Points

There several common types of data analysis solutions.

- Relational reporting involves reporting directly from data in an Online Transaction Processing (OLTP) database, usually in pre-defined (canned) reports or ad-hoc reports that administrators or users can create themselves.
- Online Analytical Processing (OLAP) performs analysis of data across several dimensions.
- Data mining is the process of sorting through large volumes of data to find patterns and correlations that might otherwise go unnoticed.

Question: What types of data analysis solutions are currently in use in your organization?

Relational Reporting Concepts



Key Points

In relational reporting, a reporting server such as SQL Server Reporting Services or other reporting tool is used to retrieve information from an OLTP database and present it to a client application, such as a report viewer, an office application, like Microsoft Office Excel®, or through a web interface.

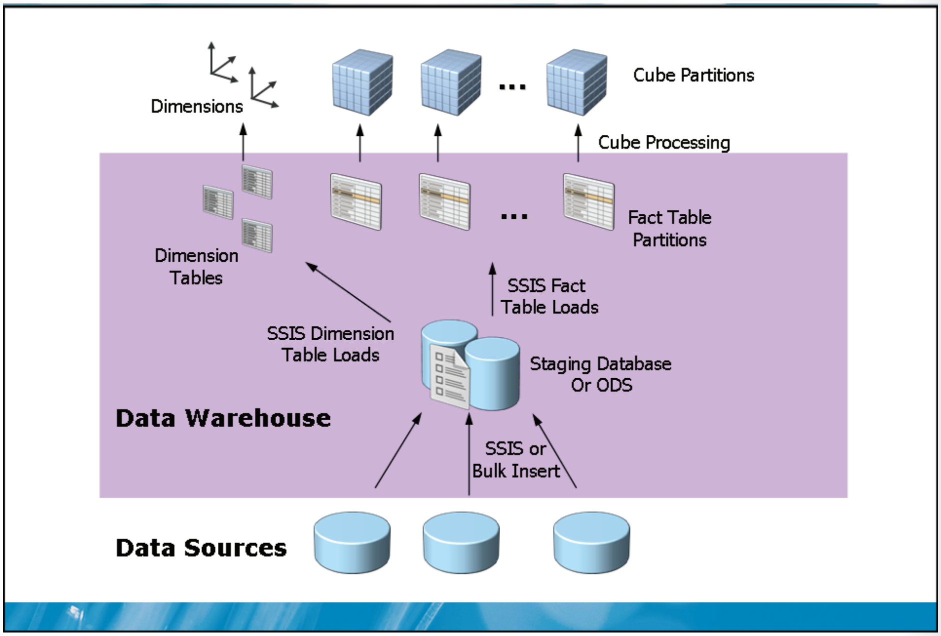
Relational reporting typically uses pre-defined reports that are created by an administrator or ad-hoc reports that can be created by administrators or users. The primary benefit of relational reporting is that it is good for obtaining the most up-to-date data, because the data is coming directly from the OLTP database.

There are some drawbacks to relational reporting:

- OLTP databases are designed for ease of entering transactions, not ease of reporting. Building a report on a complex OLTP schema can be difficult.
- Report generation can have a negative impact on the performance of an OLTP database.
- OLTP databases frequently do not hold historical data; to keep the OLTP databases manageable, historical transactions are often moved out of the database.

Question: When would you want to use relational reporting?

OLAP Concepts



Key Points

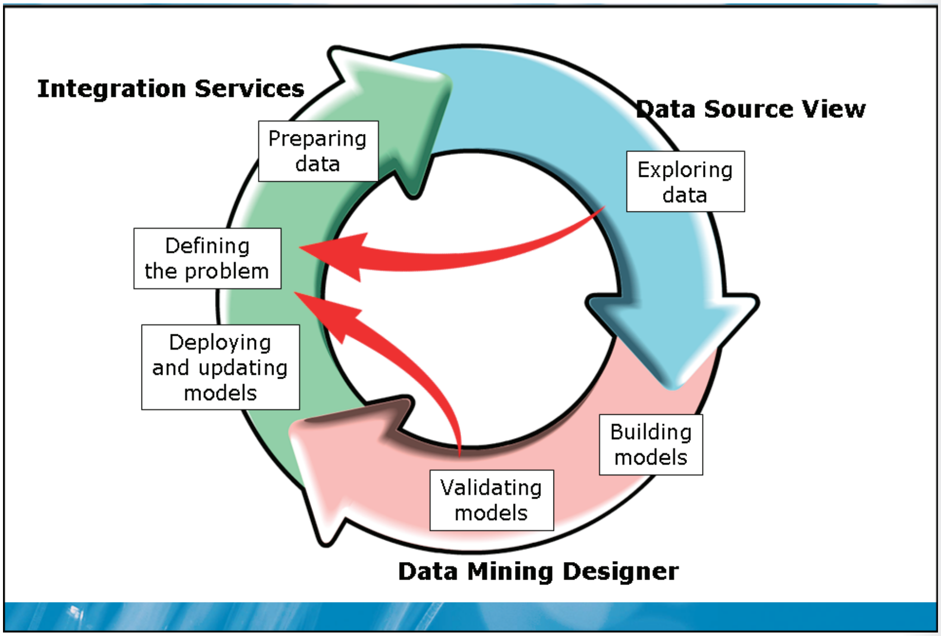
With OLAP, data can be loaded from a number of heterogeneous data sources, often OLTP databases, using SQL Server Integration Services or another method. The database that results from the integration of this data is called a data warehouse. A data mart, holds a subset of data from a data warehouse, usually on a specific business activity.

- Once data is in the data warehouse, facts and dimensions can be created. Facts are predominantly numeric measurements, such as price or quantity, that represent the key business metrics you want to aggregate and analyze. Dimensions are the contexts for the facts, and define the aspects of a business by which the facts are aggregated.

- Cubes store summarized fact and dimension data in multidimensional structures. Once the cube is created, it can be sliced and diced, which involve rearranging data so that it can be viewed through different perspectives.
- Users typically browse cubes through an intuitive interface called a pivot table. Pivot tables allow users to view large amounts of data in summary reports.

Question: Why would you want to use OLAP over relational reporting?

Data Mining Concepts

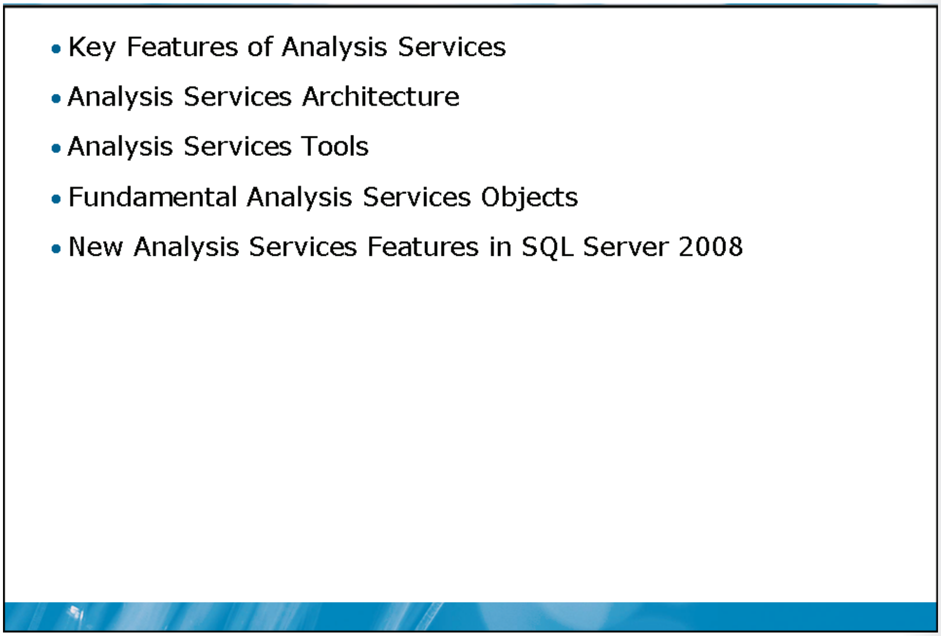


Key Points

Data mining derives patterns and trends that exist in large databases.

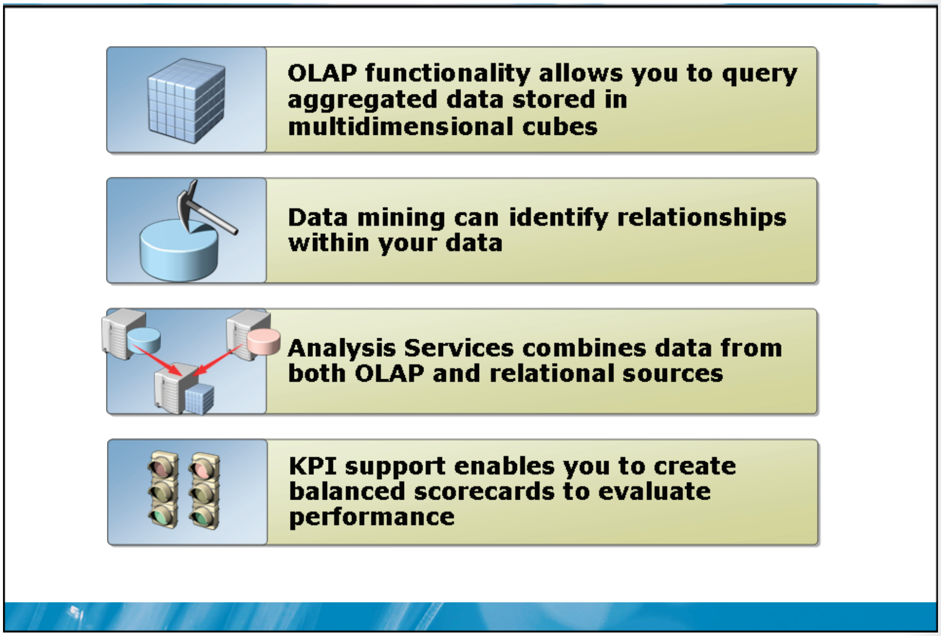
- Data mining uses a data source view to examine data that resides in a data warehouse.
- A data mining structure defines how data is used in a mining model.
- A data mining model uses an algorithm to process the data in the mining structure and find patterns and correlation among the data.
- Several types of data mining models are available in Analysis Services and a mining structure can contain several different mining models.

Lesson 2
Overview of SQL Server Analysis Services



SQL Server Analysis Services gives you the ability to perform OLAP and data mining tasks on your organization's data. Several tools are available to help you develop BI solutions. Developing BI solutions requires familiarity with several types of Analysis Services Objects.

Key Features of Analysis Services

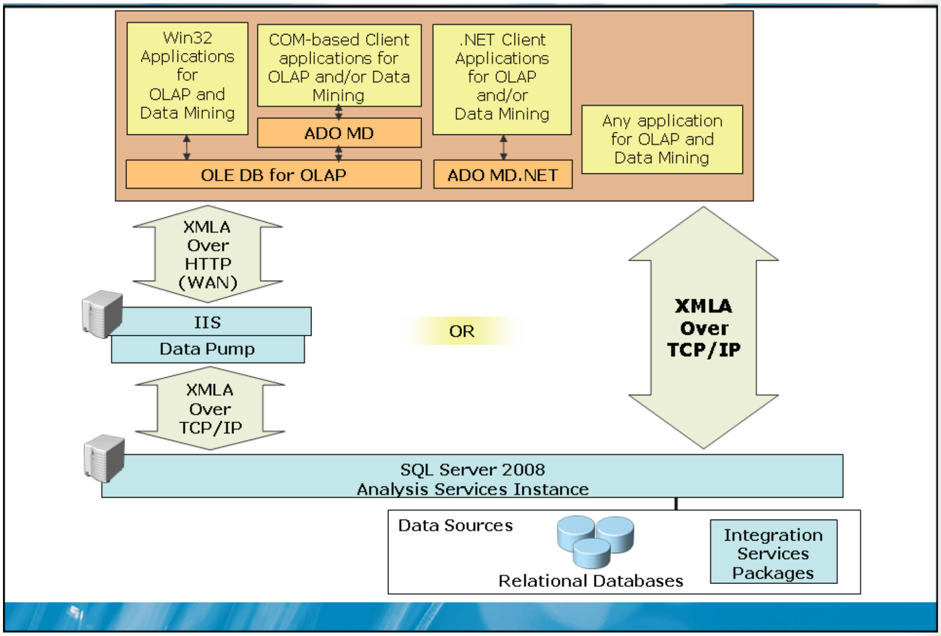


Key Points

SQL Server Analysis Services provides a single point of access for an organization's business intelligence (BI) data. By combining OLAP and data mining technologies and key performance indicator (KPI) information in one unified system, it simplifies development of BI solutions and makes results more accessible to users.

- OLAP functionality allows you to query aggregated data stored in multidimensional cubes.
- Data mining can identify relationships within your data and assess probabilities of future results based on past actions.
- Analysis Services combines data from both OLAP and relational sources.
- KPI support in Analysis Services enables you to create balanced scorecards to evaluate performance against business objectives.

Analysis Services Architecture



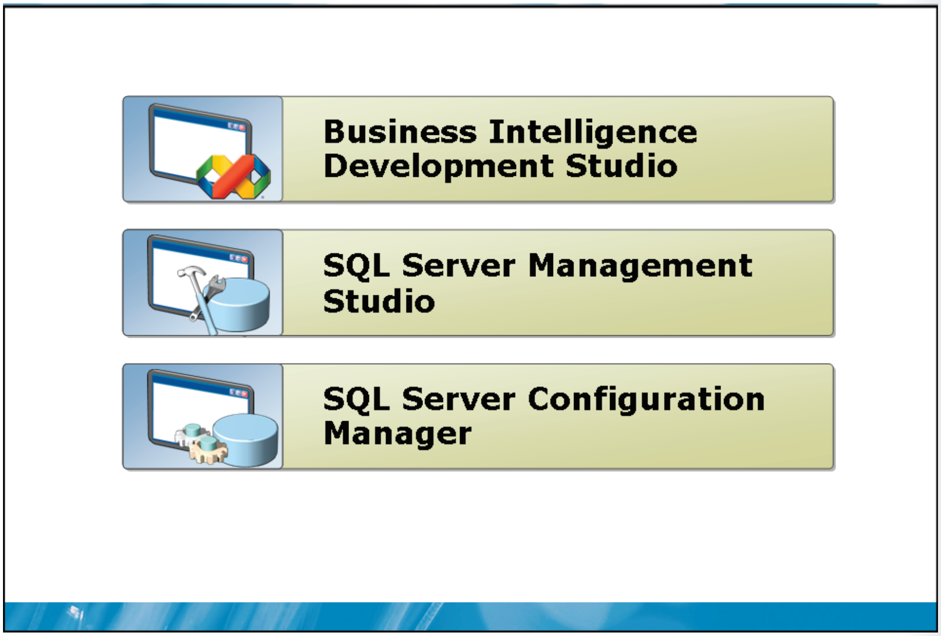
Key Points

Analysis Services architecture consists of two main components: server architecture and client architecture.

- Server architecture includes all of the storage and processing tasks handled by the server.
  - The Analysis Services server architecture consists of an application that runs as a service.
  - Analysis Services databases are stored on the server and accessed through the Analysis Services service. The Analysis Services service consists of components that handle aggregations, transactions, calculations, metadata management, security and XML for Analysis (XMLA), as well as numerous other functions.

- Client architecture includes the connection and caching tasks handled by the client.
    - The Analysis Services client architecture allows clients to connect to Analysis Services cubes using ADO MD, ADO MD.Net, XML/A, or OLEDB for OLAP.
    - Users access the cube from an application such as Microsoft Office Excel.

Analysis Services Tools

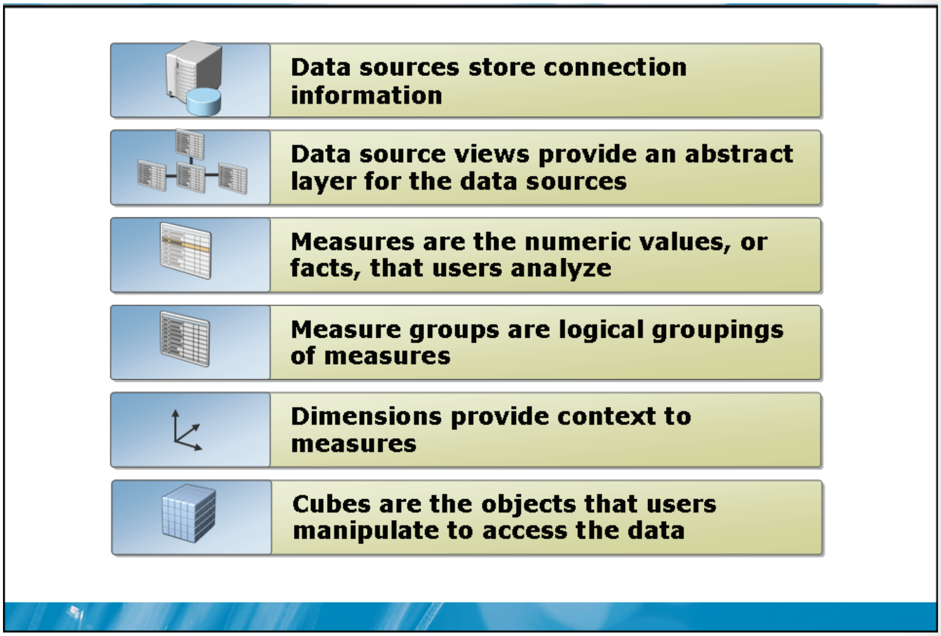


Key Points

There are three key tools in the development and management of Analysis Services solutions: SQL Server Configuration Manager, SQL Server Management Studio, and Business Intelligence Development Studio. You can accomplish most tasks by using more than one method, and you can accomplish many tasks by using multiple tools.

- Business Intelligence Development Studio is a version of Microsoft Visual Studio® 2008. It is the principal tool for the creation of BI solutions.
- SQL Server Management Studio is the main relational and Analysis Services database management tool. You can use this tool to develop Analysis Services solutions, but it is more commonly used to manage Analysis Services databases in a production environment.
- SQL Server Configuration Manager is a tool that you can use to manage SQL Server services and server-and-client network configurations.

Fundamental Analysis Services Objects



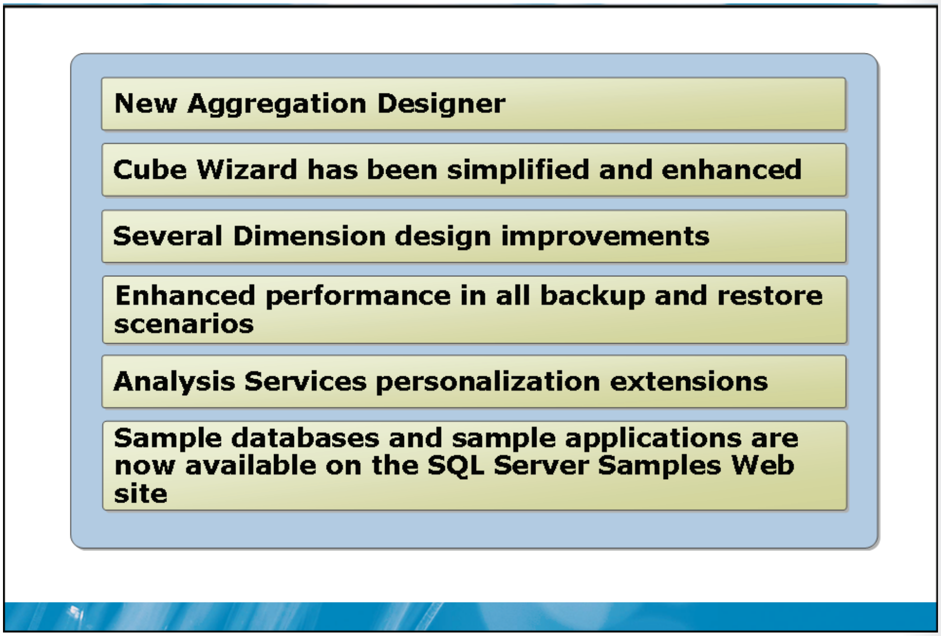
Key Points

OLAP databases are defined by the cubes, dimensions, and measures that they contain. You also need to define connection information and organize the dimensions and measures. The objects combine to form an Analysis Services database.

- Data sources store the connection information in the Analysis Services project.
- Data source views provide an abstract layer for the data sources.
- Measures are the numeric values, or facts, that users analyze.

- Measure groups are logical groupings of measures.
- Dimensions provide context to the measures. Dimensions are organized into hierarchical structures. For example, a time dimension might contain day, month, and year levels.
- Cubes combine dimensions and measures to form a multidimensional structure containing the aggregation of each measure at the intersection of dimension members. Cubes are the objects that users manipulate to access the data.

New Analysis Services Features in SQL Server 2008



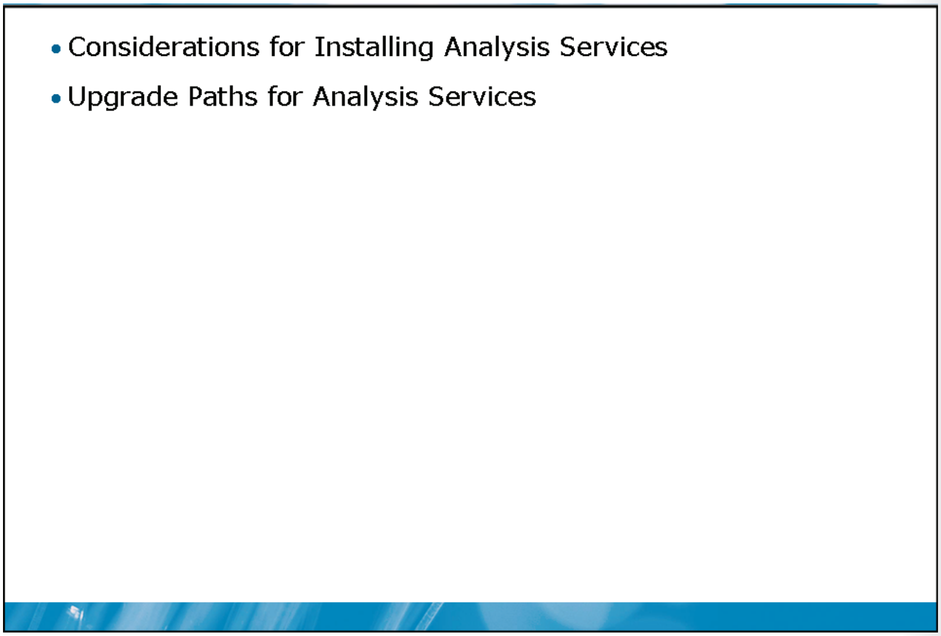
Key Points

There are many new Analysis Services features in SQL Server 2008.

- A new Aggregation designer makes it easier to browse and modify aggregation designs. Aggregation designs are now shown grouped by measure group.
- The updated Aggregation Design and Usage-Based Optimization wizards let you modify the storage settings for aggregations in one or more partitions at a time and more easily set aggregation usage settings.
- New AMO Warning messages alert users when they depart from design best practices or logical errors.
- The dimension editor has a new Attribute Relationship designer that makes it easier to browse and modify attribute relationships.
- The latest version of the Dimension Wizard auto-detects parent-child hierarchies, provides safer default error configuration, and supports specification of member properties.

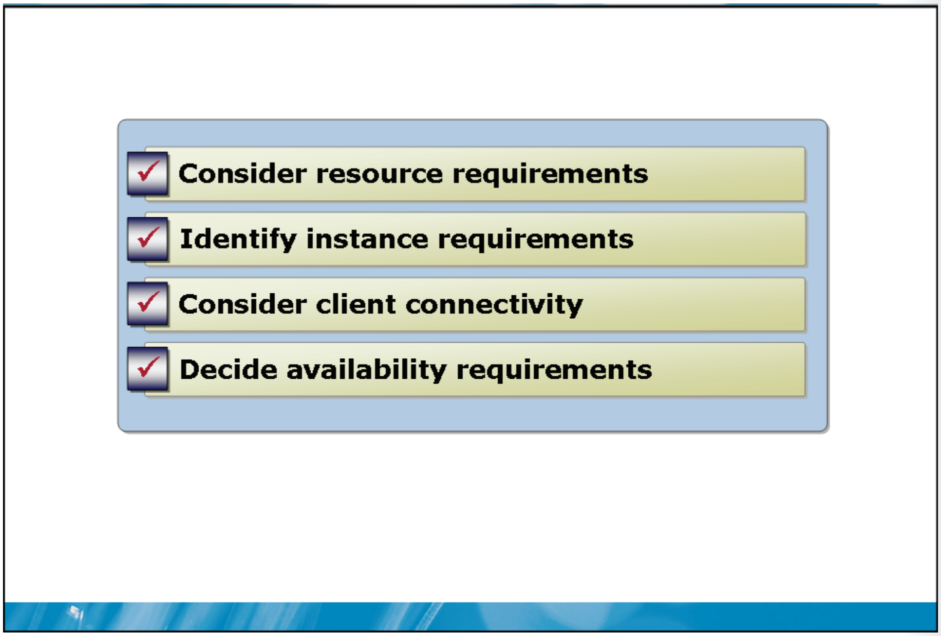
- A new Key Columns dialog box makes editing key columns easier.
- The Dimension Structure tab now works with the new Attribute Relationship designer and is simpler to use, which makes modifying attributes and hierarchies easier.
- The new storage structure provides a more robust repository for the archived database.
- The new backup and restore functionality achieves increased performance.
- New Analysis Services personalization extensions enable developers to create new Analysis Services objects and functionality and to provide these objects and functionality dynamically in the context of the user session.
- Books Online sample databases and sample applications are now available on the SQL Server Samples web site.

Lesson 3
Installing SQL Server Analysis Services



Before installing or upgrading Analysis Services, you should take into consideration your organization's needs, and your resource requirements.

Considerations for Installing Analysis Services



Key Points

Before performing an installation of Analysis Services there are several things to take into consideration:

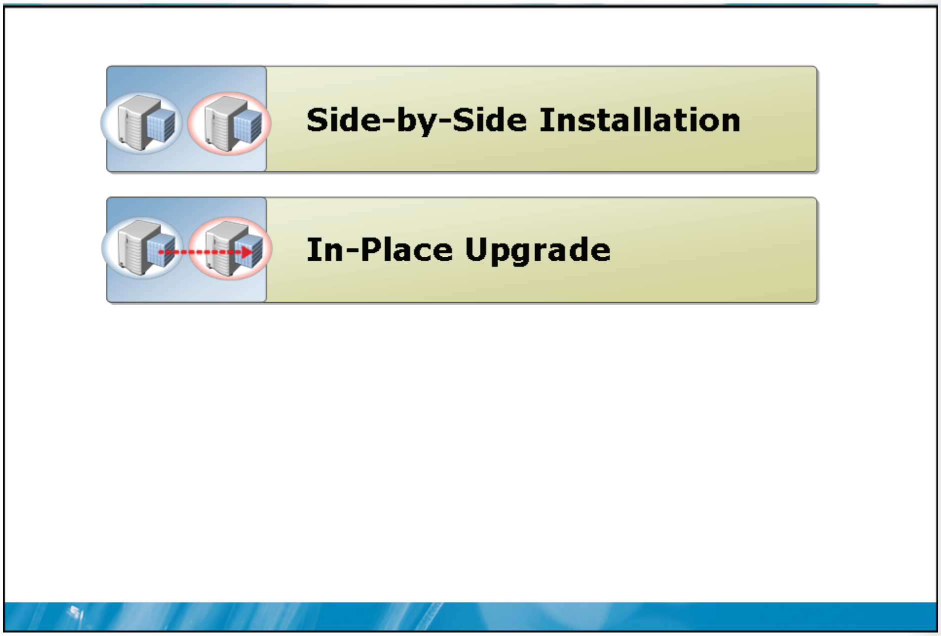
- Consider resource requirements for your installation. The number, size, and complexity of cubes you plan to have, the number of instances and the number of databases in each instance, as well as the number of concurrent users will all affect your hardware needs.

- Consider instance requirements. You can run a single default instance and multiple named instances of Analysis Services on a single server. Each instance is completely independent from the others.

- Consider how your clients will connect to Analysis Services. The default port for an Analysis Services connection is 2383.
- Decide on the availability requirements. There are several options available for high availability, including server clustering and Network Load Balancing.

Question: When would you choose to run multiple Analysis Server instances on a single server?

Upgrade Paths for Analysis Services



Key Points

If you are upgrading a SQL Server 2000 or 2005 Analysis Services instance to SQL Server 2008 Analysis Services, there are two options:

- Side-by-side installation - A side-by-side installation involves installing a new Analysis Services onto a server, migrating the data from one server to another and then reconfiguring client applications to point to the new server.
- In-place upgrade - In an in-place upgrade, you install SQL Server onto your existing server and allow an automatic upgrade of the existing instance and databases.

Lab: Installing and Verifying SQL Server Analysis Services

- Exercise 1: Installing SQL Server Analysis Services
- Exercise 2: Verifying Installation

Logon information

Virtual machine	NY-SQL-02
User name	Student
Password	Pa$$w0rd

Estimated time: 40 minutes

Exercise 1
Installing SQL Server Analysis Services

Scenario

You are a database administrator in the IT department of the Adventure Works bicycle manufacturing company. The manager of the IT department has identified the need to provide business intelligence information throughout the organization.

You must install SQL Server Analysis Services.

The main tasks for this exercise are as follows:

1. Start the NY-SQL-02 virtual machine and log on as Administrator.
2. Install an instance of SQL Server Analysis Services.

▶ Task 1: Start the NY-SQL-02 virtual machine and log on as Administrator

- Start 6234A-NY-SQL-02 and logon as **Student** using the password **Pa$$w0rd.**

▶ Task 2: Install an instance of SQL Server Analysis Services

- Start the SQL Server Installation Wizard. Choose an installation with the following components:
  - Analysis Services
  - Client Tools
- Configure the Analysis Services component as follows:
  - Use a default instance.
  - For the service account use the "**sqlserver**" user with the password **Pa$$w0rd**.
  - Give the Administrator account permission to administer Analysis Services.

Results: After this exercise, SQL Server Analysis Services is installed.

Exercise 2
Verifying Installation

Scenario

Now that you have finished the installation of Analysis Services, you need to verify that the installation was successful and that you can connect to the Analysis Services instance.

The main tasks for this exercise are as follows:

1. View the installation log file.
2. Connect to the Analysis Services instance.

▶ Task 1: View the installation log file

- Click the link to the log file.
- Open the log file in Notepad.

▶ Task 2: Connect to the Analysis Services instance

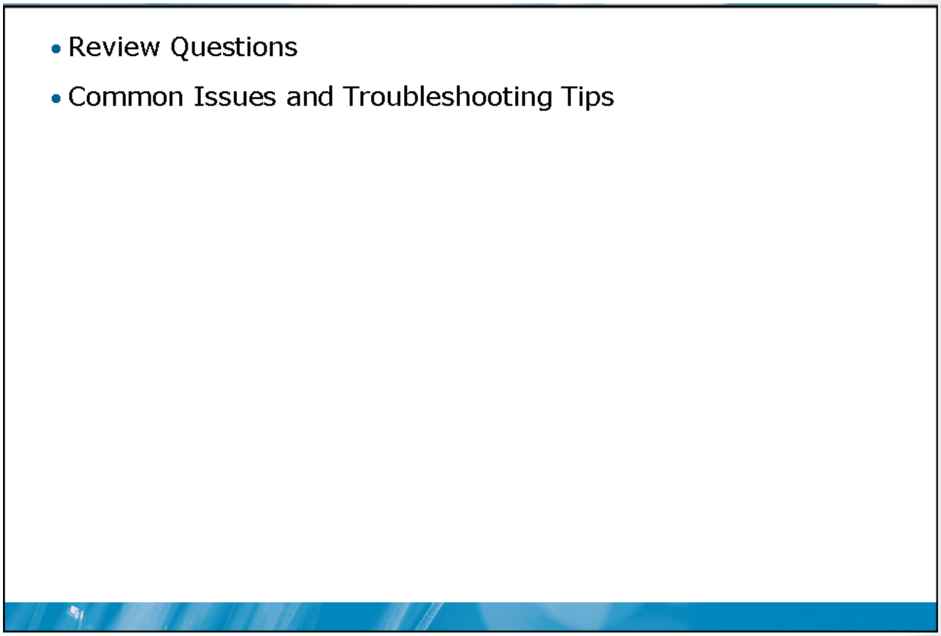
- Start SQL Server Management Studio.
- Connect to the **Analysis Services** instance and expand the **Databases** folder.

> **Results**: After this exercise, you have verified that the Analysis Services installation was successful.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-02 virtual machine and discard any changes.

Module Review and Takeaways



Review Questions

1. What are some common scenarios where organizations would employ Analysis Services?
2. What is the difference between OLTP and OLAP?
3. What are the available upgrade methods for Analysis Services?

Common Issues and Troubleshooting Tips

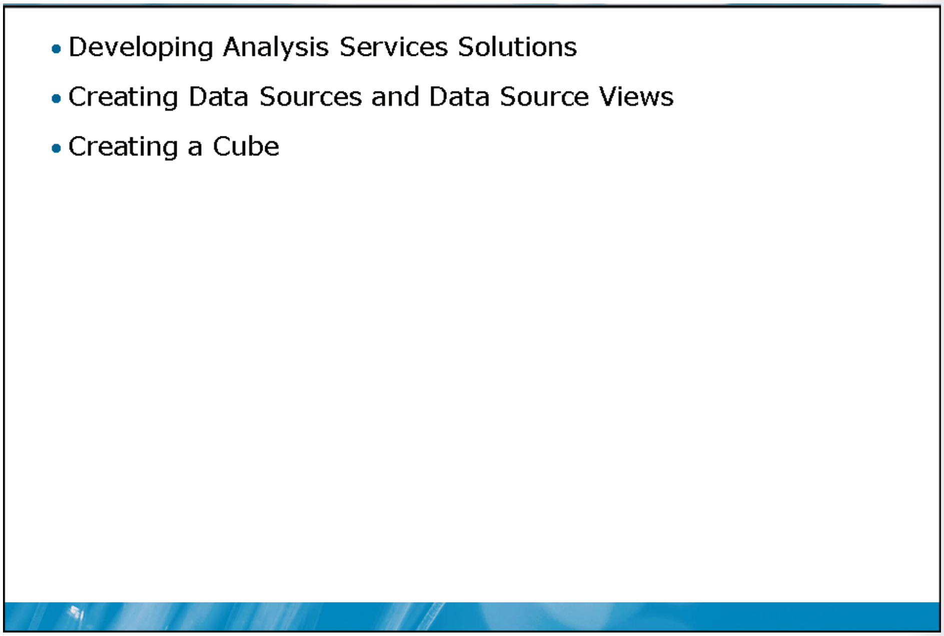
For troubleshooting installation issues, make sure to review the log files that are created in the SQL Server installation directory.

Module 2

Creating Multidimensional Analysis Solutions

Contents:

Module Overview



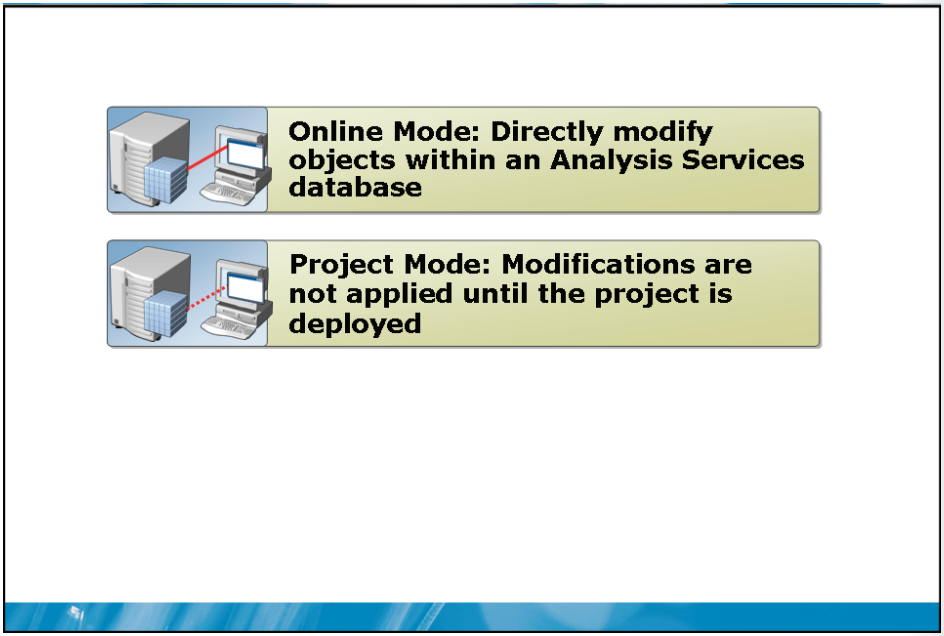
With Analysis Services developers can use the Microsoft® SQL Server® Business Intelligence Development Studio (BIDS) to develop multidimensional business intelligence solutions. Developers create data sources and data source views that allow Analysis Services to access enterprise data. Measures and dimensions are then added to create a cube that gives users a variety of perspectives on the data.

Lesson 1
Developing Analysis Services Solutions

- Options for Analysis Services Development
- Introduction to Business Intelligence Development Studio
- Source Control for Developing Analysis Services Projects

Analysis Services solutions can be developed in Online mode by attaching directly to a database, or in Project mode, which allows a solution to be designed and later deployed. Business Intelligence Development Studio provides a Microsoft Visual Studio®-based development environment for Analysis Services solutions, and it integrates with source control through Team Foundation Server to allow multiple developers to work on a project.

Options for Analysis Services Development



Key Points

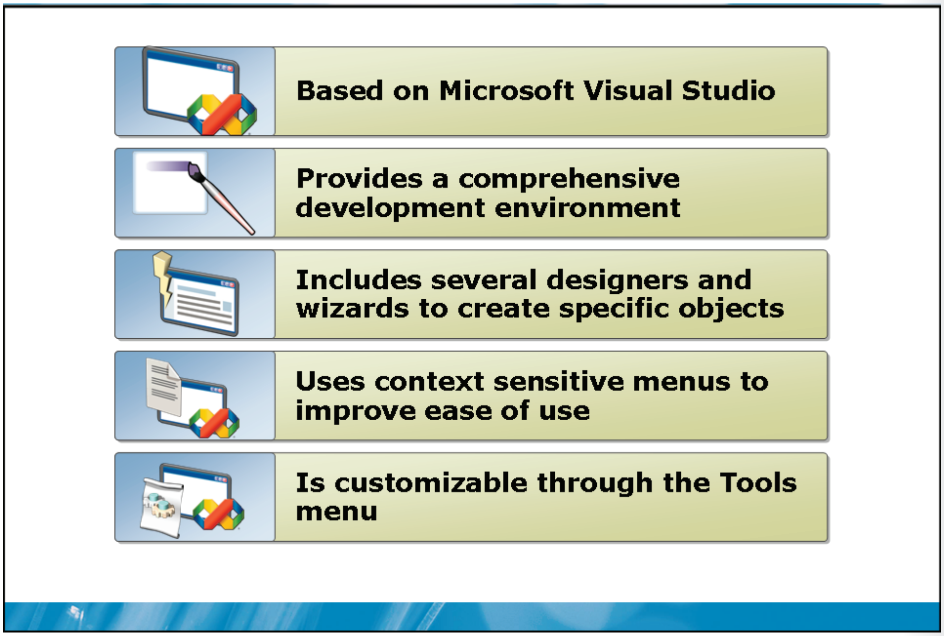
Although you can use Microsoft SQL Server Management Studio for some development work, Business Intelligence Development Studio is more appropriate. Management Studio is appropriate for managing the solution after you have deployed it.

- You can use Business Intelligence Development Studio in Online Mode. Online Mode connects directly to an existing SQL Server Analysis Services instance. It does not create an Analysis Services project, and any changes to the database occur immediately.

- You can also use Business Intelligence Development Studio in Project Mode. Project Mode creates an Analysis Services project that stores the objects as XML files independent of a particular Analysis Services database. You can later deploy the objects to a local or remote Analysis Services database.

Question: When would you choose to develop in Online mode instead of Project mode?

Introduction to Business Intelligence Development Studio

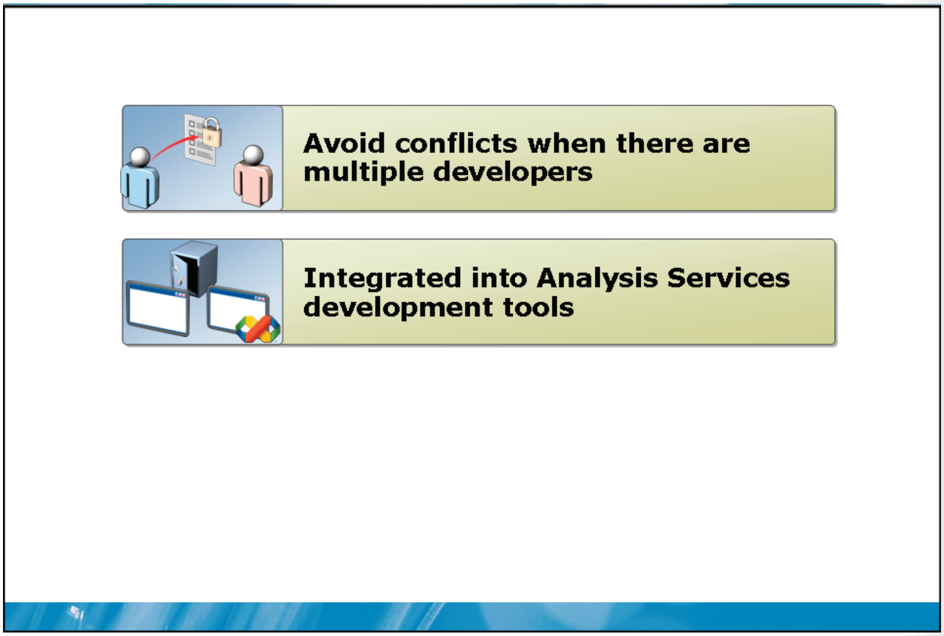


Key Points

Business Intelligence Development Studio is a new project development and management tool for business intelligence solution developers. BI Development Studio is built on Visual Studio 2008.

- BI Development Studio lets you view and edit object definitions, such as cubes, mining models, packages, and reports, either through a graphical user interface, called a designer, or directly by editing the XML-based code that defines the object.
- BI Development Studio also provides several wizards to automate creation of Analysis Services project components, and the Visual Studio-based environment provides context-sensitive menus and several customizable settings.

Source Control for Developing Analysis Services Projects

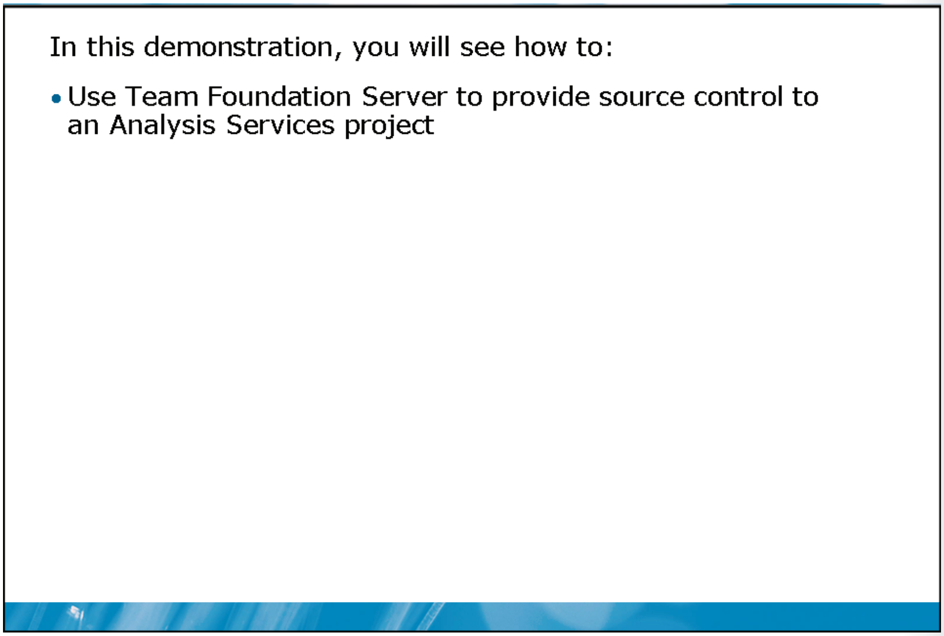


Key Points

Source control allows multiple developers to work on a single project while avoiding conflicts in file versions. Source control providers support both shared and exclusive file access.

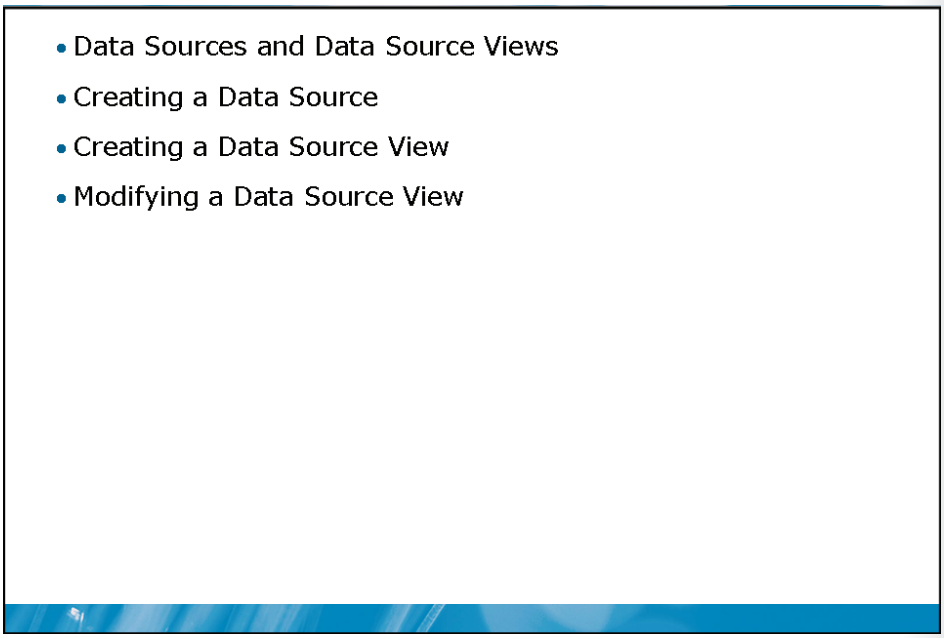
- If access to project files is exclusive, the source control provider allows only one user at a time to check files out and modify them.
- If access is shared, more than one user can check out the script file, and the source control provider provides a mechanism for merging the versions as they are checked in.
- Source control integration with Team Foundation Server is integrated in to BI Development Studio. You can also use Visual Source Safe for source control.

Demonstration: Using Source Control for Analysis Services Projects



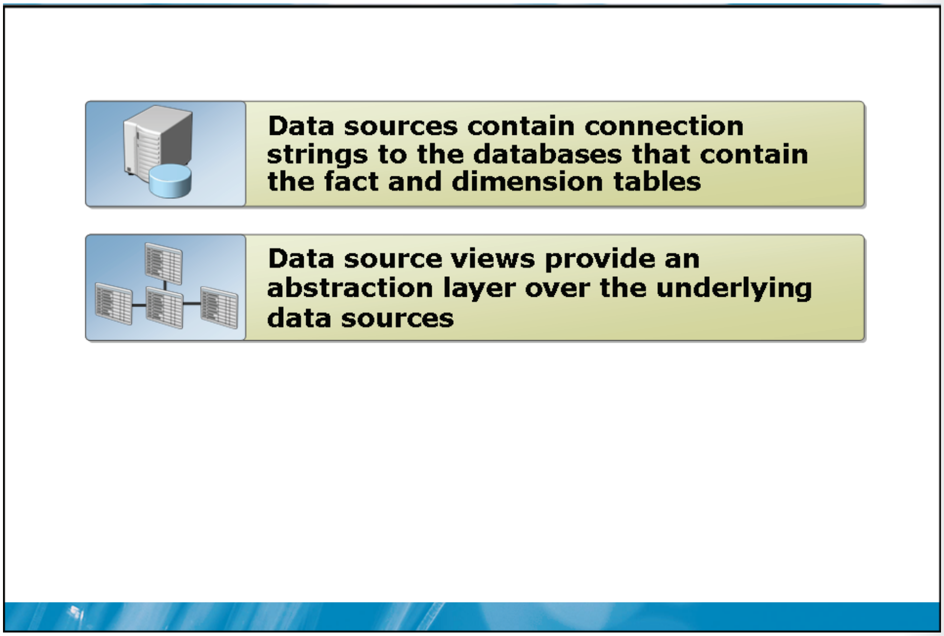
Question: How do multiple developers avoid conflicts while working on the same project?

Lesson 2
Creating Data Sources and Data Source Views



OLAP and data mining projects in Analysis Services are designed based on a logical data model of related tables, views, and queries from one or more data sources. This logical data model is called a data source view. Data source views let you define a subset of the data that populates a large data warehouse.

Data Sources and Data Source Views



Key Points

Data sources and data source views provide the connection information to the underlying databases.

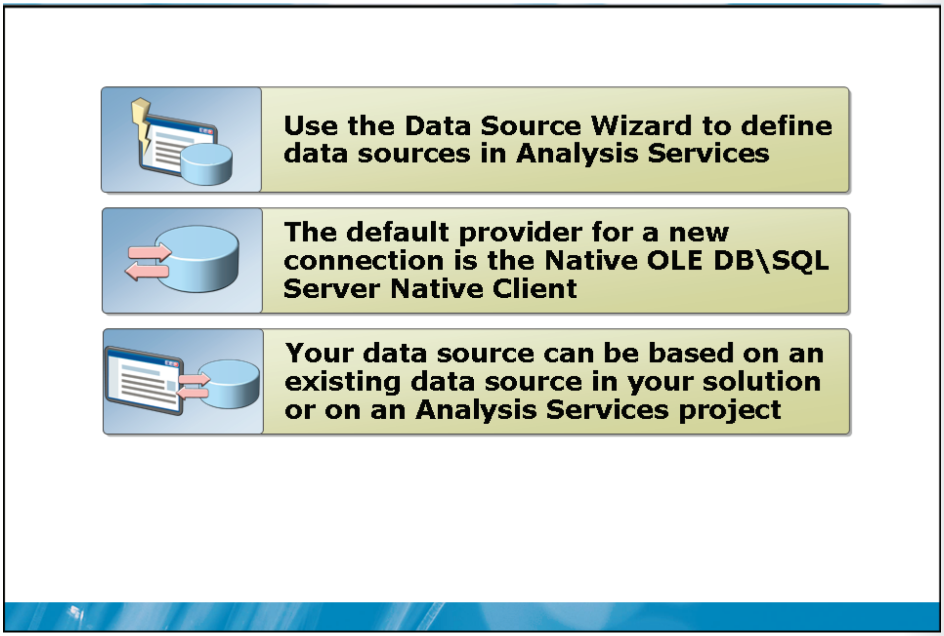
At a minimum, a data source includes an identifier, a name, and a connection string. The connection string used to access the source data specifies the following information:

- The provider name.
- The information needed to connect to the data source using the specified provider. The property settings for particular data source objects vary according to the provider.
- Other properties supported and /or required by the provider.

A data source view contains the following items:

- A name and a description.
- A definition of any subset of the schema retrieved from one or more data sources, up to and including the whole schema, including the following:
  - Table names
  - Column names
  - Data types
  - Nullability
  - Column lengths
  - Primary keys
  - Primary key - foreign key relationships
- Annotations to the schema from the underlying data sources, including the following:
  - Friendly names for tables, views, and columns.
  - Named queries that return columns from one or more data sources (that show as tables in the schema).
  - Named calculations that return columns from a data source (that show as columns in tables or views).
  - Logical primary keys (needed if a primary key is not defining in the underlying table or is not included in the view or named query).
  - Logical primary key - foreign key relationships between tables, views, and named queries.

Creating a Data Source



Key Points

You use the Data Source Wizard in Business Intelligence Development Studio to define one or more data sources for a Microsoft SQL Server Analysis Services project.

The default provider for a new connection is the Native OLE DB\SQL Server Native Client provider. Other supported providers include:

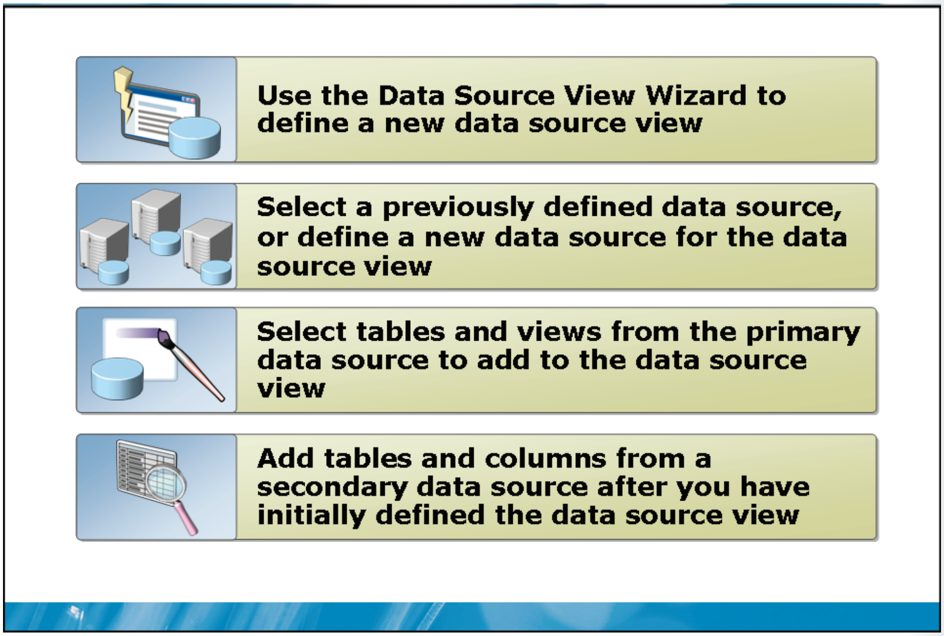
- SQL Server 7.0 using the SQL OLE DB Provider or the .NET native OLE DB provider (x86, x64, and ia64).
- SQL Server 2000 using the SQL OLE DB Provider or the .NET native OLE DB provider (x86, x64, and ia64).
- SQL Server using the SQL OLE DB Provider or the .NET native OLE DB provider (x86, x64, and ia64).
- Oracle 9.0 using the Microsoft OLE DB Provider for Oracle or the .NET native OLE DB provider (x86 only).

- IBM DB2 8.1 using Microsoft OLE DB Provider for DB2 (x86, x64, ia64) - only available for Microsoft SQL Server 2005 Enterprise Edition or Microsoft SQL Server 2005 Developer Edition and downloadable as part of the Feature Pack for Microsoft SQL Server Service Pack 1.
- Microsoft Access® with Microsoft Jet 4.0 OLE DB provider (x86 only).
- Teradata v2R6 with OLE DB 1.3 provider from NCR (x86 only).

If you have an existing data source defined in an Analysis Services database or project and wish to create a new data source object that connects to the same underlying data source, you can simply copy properties of the first data source object into a new data source object.

Question: What providers would you need to use in your own organization to make data available to Analysis Services?

Creating a Data Source View

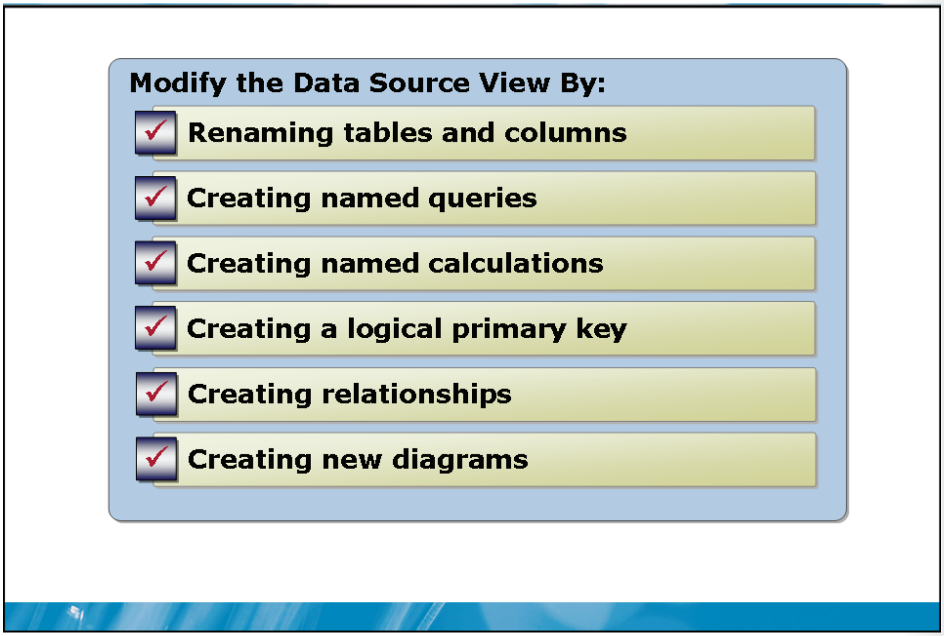


Key Points

Analysis Services design tools use data source views to maintain a cache of relational metadata and to take advantage of some of the annotations within a data source view.

- By describing a subset of tables and views in a data source, a data source view makes available only the tables required by OLAP and data mining objects.
- A data source view handles the layout of tables, filters, SQL expressions, relationships, and other complexities of the schema. Therefore, a data source view enables simple bindings by Analysis Services cubes, dimensions, and mining models to the tables and columns in the data source view.
- Data source views can be defined using the Data Source View Wizard in BI Development Studio. One of the benefits of data source views is that they can bring together data from multiple data sources. To do this, you can specify secondary data sources in the Data Source View Designer. You can also use the Data Source View Designer to browse source data.

Modifying a Data Source View



Key Points

Using the Data Source View Designer, you can make several modifications to a data source view without modifying the underlying source data or database schemas. For example, you can:

- Use the FriendlyName property to specify a name for a column from a table or view that is easier for users to understand or more relevant to the subject area.
- Create a named query and named calculations that lets you extend the relational schema of existing tables in a data source view without modifying the underlying data source.
- Create logical primary keys and relationships for improved performance.
- Create diagrams to reduce the visual clutter when you only want to view a subset of the tables in the data source view.

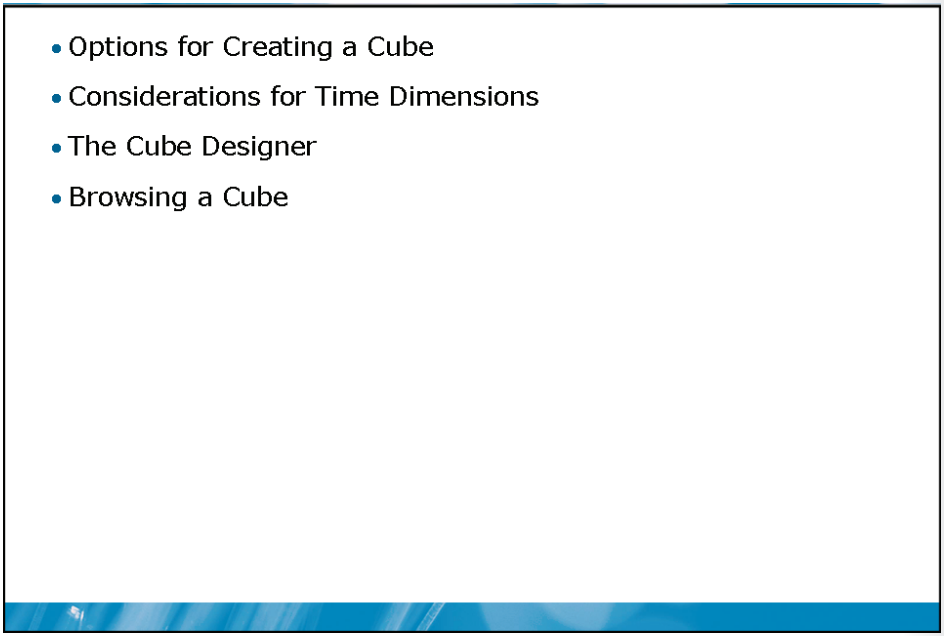
Demonstration: Using Data Source Views

In this demonstration, you will see how to:

- Create a data source
- Create a data source view
- Modify a data source view

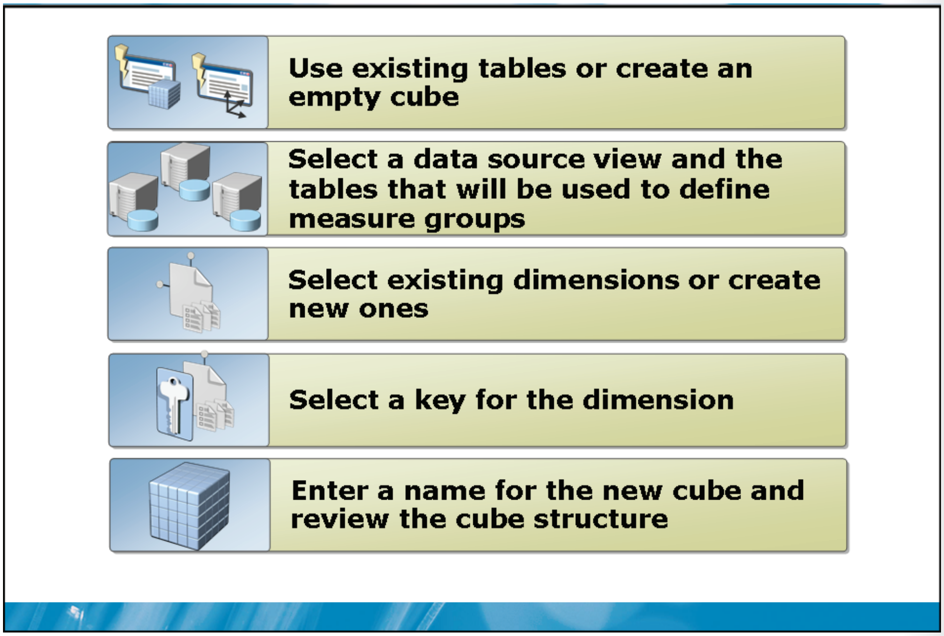
Question: What are some of the name matching methods available when creating a data source view?

Lesson 3
Creating a Cube



A cube is a multidimensional structure that contains dimensions and measures. Dimensions define the structure of the cube, and measures provide the numerical values of interest to the end user. As a logical structure, a cube allows a client application to retrieve values as if cells in the cube defined every possible summarized value.

Options for Creating a Cube

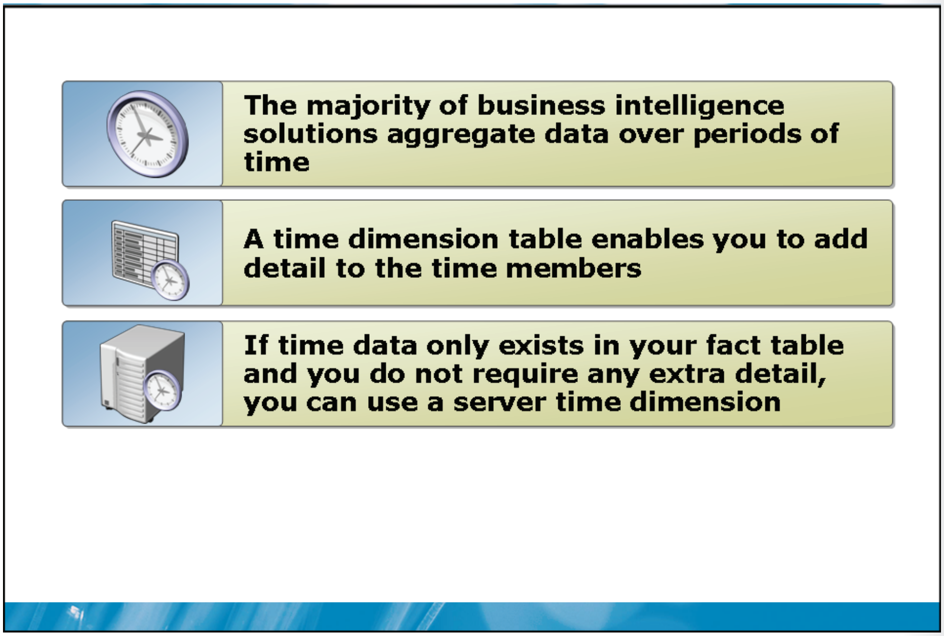


Key Points

Use the Cube Wizard to create a cube quickly and easily.

- The Cube Wizard guides you through the steps to specify the data source view and measures in the cube.
- When you create the cube, you can add existing dimensions or create new dimensions that structure the cube.
- You can also create dimensions separately, using the Dimension Wizard, and then add them to a cube.
- You can create a cube by choosing the relevant data source and data source view or you can build the cube without using a data source and then subsequently generate the database schema.
- You can use the Cube Wizard to automatically build attributes and hierarchies, or you can choose to define them in the Cube Designer later.

Considerations for Time Dimensions



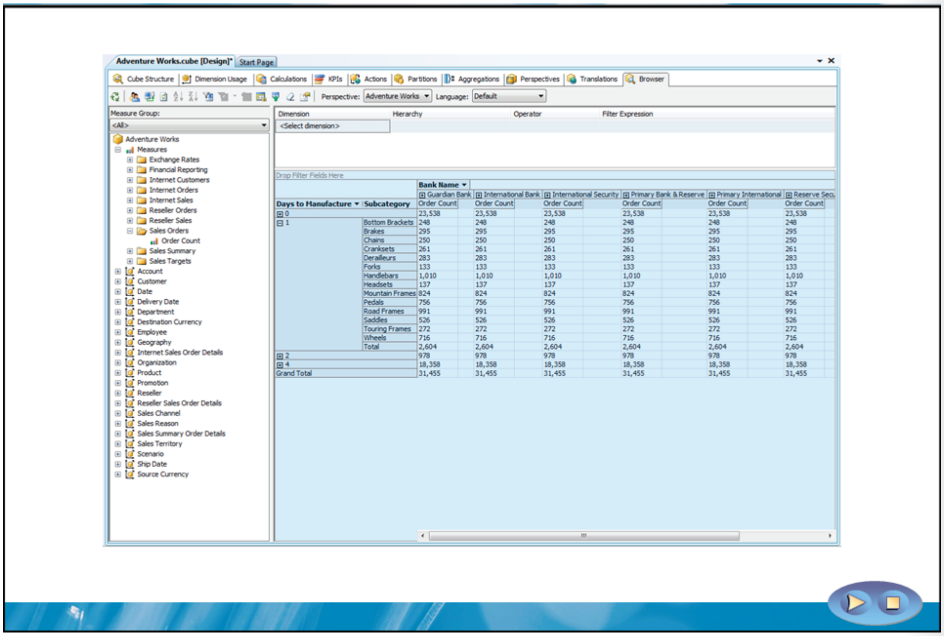
Key Points

Most cubes include a time dimension. Some solutions do not use time and instead use another factor to delineate the data, such as a unit of work or a project.

- You can create multiple time dimensions or one time dimension with multiple hierarchies.
- Time dimensions are distinct from other dimensions because Analysis Services contains inherent functionality to group the members into levels.
- A time dimension table enables you to add detail to the time members.
- Server time dimensions contain hierarchies, levels and attributes, but these are stored on the server rather than in a separate dimension table.

Question: When would it be appropriate to use a Server Time dimension instead of a Time dimension?

The Cube Designer



Key Points

The Cube Designer in BI Studio lets you view and edit the properties of the cube and its objects, and to browse cube data.

- You can modify cube structure, dimension usage, and calculations.
- You can add Key Performance Indicators (KPIs), and actions, change cube partitioning and storage, create aggregations, create perspectives on cube data, and add translations to localize cube data.
- You can also browse the cube, to see how modifications will affect the cube.

Question: How do you modify storage, caching, and writeback settings for a cube?

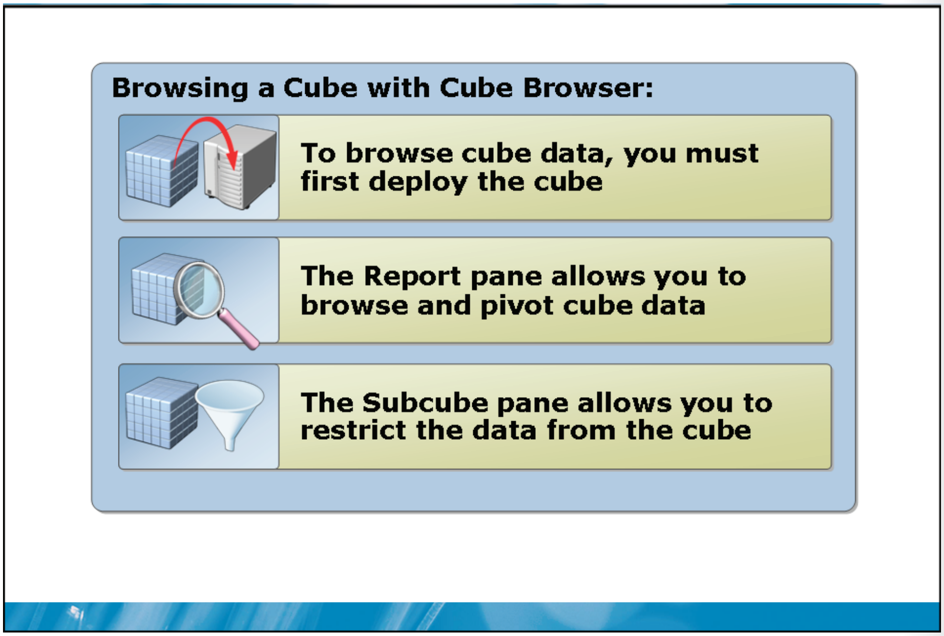
Demonstration: Designing a Cube

In this demonstration, you will see how to:

- Create a cube using the Cube Wizard
- Modify the cube using the Cube Designer

Question: How can you use the Cube Designer to configure a cube once it has been created?

Browsing a Cube



Key Points

Before browsing cube data in the Cube Browser tab, the cube must be deployed.

- Once deployed, you can use the Report pane to browse and pivot the cube to see the data from the different perspectives.
- The Subcube pane allows you to filter the cube, restricting the cube data.

Question: Why would you want to use a subcube?

Demonstration: Browsing a Cube

In this demonstration, you will see how to:

- Deploy a cube
- Browse cube data

Question: When do you need to re-deploy a cube?

Lab: Creating a Multidimensional Analysis Solution

- Exercise 1: Creating a Data Source
- Exercise 2: Creating and Modifying a Data Source View
- Exercise 3: Creating and Modifying a Cube

Logon information

Virtual machine	NY-SQL-01
User name	Administrator
Password	Pa$$w0rd

Estimated time: 60 minutes

Exercise 1
Creating a Data Source

Scenario

You are a junior database developer at Adventure Works. The senior database developer has asked you to develop a new Analysis Services solution in order to demonstrate your ability with SQL Server 2008 to the database development team.

In this exercise, you will use Business Intelligence Development Studio to create a new Analysis Services project and a data source.

The main tasks for this exercise are as follows:

1. Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator.
2. Create a new SQL Server user.
3. Create a data source for the AdventureWorksDW relational database.

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

- Start 6234A-NY-SQL-01 and logon as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

- Use SQL Server Management Studio to create a new user on the AdventureWorksDW database.
- Name the user **sqlserver**, set the default schema to **dbo**, grant the user database role membership in **db_owner** and **db_securityadmin**.

▶ Task 3: Create a new Analysis Services project

- Use BI Development Studio to create a new Analysis Services project.
- Name the project **Adventure Works OLAP**, and save it in the **E:\Mod02-03\Labfiles\Starter** folder.

▶ Task 4: Create a data source for the AdventureWorksDW2008 relational database

- Use the Data Source Wizard in the Business Intelligence Development Studio.
- The AdventureWordsDW database is on the server NY-SQL-01.
- Use Windows Authentication to connect to the data source.
- Use the service account for the impersonation information.
- Name the data source **Adventure Works Data Warehouse**.

Results: After this exercise, you should see the Adventure Works Data Warehouse.ds data source in the Data Sources folder.

Exercise 2
Creating and Modifying a Data Source View

Scenario

In this exercise, you will build upon your Analysis Services solution in order to further demonstrate your competency with SQL Server 2008 to the database development team at Adventure Works.

The main tasks for this exercise are as follows:

1. Create a data source view.
2. Modify the data source view.

▶ Task 1: Create a data source view

- Use the Data Source View Wizard in the Business Intelligence Development Studio.
- Use the Adventure Works Data Warehouse data source.
- Include the **DimCustomer**, **DimGeography**, **DimProduct**, **DimTime**, and **FactInternetSales** tables.
- Name the data source view **Adventure Works DSV**.

▶ Task 2: Modify the data source view

- Use the Data Source View Designer in the Business Intelligence Development Studio.
- Change the **FriendlyName** of **FactInternetSales** to **InternetSales**.
- Change the **FriendlyName** of the remaining tables to remove the "**dim**" prefix from the name.
- Add a named calculation to the **Customer** table.
  - Name the named calculation **Full Name**.
  - The expression should concatenate the **FirstName**, **MiddleName** (if it exists), and **LastName** columns.

- Add named calculations to the **Date** table.
  - Create a calulation named **Simple Date** that uses the DATENAME function to concatenate the month, day, and year portions of the **FullDateAlternateKey** column.
  - Create a calculation named **Calendar Semester Description** that formats the semester string as "**H1 CY <year>**" for the first half of the year and as "**H2 CY <year>**" for the second half of the year.
  - Create a calculation named **Calendar Quarter Description** that formats the quarter string to place a "**Q**" before the quarter number of the year, followed by "**CY <year>**".

Results: After this exercise, you should see the Adventure Works DSV.dsv data source view in the Data Source Views folder.

Exercise 3
Creating and Modifying a Cube

Scenario

In this exercise, you will expand on your existing Analysis Services solution in order to demonstrate to the Information Services department of Adventure Works that you can work with cubes and cube data.

The main tasks for this exercise are as follows:

1. Create a cube.
2. Modify the cube.
3. Modify dimensions.
4. Modify attribute columns.
5. Browse the cube.

▶ Task 1: Create a cube

- Use the Cube Wizard in the Business Intelligence Development Studio to create a new cube.
- Build the cube using existing tables.
- Have the Cube Wizard suggest measure group tables.
- Use all suggested dimensions except for **Internet Sales**.
- Name the cube **Adventure Works Cube**.

▶ Task 2: Modify the cube

- Change the name of the **Unit Price Discount Pct** measure to **Unit Price Discount Percent**.
- Change the name of the **Tax Amt** measure to **Tax Amount**.

- Change **FormatString** to **Currency** for the following measures:
  - **Unit Price**
  - **Extended Amount**
  - **Discount Amount**
  - **Product Standard Cost**
  - **Total Product Cost**
  - **Sales Amount**
  - **Tax Amount**
- Change **FormatString** to **Percent** for the **Unit Price Discount Percent** measure.

▶ Task 3: Modify dimensions

- Open the **Customer** dimension in **Dimension Designer**.
- Add all the attributes from the **Geography** and **Customer** tables to the dimension.
- Open the **Product** dimension in **Dimension Designer**.
- Add all the attributes from the **Product** table to the dimension.
- Remove the **Large Photo** attribute from the dimension.
- Open the **Date** dimension in **Dimension Designer**.
- Add all the attributes except **Calendar Semester Description** and **Calendar Quarter Description** to the dimension.

▶ Task 4: Modify attribute columns

- Modify the **Calendar Quarter** attribute.
  - In the properties of the **Calendar Quarter** attribute, set the **KeyColumns** property so that the **Key Columns** are **CalendarYear** followed by **CalendarQuarter**.
  - Set the **NameColumn** property so that the **Source Column** value is **Calendar Quarter Description**.

- Modify the **English Month Name** attribute.
  - In the properties for the **English Month Name** attribute, change the **Name** property to **Month**.
  - Set the **KeyColumns** property so that the **Key Columns** are **Calendar Year** followed by **MonthNumberofYear**.
  - Set the **NameColumn** property so that the **Source Column** value is **EnglishMonthName**.
- Modify the **Calendar Semester** attribute.
  - In the properties of the **Calendar Semester** attribute, set the **KeyColumns** property so that the **Key Columns** are **CalendarYear** followed by **CalendarSemester**.
  - Set the **NameColumn** property so that the **Source Column** value is **Calendar Semester Description**.

▶ Task 5: Browse the cube

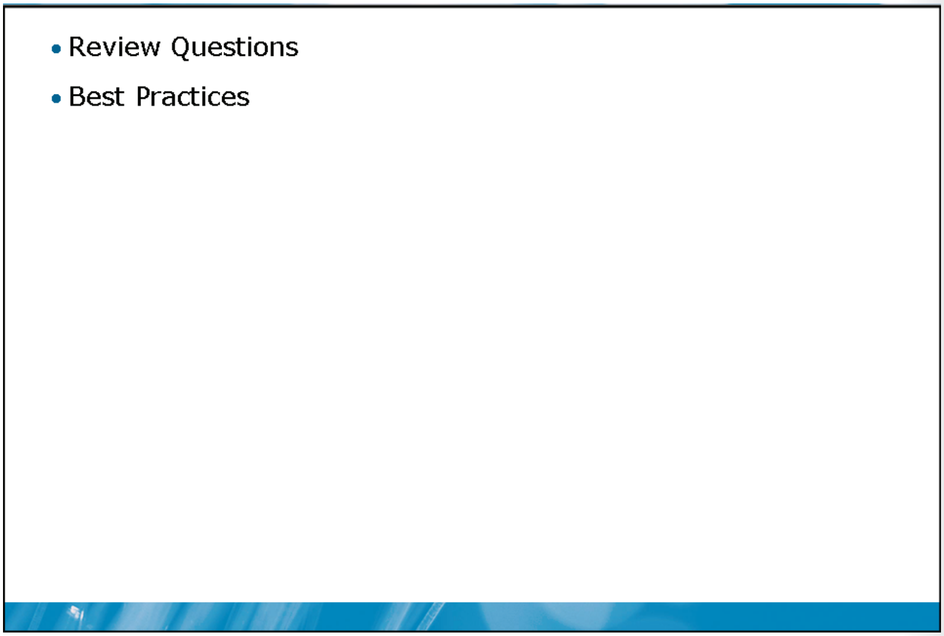
- Deploy the **Adventure Works OLAP** project.
- Use the **Cube Browser** to view the **Adventure Works Cube**.
- Add the **Sales Amount**, **Internet Sales Count** and **Unit Price Discount Percent** measures to the **Data** area to view the different **FormatString** fields you configured.
- Add the **Order Date.Calendar Year** hierarchy on rows.
- Browse the **Full Name** members of the **Customer** dimension to verify the named calculation you created.
- Open Microsoft Office Excel® 2007 and make a connection to **NY-SQL-01**.
- Select **Internet Sales Count** from the **Internet Sales** measure in the **PivotTable Field List**.
- Select **Full Name** from the **Customer** dimension in the **PivotTable Field List**.

Results: After this exercise, you should have successfully deployed the cube and reviewed the modifications made to ensure they are reflected when browsing the cube.

Lab Shutdown

After you complete the lab, leave the 6234A-NY-SQL-01 virtual machine running and do not discard any changes because the solutions created in this lab are used in the lab for module 3.

Module Review and Takeaways



Review Questions

1. What is the purpose of using a data source view?
2. What does a Named Calculation do?
3. What makes the Time dimension unique from other dimensions?

Best Practices Related to Data Source Design

Supplement or modify the following best practices for your own work situations:

- **Use only supported OLEDB providers in a Data Source**. Analysis Services was designed and tested to work with a specific set of OLE DB providers. Although other OLE DB providers are available, and the Data Source Wizard lets you choose any compatible provider, the capabilities and behavior of different providers can differ substantially. This is true even for different providers that connect to the same database. Therefore, you should use only the supported providers.
- **Do not use the .NET SqlClient Data Provider to connect to a SQL Server data source**. Because the Analysis Services server runs in native code, you can get better performance by using a native provider. Therefore, do not use the .Net SqlClient Data Provider; instead, use the Microsoft OLE DB Provider for SQL Server or the SQL Native Client provider.

Best Practices Related to Cube Design

Supplement or modify the following best practices for your own work situations:

- **Avoid including unrelated measure groups in the same cube**. Having many measure groups in a cube can adversely affect the query performance of the cube, even for queries that only use a single measure group. If your DSV contains many potential fact tables and they will not usually be used together, consider creating multiple smaller cubes containing topic areas that are used together. Then, you can enable occasional cross-topic querying by creating linked measure groups that join together all measure groups in a special cross-topic cube.

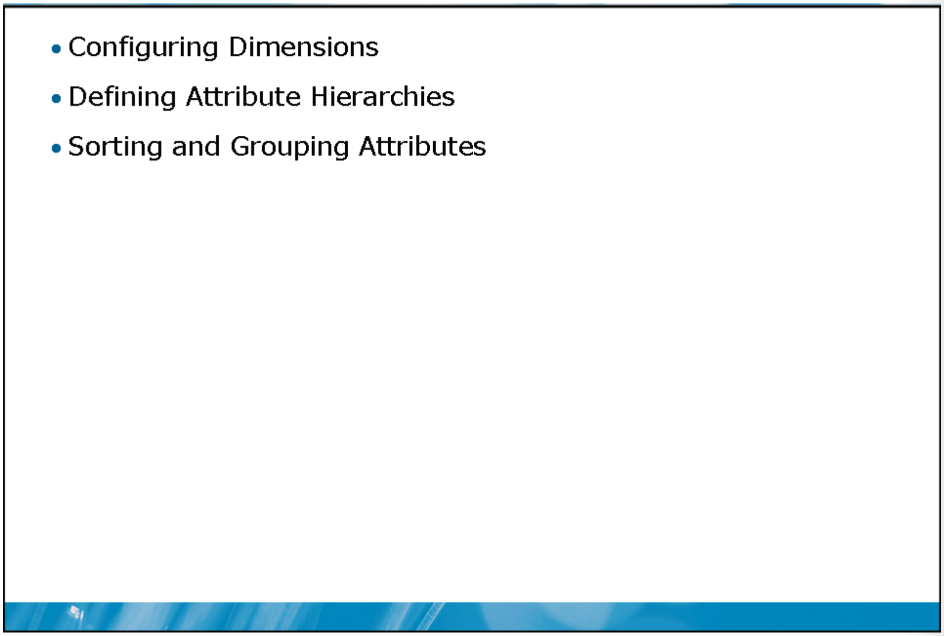
- **Do set AttributeHierarchyEnabled to False on any cube attributes that are below the level of granularity of all measure groups in the cube**. Attributes are considered below the level of granularity for a measure group when no chain of attribute relationships exists between the granularity attribute and the attribute in question. Such attributes will never have any data associated with them in the cube and will only confuse end-users. For example, if a time dimension contains day, week, month and year attributes, whereas an inventory cube contains only data updated on a weekly basis, it is best to disable the day attribute for this cube by setting AttributeHierarchyEnabled to False for the day attribute only.
- **Do use the smallest numeric data type possible for measures**. The data type of measures must be large enough to hold the largest aggregated value (the "all" value) but should be no larger than necessary to reduce storage costs.

Module 3

Working with Cubes and Dimensions

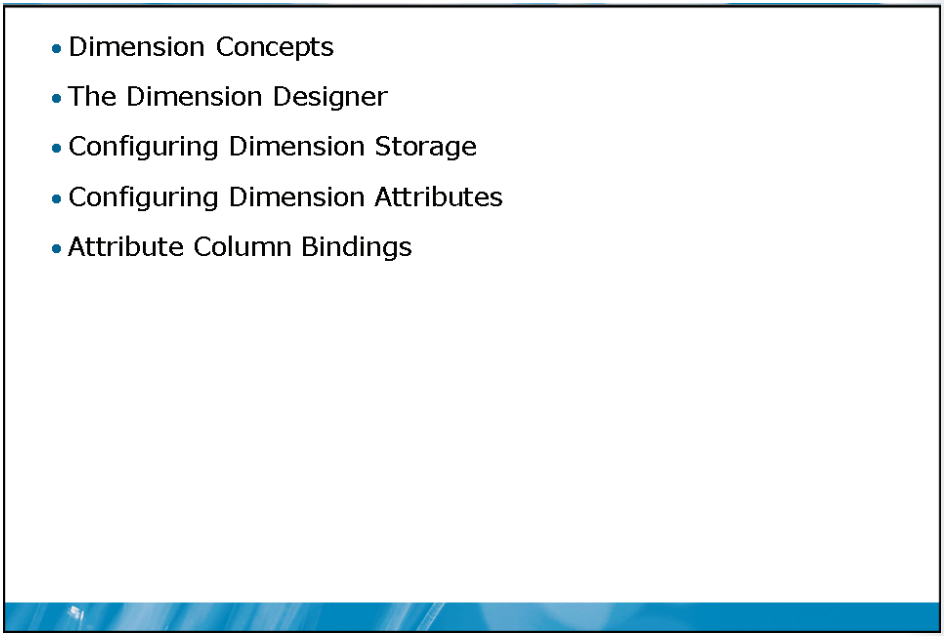
Contents:

Module Overview



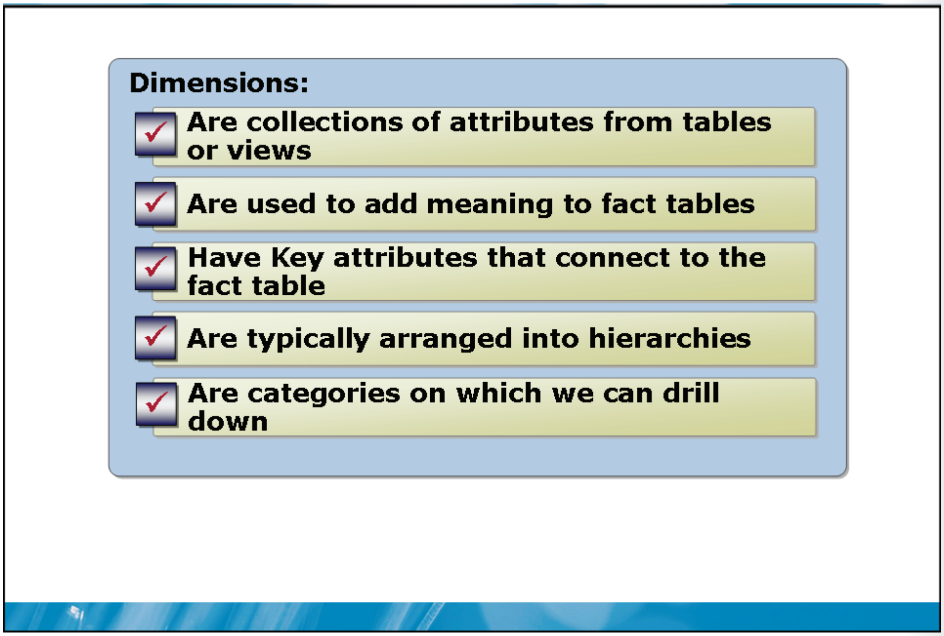
In Microsoft® SQL Server® Analysis Services, dimensions are a fundamental component of cubes. Dimensions organize data with relation to an area of interest, such as customers, stores, or employees, to users. Dimensions in Analysis Services contain attributes that correspond to columns in dimension tables. These attributes appear as attribute hierarchies and can be organized into user-defined hierarchies, or can be defined as parent-child hierarchies based on columns in the underlying dimension table. Hierarchies are used to organize measures that are contained in a cube.

Lesson 1
Configuring Dimensions



All Analysis Services dimensions are groups of attributes based on columns from tables or views in a data source view. Dimensions exist independent of a cube, can be used in multiple cubes, can be used multiple times in a single cube, and can be linked between Analysis Services instances. A dimension that exists independent of a cube is called a database dimension and an instance of a database dimension within a cube is called a cube dimension.

Dimension Concepts



Key Points

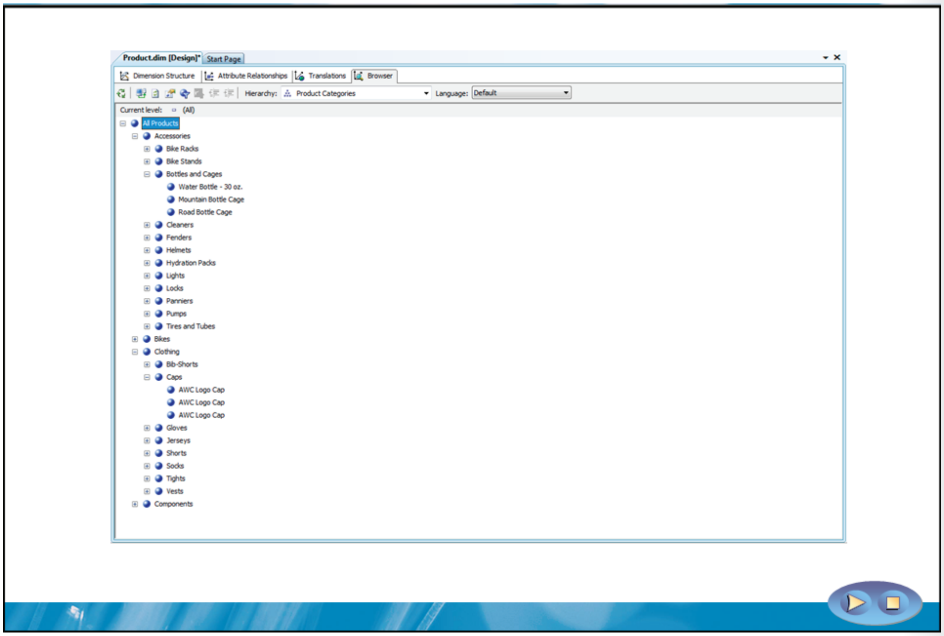
Dimensions form the contexts for the facts, and define the aspects of the business by which the facts are aggregated. For example, Product and Time dimensions might provide context to the SalesValue measure.

- Dimensions are collections of attributes from tables or views. You can use dimensions in multiple cubes and link to them from remote instances.
- Attributes add meaning to dimensions. Each column in the dimension table can provide an attribute that provides a piece of information about the dimension member. For example, a RetailStore dimension might have Manager and PostalCode attributes.

- The Key attribute is typically the primary key of the dimension table. The Key attribute is the column in the dimension table that links to the fact table.
- Attributes are typically arranged into hierarchies that define the drill-down paths through aggregations. Each layer of the hierarchy is called a level. For example, a Time dimension might have a hierarchy with Day, Month, and Year levels, but other hierarchies could be defined with attributes such as Week, Quarter, and Century.

Question: How do dimensions, attributes, and fact tables relate to one another?

The Dimension Designer

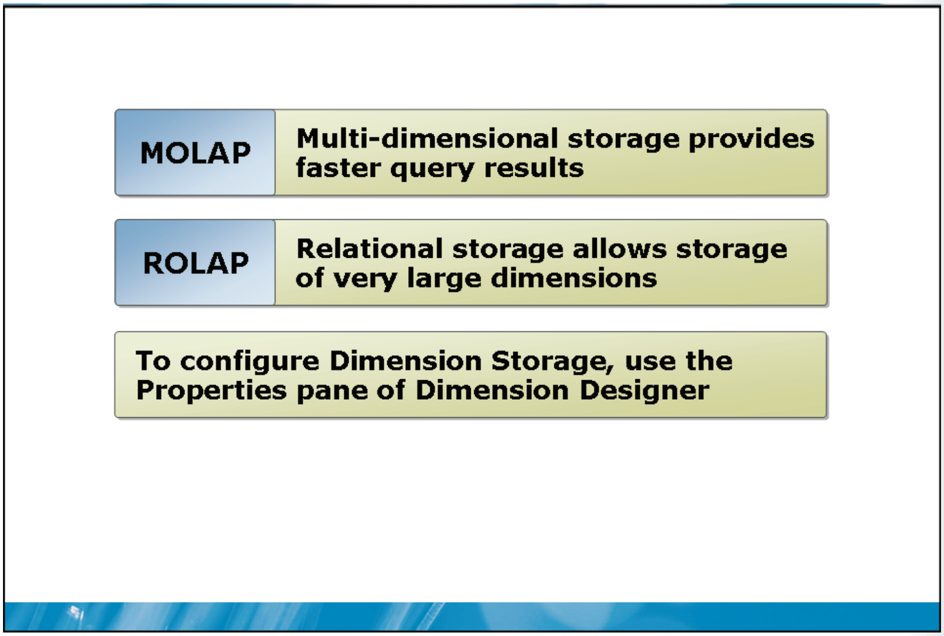


Key Points

Use the Dimension Designer to edit the attributes, levels, hierarchies, and translations of a dimension, and to browse the dimension.

- Use the Dimension Structure tab to view and edit the attributes, levels, and hierarchies of the dimension.
- Use the Attribute Relationships tab to create, modify or delete the attribute relationships of the selected dimension.
- Use the Translations tab to view and edit the multi-language translations for the attributes, levels, and hierarchies.
- Use the Browser tab to browse members of each hierarchy in the dimension. You can only browse members after you have deployed the solution.

Configuring Dimension Storage

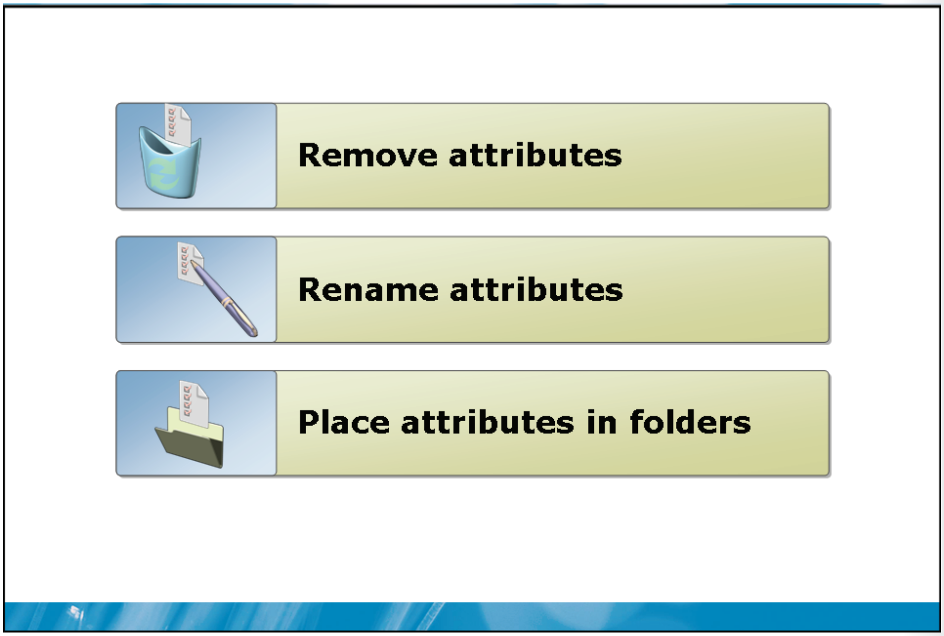


Key Points

The two dimension storage modes in Analysis Services are Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP). These storage modes define where, and in what structure type, the dimension data is stored.

- Data for a dimension that uses MOLAP is stored in a multidimensional structure in the instance of Analysis Services. This multidimensional structure is created and populated when the dimension is processed. MOLAP dimensions provide better query performance than ROLAP dimensions.
- Data for a dimension that uses ROLAP is actually stored in the tables used to define the dimension. The ROLAP storage mode can be used to support large dimensions without duplicating large amounts of data, but at the expense of query performance. Because the dimension relies directly on the tables in the data source view used to define the dimension, the ROLAP storage mode also supports real-time OLAP.

Configuring Dimension Attributes



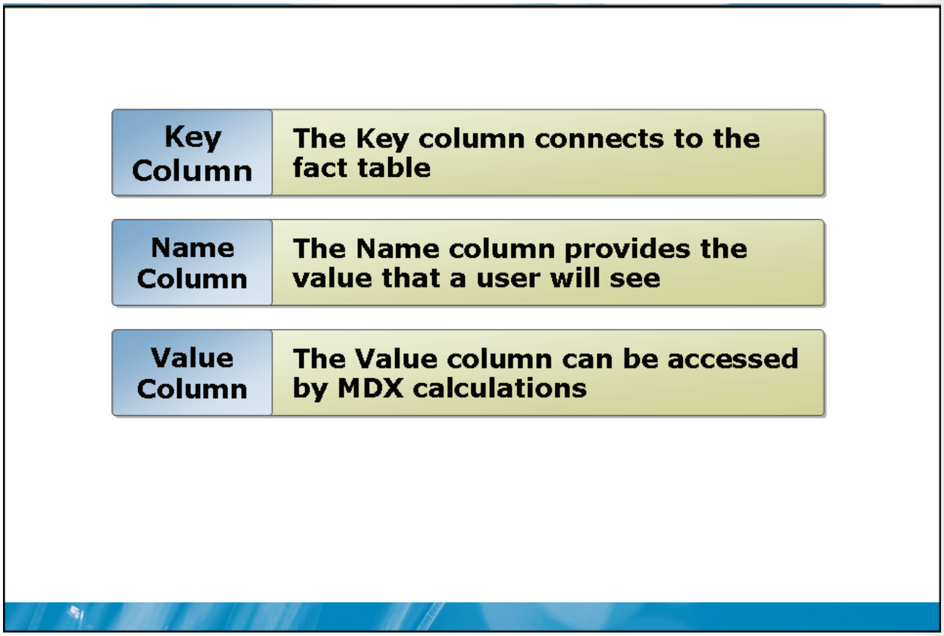
Key Points

The Cube Wizard and Dimension Wizard set attributes on a dimension. Other wizards and elements of the Analysis Services user interface may further modify these attributes. These default settings are sufficient in most situations, but you can use Dimension Designer to edit the attributes if necessary.

- You can remove attributes from the dimension by right-clicking the attribute and clicking Delete. Removing attributes from a dimension does not affect the Data Source View, which makes it possible for a Data Source View to be used by multiple dimensions without forcing each dimension to use all of the attributes.

- You can rename an attribute to provide a more meaningful or user-friendly name than the dimension table. You can do this by right-clicking it in the Dimension Structure tab of Dimension Designer. You can also set its Name property in the Properties window or edit it directly if the Attributes pane is in grid view.
- Having many attributes for a dimension can be confusing for users. You can organize attributes into Display Folders to simplify browsing. This only affects the way client applications display the dimension and has no other effects on the hierarchies or attributes. After you have created Display Folders, you should deploy the solution and reconnect to see the results on the Cube Browser tab.

Attribute Column Bindings



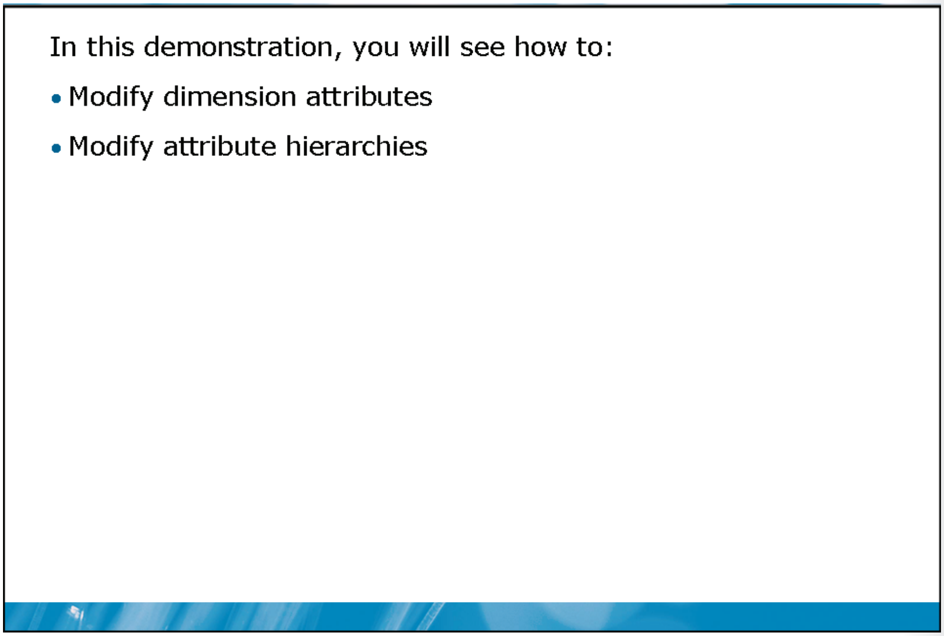
Key Points

To control the output from attributes, you can define the column that connects to the fact table, the column that users see, and an optional value column that you can use for MDX calculations.

- The KeyColumn is the column, or columns, that are used to connect to the fact table. This is typically the primary key of the dimension table, and is used when you choose to order the hierarchy by key.

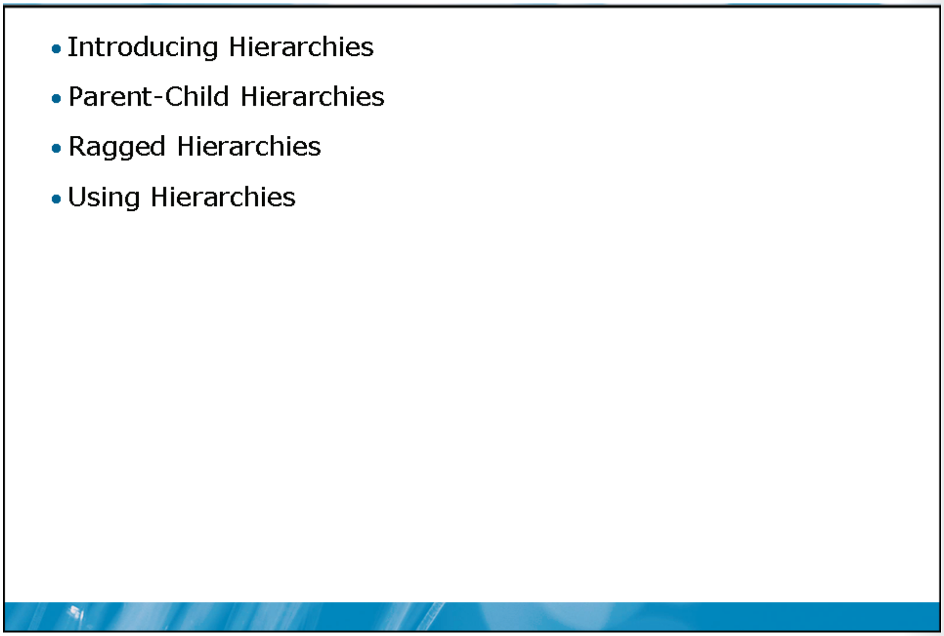
- The NameColumn provides the value that a user will see. This can provide more useful information to a user than the key column. You can define a name column as a calculated column that you have created in a Data Source View. For example, an Employee dimension might have an EmployeeID key column, but you might want the name of the employee to be displayed. To do this, you can create a calculated column based on first name and last name in the Data Source View, and then use this as the name column.
- The ValueColumn is returned by the MDX MemberValue function. This allows you to create calculations that are based on a value other than the name or key. For example, a time dimension is stored in a dimension table with one row for each day. The key column is a SMALLINT and the name Column is a CHAR(11) that displays the day, month, and year. The value column contains the date as a SMALLDATETIME. This allows calculations to use the true date value rather than converting it from string values.

Demonstration: Designing Dimensions



Question: What are some of the ways you can modify dimensions?

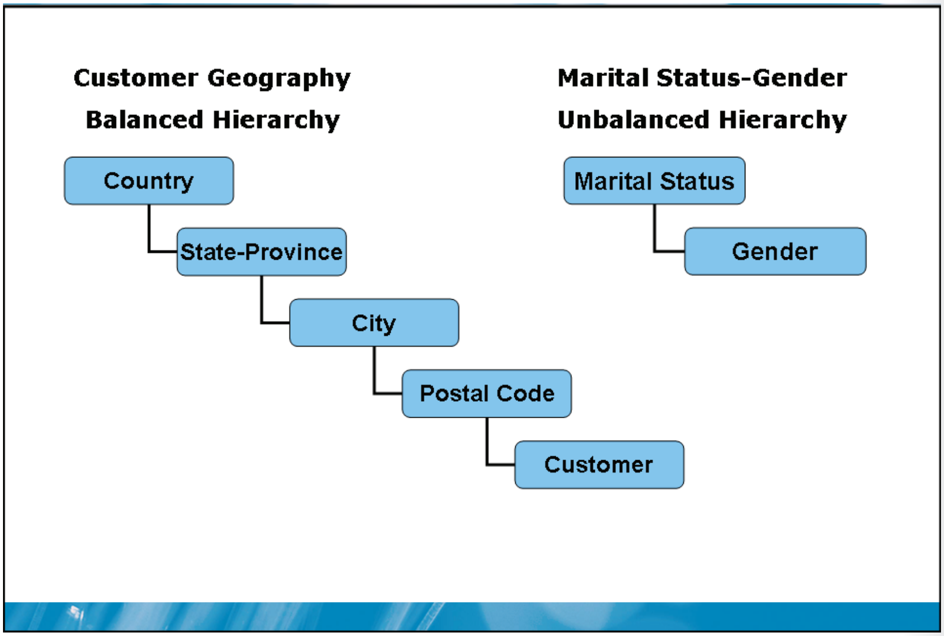
Lesson 2
Defining Attribute Hierarchies



Attributes are exposed to users through attribute hierarchies. An attribute hierarchy in a dimension includes an optional All level and the distinct members of the attribute. For example, a Customer dimension might include a Name attribute hierarchy with two levels: the All level and a second level with a member for each name. It is attribute hierarchies that define the space of a cube. You can think of a cube as the multidimensional space created by the product of its attribute hierarchies. Dimensions are containers for attribute hierarchies. A dimension can also contain user hierarchies as a navigational convenience, but these do not affect the space in the cube.

Defining relationships between levels in a hierarchy enables Analysis Services to define more useful aggregations to increase query performance and can also save memory during processing performance, which can be important with large or complex cubes.

Introducing Hierarchies



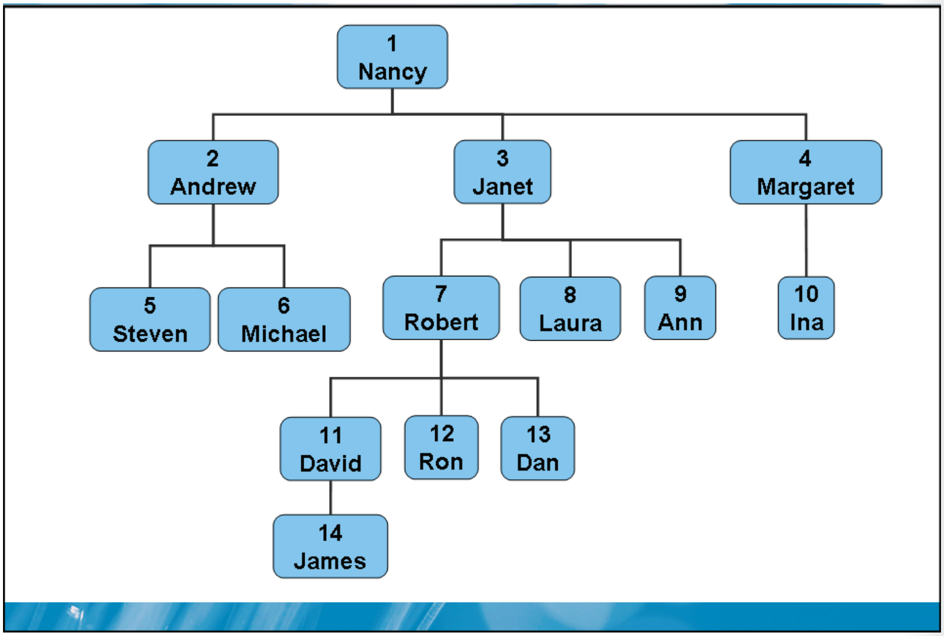
Key Points

When attributes are arranged into user-defined hierarchies, you define relationships between hierarchy levels when levels are related in a many-to-one or a one-to-one relationship (called a natural relationship). For example, in a Calendar Time hierarchy, a Day level should be related to the Month level, the Month level related to the Quarter level, and so on.

- A natural hierarchy is composed of attributes where each attribute is a member property of the attribute below. For example, the Customer Geography hierarchy Country, State-Province, City, Postal Code and Customer is a natural hierarchy because of the relationships between the attributes. The hierarchy Marital Status-Gender is a non-natural hierarchy because the Marital Status and Gender attributes do not have a relationship to each other.

- Natural hierarchies are preferred for performance reasons. The Dimension Designer will warn you if you create a hierarchy that is non-natural.

Parent-Child Hierarchies



Key Points

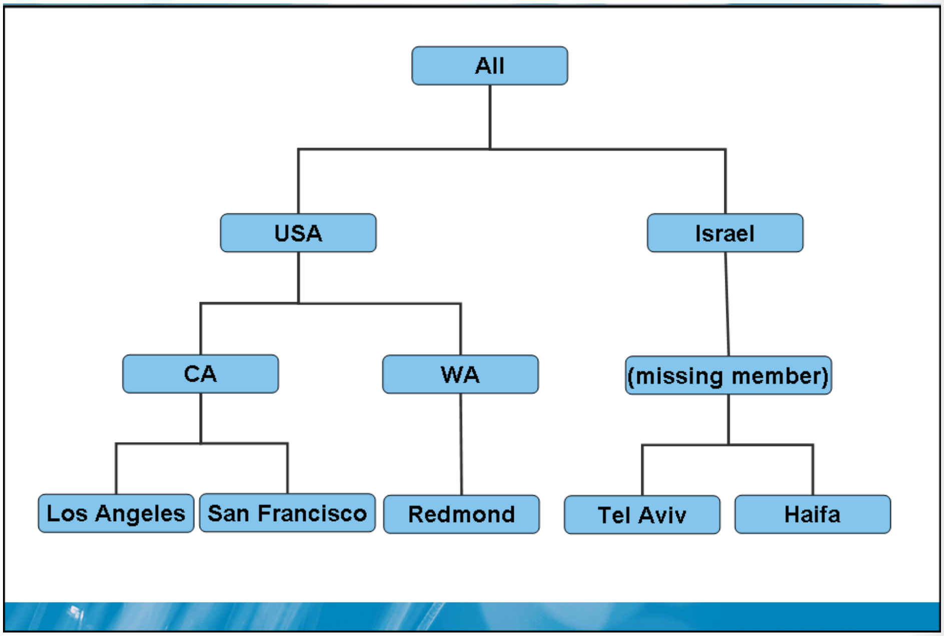
A parent-child hierarchy exists when you have a self-referencing dimension table. For example, an Employee dimension might have a Manager attribute. Only the most senior employee has no manager; all other employees have a position in the hierarchy that is defined by the identities of their managers and of those that they manage.

- A parent-child hierarchy is an unbalanced hierarchy. This occurs when the number of levels from leaf to root is different for different leaf members. For example, some managers might have three levels of employee working for them while others have just two.

- Parent-child hierarchies are not defined by the order in which attributes are added, and are defined by the relationships in the dimension table instead.

Question: Aside from organizational charts, where else might you find parent-child hierarchies?

Ragged Hierarchies



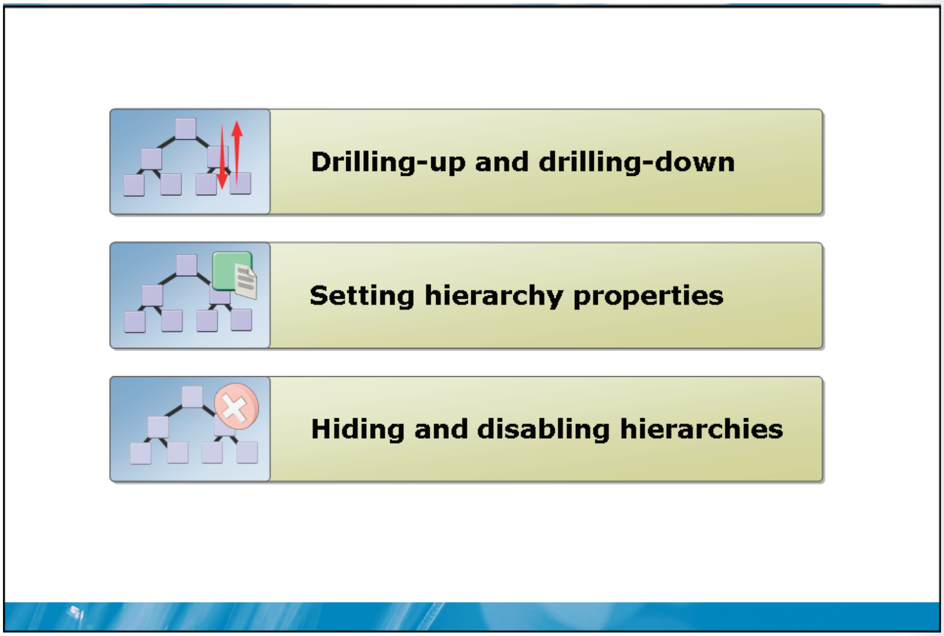
Key Points

Ragged hierarchies often seem similar to parent-child hierarchies to the user. Like a parent-child hierarchy, a ragged hierarchy will also have a different number of levels in different parts of the hierarchy, but these levels are formed from different columns in the dimension table rather than from a self-referencing relationship. For example, in a Location dimension you might have Location, Region, State and Country levels, but the State level is only used for the United States, Canada, and Australia. All other countries skip the State level.

- The HideMemberIf property makes a regular hierarchy ragged. There are five possible values for this property:
  - **Never**: This value creates a regular hierarchy.
  - **OnlyChildWithNoName**: This value hides a level member when it is an only child and is null or a zero-length string.
  - **OnlyChildWithParentName**: This value hides a level member when it is an only child and it has the same name as its parent.

  - **NoName**: This value hides a level member when it is null or a zero-length string.
  - **ParentName**: This value hides a level member when it has the same name as its parent.
- Use the SkippedlevelsColumn property to make a parent-child hierarchy ragged. This allows a leaf level salesperson to report directly to the CEO while still remaining at the leaf level.
- The SkippedlevelsColumn property points to a column in the data source view that contains the number of levels skipped for each member.

Using Hierarchies



Key Points

You typically create user-defined hierarchies to allow users to drill-up and drill-down through the data, and to allow Analysis Services to create meaningful aggregations of the measures.

- There are several important hierarchy properties:
  - The IsAggregatable property defines whether an All level is created. In some scenarios, an All level is irrelevant. For example, the Time dimension in a solution holds data from 12th January 1995, because earlier data is too costly to convert and gives minimal business benefit. This range of dates is not relevant to the business and does not need to be aggregated.
  - The AttributeHierarchyOrdered property defines whether the hierarchy is ordered. If this is set to False and you do not query the attribute hierarchy, you will reduce processing time. If you only use an attribute hierarchy to order another attribute hierarchy, it is not queried.

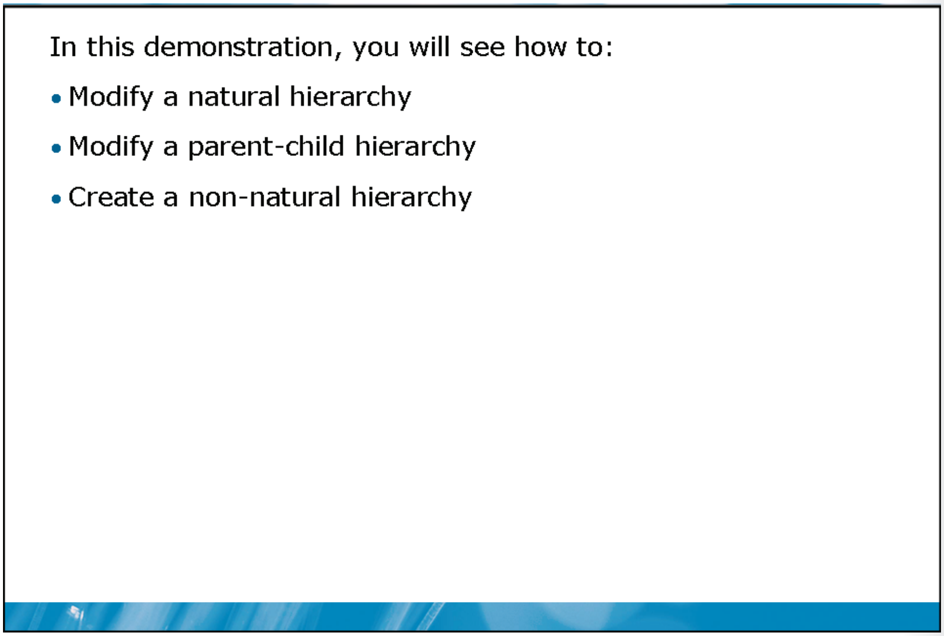
    - The AttributeHierarchyOptimizedState can be set to NotOptimized to prevent Analysis Services from creating indexes on the attribute hierarchy. This will reduce processing time but increase query time, and you should only use this when you do not query the attribute hierarchy.

- Hiding and disabling attribute hierarchies can improve performance and focus data for users.

    - If you set the AttributeHierarchyEnabled property to False, the hierarchy is disabled and Analysis Services only creates the attribute hierarchy as a member property. This is useful if the attribute is providing detail, but you do not want to use it as a level for aggregation.

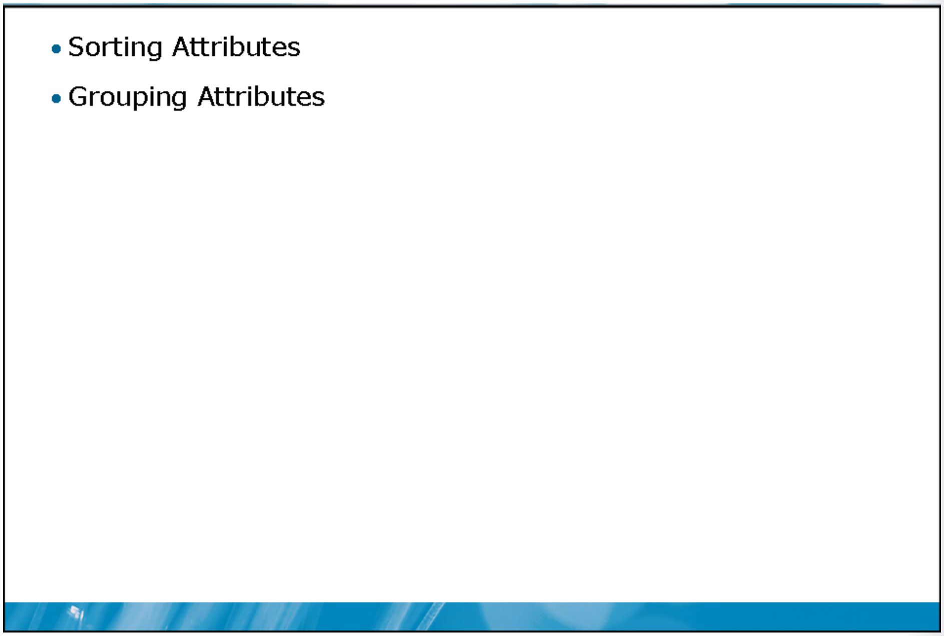
    - If you set the AttributeHierarchyVisible property to False, the attribute is only visible from user-defined hierarchies. This is useful if there are large numbers of distinct values in the attribute hierarchy that would cause confusion and would not add any benefit.

Demonstration: Creating and Using Hierarchies



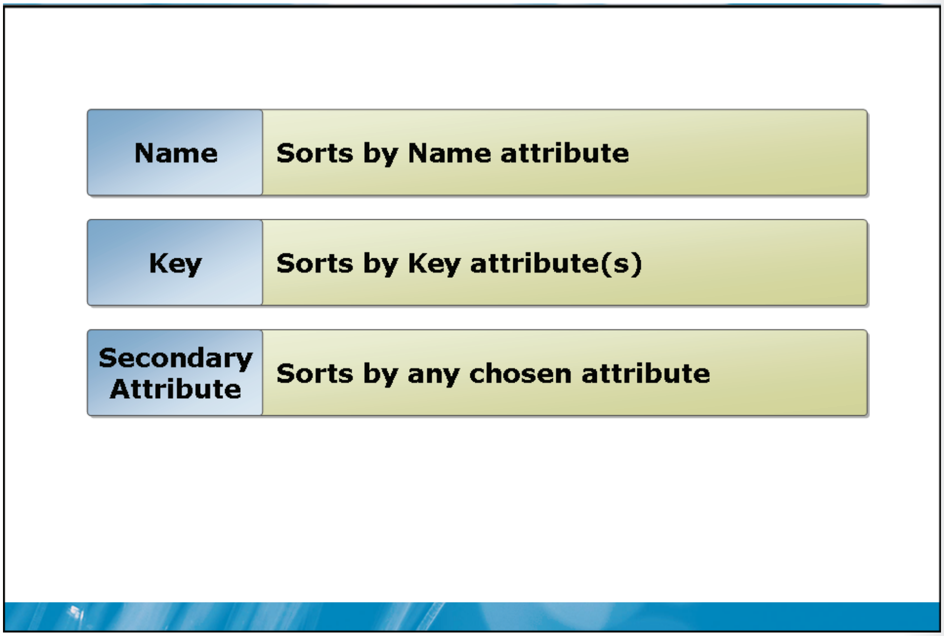
Question: What kind of hierarchies would you implement at your company and why?

Lesson 3
Sorting and Grouping Attributes



A cube can be difficult to navigate if it has numerous attributes and attribute hierarchies. In Analysis Services you can sort by the member name, the member key, or by a related attribute. You can also group attributes. A member group is a system-generated collection of consecutive dimension members. To improve usability in Analysis Services, members of an attribute can be grouped into a number of member groups through a process called discretization.

Sorting Attributes



Key Points

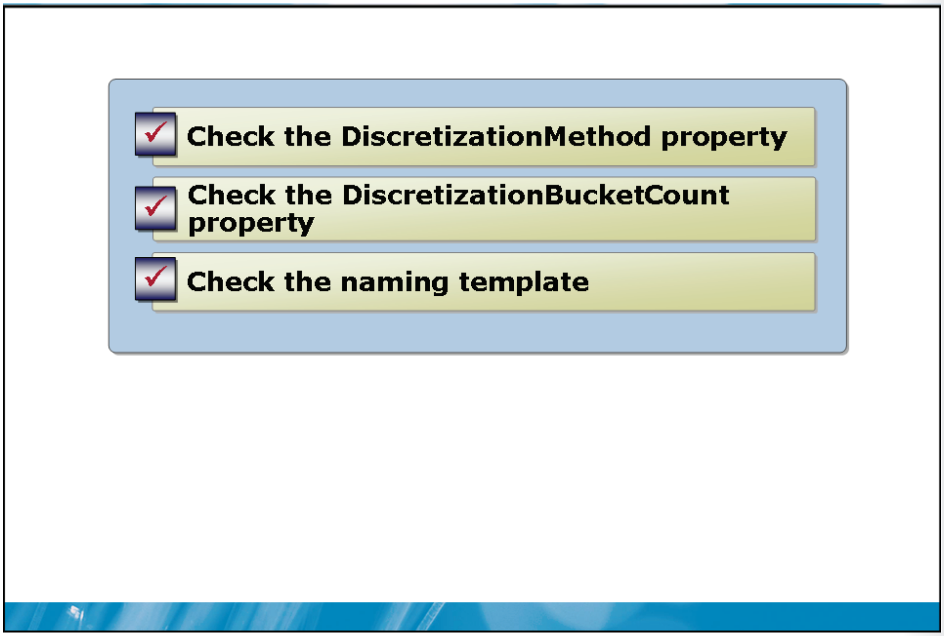
You can choose any attribute to sort the hierarchy by. This can be the Key, the Name, or any other secondary attribute in the dimension table.

- If you order by name, Analysis Services will order the members in alphanumeric order.
- If you order by key, you can specify one or more key columns and use these to sort the members. This allows you to sort dates by quarters. With a single key, the order of quarters would be Q1, Q2, Q3, and Q4 with all years aggregated, for example, Q1 would aggregate all Q1 data from every year. If you add the Year column to the key, the order would be Q1 2000, Q2 2000, Q3 2000, Q4 2000, Q1 2001, Q2 2001, Q3 2001, and so on.

- You can also order by a secondary attribute. This attribute can be a standard column or a calculated column created in the Data Source View. For example, you want to sort a Course dimension. The Name of the course runs from "Course 1" to "Course 450". The Key of the course is provided by the course vendor and is not a relevant sort order. If you sort by the Name, the order is "Course 1", "Course 11", "Course 111", "Course 2", "Course 22", and so on. Therefore, you create a new column in the dimension table with the course number as a numeric field. This can then be used to sort the data in the correct order.

Question: When might you want to use a secondary attribute for sorting data?

Grouping Attributes



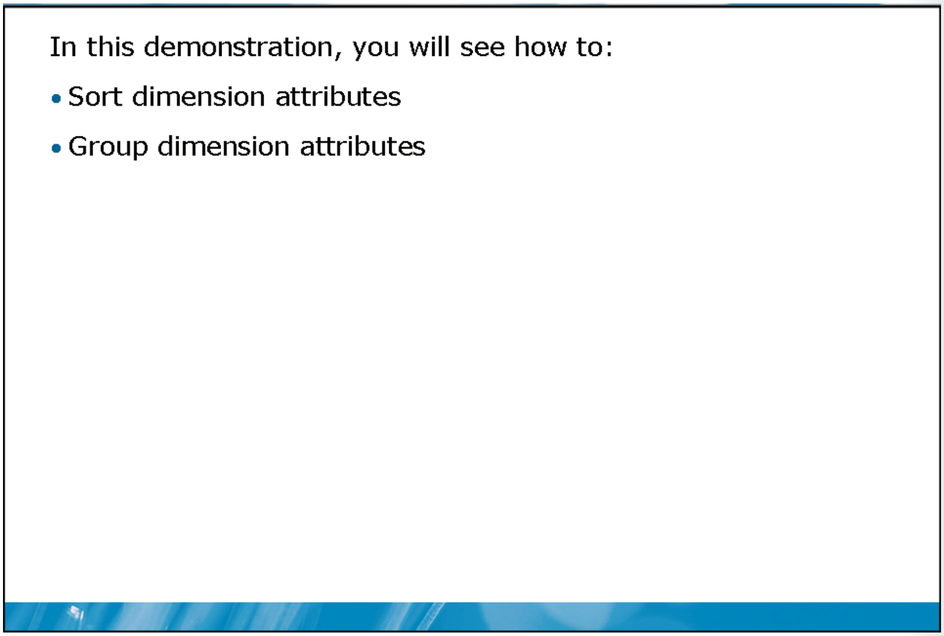
Key Points

Some hierarchies have no natural levels, so Analysis Services only creates an All level and the leaf level. This can make the cube difficult to navigate if there are many members of the hierarchy. You can use grouping to organize these members into groups to simplify cube browsing. For example, customers could be grouped into income brackets rather than showing individual incomes. You can then drill-down into the groups to show the detail as needed.

- To enable grouping, you must set the DiscretizationMethod property of the attribute. This can be set to:
  - Equal Areas, to divide the members into groups with an equal number of members.
  - Clusters, to use a clustering algorithm to group members based on the training data. This can form useful groups, but has a higher processing cost.

- To specify the number of groups, you must set the DiscretizationBucketCount property. The default is the square root of the number of distinct members.
- To specify a naming template, you must set the Format option for the NameColumn property of an attribute. The default naming template displays the first and last group members in the format "January – March". You can create your own naming templates to modify group names. Because naming templates are based on the members that they contain, the names of groups can change as members are added or removed.

Demonstration: Using Sorting and Grouping



Question: What are some sorting or grouping scenarios that you would implement at your company and why?

Lab: Defining Dimensions

- Exercise 1: Configuring Dimensions
- Exercise 2: Defining Relationships and Hierarchies
- Exercise 3: Sorting and Grouping Dimension Attributes

Logon information

Virtual machine	NY-SQL-01
User name	Administrator
Password	Pa$$w0rd

Estimated time: 90 minutes

Exercise 1
Configuring Dimensions

Scenario

Several users of the Adventure Works company have reported that they are having problems with browsing the cube because the cube is just too complex. The senior database developer has requested that you add dimensions to the existing cube space to improve usability of the cube. In this exercise, you must configure additional settings for several dimensions in the cube.

The main tasks for this exercise are as follows:

1. Open the Adventure Works OLAP solution.
2. Remove attributes from the Customer dimension.
3. Remove an attribute from the Product dimension.
4. Add dimension intelligence.

▶ Task 1: Open the Adventure Works OLAP solution

- Start 6234A-NY-SQL-01 and logon as **Administrator** using the password **Pa$$w0rd**.
- Use Business Intelligence Development Studio to open the **Adventure Works OLAP** solution.
- The **Adventure Works OLAP.sln** file is located in the **E:\Mod02-03\Labfiles\Starter\Adventure Works OLAP** folder.

▶ Task 2: Remove attributes from the Customer dimension

- Use the Dimension Designer to modify the **Customer** dimension.
- The company will not be using translations at this time, so remove all attributes that start with the words “**French**” or “**Spanish**”.

▶ Task 3: Remove an attribute from the Product dimension

- Use the Dimension Designer to modify the **Product** dimension.
- Remove the **Thai Product Description**.

▶ Task 4: Add dimension intelligence

- Start the Business Intelligence Wizard on the **Date** dimension.
- Add dimension intelligence.
- Specify that the **Date** dimension is a **Time** dimension.
- Map the dimension attribute columns as follows:
  - **CalendarYear** for **Year**
  - **CalendarSemester** for **Half Year**
  - **CalendarQuarter** for **Quarter**
  - **Month** for **Month**
  - **Simple Date** for **Date**

Results: After this exercise, you should see that the Customer, Product, and Date dimensions have been modified, and time intelligence has been added to the date dimension.

Exercise 2
Defining Relationships and Hierarchies

Scenario

The senior database developer at Adventure Works has received requests from several other departments who have also reported as having issues with browsing the cube, and has asked that you make additional modifications to the database in order to make browsing the cube easier for your users which will, in turn, improve productivity. In this exercise, you will define attribute relationships and create natural and non-natural hierarchies.

The main tasks for this exercise are as follows:

1. Create attributes in the Date dimension.
2. Create a natural hierarchy in the Date dimension.
3. Create hierarchies in the Customer dimension.

▶ Task 1: Create attribute relationships in the Date dimension

- Open the **Date** dimension in Dimension Designer.
- Change the **Name** property of the **Simple Date** attribute to **Date**.
- In the **Attribute Relationships** tab, create the following relationships:

Source Attribute	Related Attribute
Date	Month
Month	Calendar Quarter
Calendar Quarter	Calendar Semester
Calendar Semester	Calendar Year

▶ Task 2: Create a natural hierarchy in the Date dimension

- In the **Dimension Structure** for the **Date** dimension, drag the **Calendar Year** attribute to the **Hierarchies** pane.
- Add the following attributes to the hierarchy in this order: **Calendar Semester**, **Calendar Quarter**, **Month**, and **Date**.
- Change the hierarchy name to **Calendar Date**.

▶ Task 3: Create hierarchies in the Customer dimension

- Open the **Customer** dimension in Dimension Designer.
- Rename the **English Country Region Name** attribute to **Country-Region**.
- Rename the **State Province Name** attribute to **State-Province**.
- Drag the **Country-Region** attribute to the **Hierarchies** pane.
- Add the following attributes to the hierarchy in this order: **State-Province**, **City**, **Postal Code**, and **Full Name**.
- Change the hierarchy name to **Customer Geography**.
- Drag the **Gender** attribute to the **Hierarchies** pane.
- Add the **Marital Status** to the hierarchy.
- Change the name of the hierarchy to **Gender - Marital Status**.

Results: After this exercise, you should have created a Calendar Date hierarchy, a Customer Geography hierarchy, and a Gender - Marital Status hierarchy.

Exercise 3
Sorting and Grouping Dimension Attributes

Scenario

In this exercise, you will modify the sort order of months in the Time dimension. You will also group the members of the Yearly Income attribute hierarchy together into groups.

The main tasks for this exercise are as follows:

1. Modify the sort order of the Month attribute.
2. Group the Yearly Income attribute hierarchy members in the Customer dimension.

▶ Task 1: Modify the sort order of the Month attribute

- Browse the **Calendar Time** hierarchy in the **Time** dimension, and note that the Months level is sorted in alphabetical rather than chronological order.
- Use the **New Attribute from Column** option to add an attribute for the **MonthNumberOfYear** column.
- Set the following property values for the new attribute:
  - **AttributeHierarchyEnabled**: False
  - **AttributeHierarchyOptimizedState**: NotOptimized
  - **AttributeHierarchyOrdered**: False
- Expand the **TimeKey** attribute and notice that all attributes are directly related to the **TimeKey** attribute.
- Create a relationship between the **Month** and the **Month Number of Year** attributes by dragging the **Month Number of Year** attribute to the **<new attribute relationship>** tag under the **Month** attribute.
- Set the **Month** attribute to be ordered by the **Month Number of Year** attribute key.
- Deploy the project and browse the **Calendar Time** hierarchy to verify that the values are now sorted in the correct order.

▶ Task 2: Group the Yearly Income attribute hierarchy members in the Customer dimension

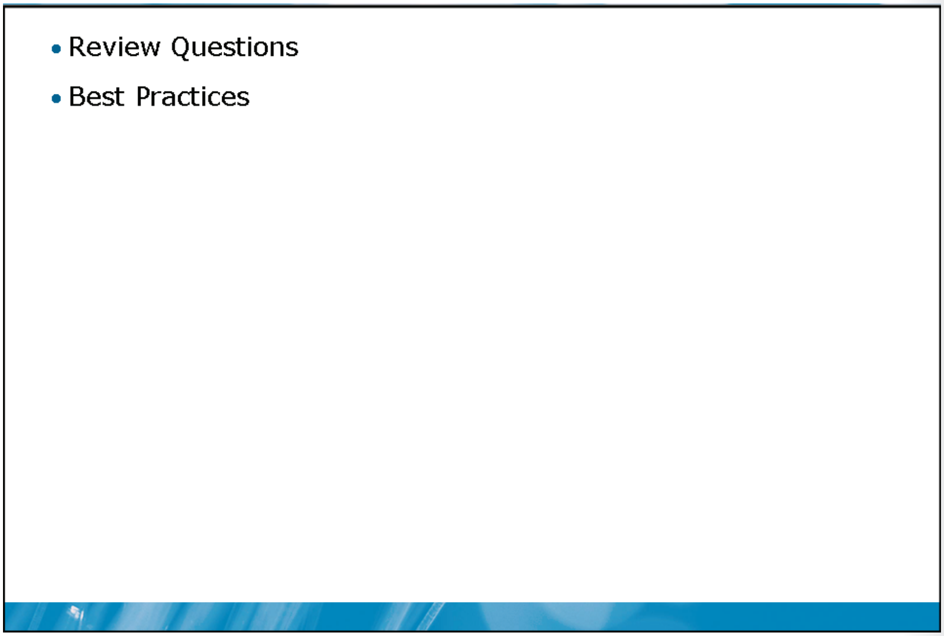
- Organize the **Yearly Income** attribute members into five groups.
- Use the **Automatic DiscretizationMethod** to have Analysis Services distribute the values into discretized groups automatically.
- Use the **DiscretizationBucketCount** property to specify the number of groups to create (in this case, five).
- Browse the **Yearly Income** attribute hierarchy to verify that the values have been discretized.

> **Results**: After this exercise, you should be able to view the correctly sorted and grouped dimensions in the cube.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-01 virtual machine and discard any changes.

Module Review and Takeaways



Review Questions

1. How can you define dimensions, attributes, and hierarchies?
2. To provide real-time OLAP, which dimension storage method would you use?
3. Which dimension storage method provides the fastest query results?
4. What is the difference between a balanced and unbalanced hierarchy?
5. How can you control the sort order of attributes?

Best Practices Related to Dimension Design

Supplement or modify the following best practices for your own work situations:

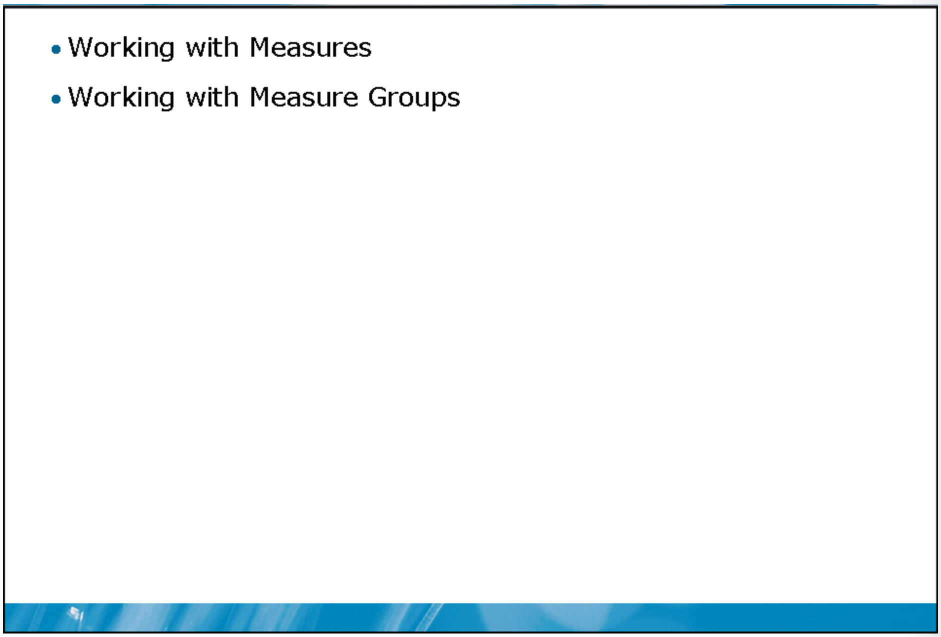
- **Create attribute relationships wherever they exist in the data**. Attribute relationships are an important part of dimension design. They help the server optimize storage of data, define referential integrity rules within the dimension, control the presence of member properties, and determine how MDX restrictions on one hierarchy affect the values in another hierarchy. For these reasons, it is important to spend some time defining attribute relationships that accurately reflect relationships in the data.
- **Avoid creating attributes that will not be used**. Attributes add to the complexity and storage requirements of a dimension, and the number of attributes in a dimension can significantly affect performance. This is especially of attributes which have AttributeHierachyEnabled set to True. Although Analysis Services can support many attributes in a dimension, having more attributes than are actually used decreases performance unnecessarily and can make the end-user experience more difficult.
- **Do not create redundant attribute relationships**. Do not create attribute relationships that are transitively implied by other attribute relationships. The alternative paths created by these redundant attribute relationships can cause problems for the server and are of no benefit to the dimension. For example, if the relationships A->B, B->C, and A->C have been created, A->C is redundant and should be removed.
- **Set dimension and attribute Type properties correctly for Time, Account, and Geography dimensions**. For time dimensions, it is important to set the dimension and attribute types correctly so that time-related MDX functions and the time intelligence of the Business Intelligence Wizard can work correctly. For Account dimensions, it is similarly important to set appropriate account types when using measures with the aggregate function ByAccount. Geography types are not used by the server, but provide information for client applications. A common mistake is to set the Type property on a dimension but not on an attribute, or vice-versa. Another common mistake when configuring time dimensions is to confuse the different time attribute types, such as [Month] and [Month of Year].

Module 4

Working with Measures and Measure Groups

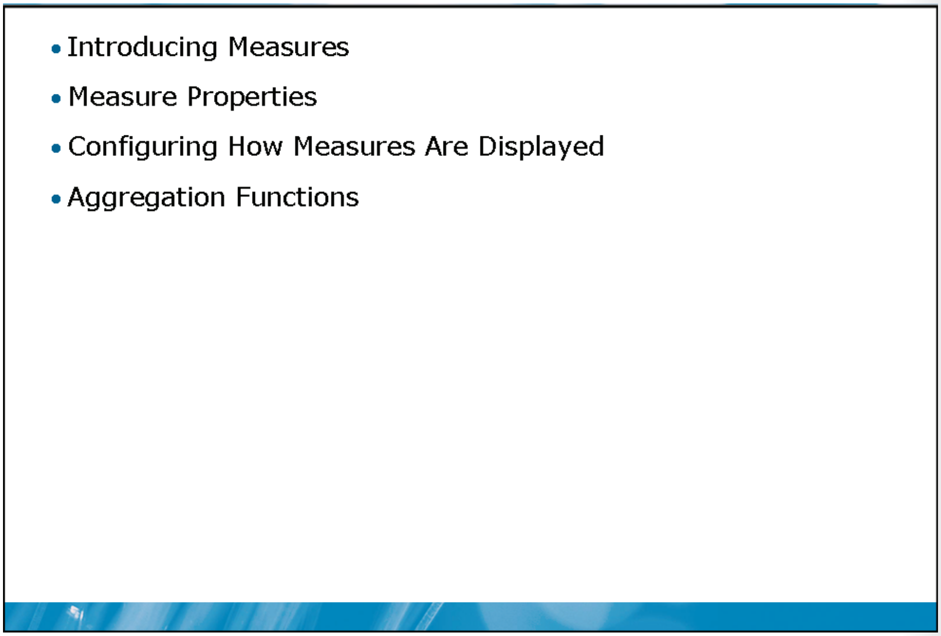
Contents:

Module Overview



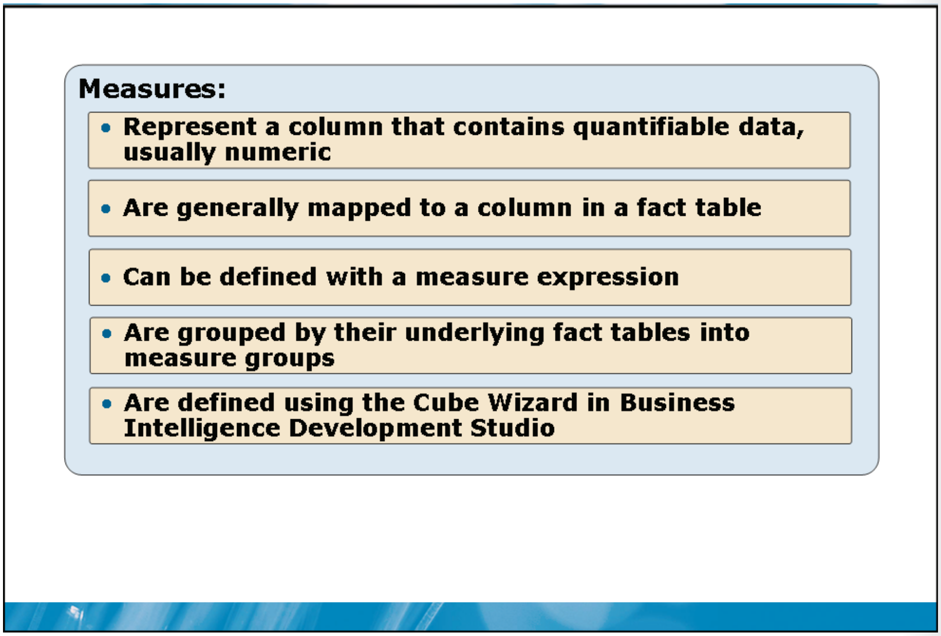
A measure represents a column that contains quantifiable data, usually numeric, that can be aggregated. Measure groups are objects composed of the type of measures in the group, the storage mode used, and information on dimensions and partitions. In this module you will learn about measures and measure groups and how they can be used to define fact tables and associate dimensions with measures.

Lesson 1
Working with Measures



In this lesson you will learn about working with measures including configuring how measures are displayed, information on measure values, and configuring how measures are aggregated.

Introducing Measures

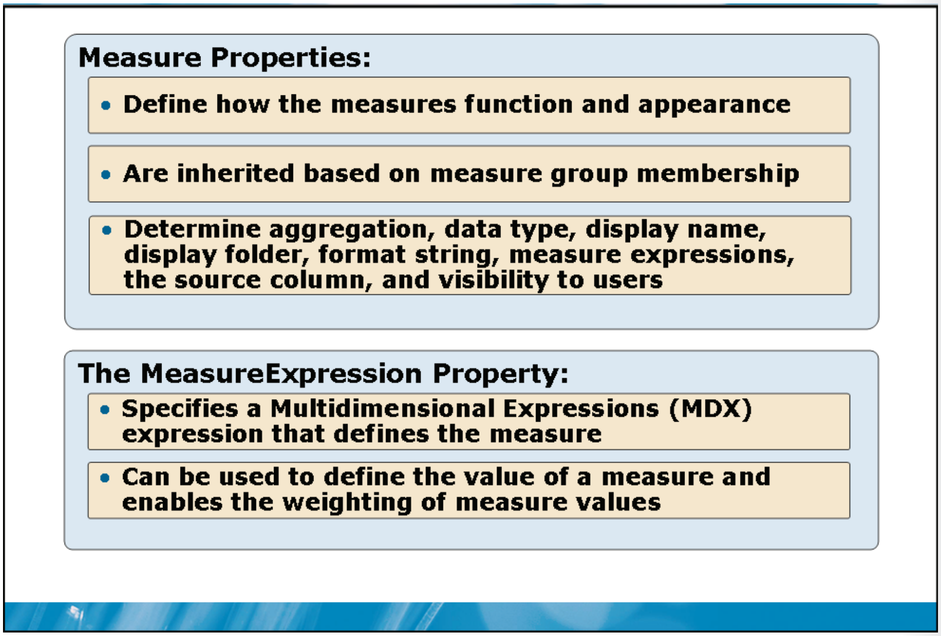


Key Points

A measure represents a column that contains quantifiable data, usually numeric, that can be aggregated.

- You can use *a measure expression* to define the value of a measure, based on a column in a fact table as modified by a Multidimensional Expression.
- Attribute columns from dimension tables can be used to define measures, but such measures are typically semiadditive or nonadditive in terms of their aggregation behavior.
- The aggregation behavior of each measure is determined by the aggregation function associated with the measure.

Measure Properties



Key Points

Measures have properties that enable you to define how the measures function and to control how the measures appear to users.

- Measures inherit certain properties from the measure group of which they are a member, unless those properties are overridden at the measure level.
- Measure properties determine:
  - How measures are aggregated.
  - The data type of the column in the underlying fact table to which the measure is bound.
  - The column in the data source view to which the measure is bound.
  - The description of the measure, which may be exposed in client applications.
  - The name that is displayed to the user.

- The folder in which the measure will appear when users connect to the cube.
- The display format of the measure.
- The unique identifier (ID) of the measure. Note that this property is read-only.
- Any Multidimensional Expressions (MDX) expressions that define the measure.
- The visibility of the measure.

Configuring How Measures Are Displayed

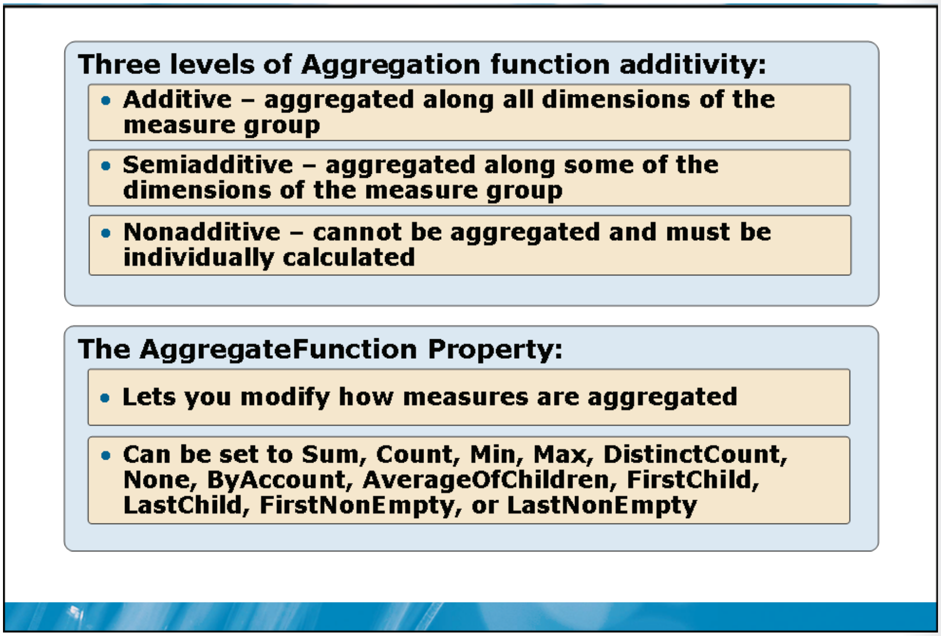
Source data type	Format	Display format value	Example output
Numeric	Named	General Number	123456789
		Fixed	123456789.0
	User-defined	$#,#.00	$123,456,789.00
		#,#0.0000	123,456,789.0000
Date/time	Named	Medium Date	31-Dec-06
		Long Date	Sunday, December 31, 2006
	User-defined	mm/dd/yyyy	12/31/2006
		mmm-dd-yyyy	Dec-31-2006
Boolean	Named	Yes/No	Yes
		True/False	True

Key Points

The measure property FormatString determines how measure values are displayed to users and can be configured using the Properties window of a measure in Cube Designer.

- You can select the format that is used to display measure values to users by using the FormatString property of the measure.
- Although a list of display formats is provided, you can specify many additional formats that are not in the list by specifying any named or user-defined format that is valid in Microsoft® Visual Basic®.
- The table here assumes that the regional setting in Control Panel on the client computer is English (United States).

Aggregation Functions



Key Points

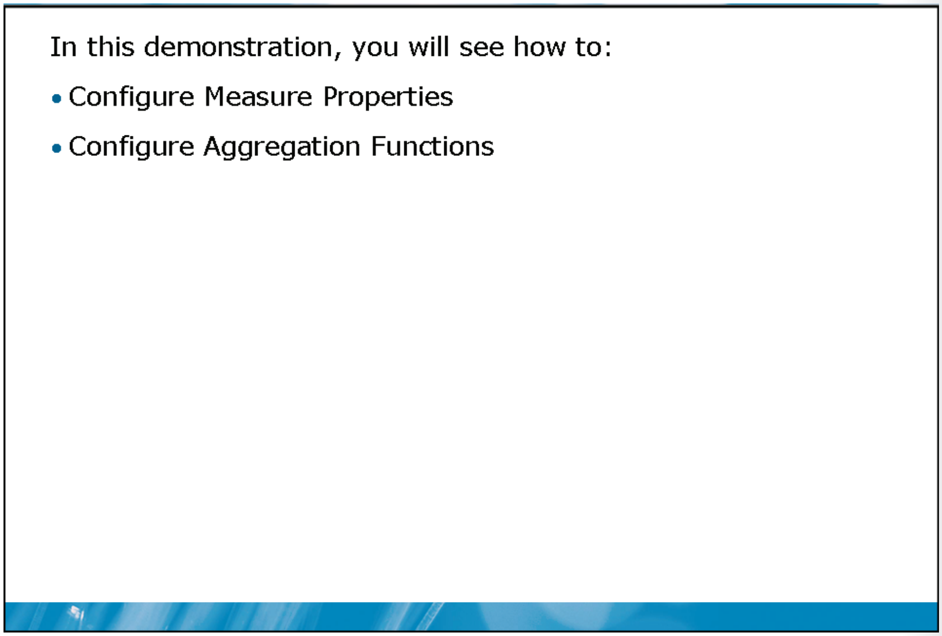
The measure property AggregateFunction determines how measures are aggregated and can be configured using the Properties window of a measure in Cube Designer.

- Aggregation functions have different levels of additivity. Based on the function used the additivity will fall into one of three categories:
  - An additive measure, also called a fully additive measure, can be aggregated along all the dimensions that are included in the measure group that contains the measure, without restriction.
  - A semiadditive measure can be aggregated along some, but not all, dimensions that are included in the measure group that contains the measure.

    - A nonadditive measure cannot be aggregated along any dimension in the measure group that contains the measure. Instead, the measure must be individually calculated for each cell in the cube that represents the measure.

- Use the AggregateFunction property to define the function that is used to aggregate the measure. Commonly used aggregation functions include:
    - The Sum function is additive and calculates the sum of all values for every child member. Sum is the default function.
    - The Count function is additive and calculates the quantity of child members.
    - The Min function is semi-additive and calculates the lowest value for each child member.
    - The Max function is semi-additive and calculates the highest value for each child member.
    - The DistinctCount function is nonadditive and calculates the count of all unique child members.
    - The None function is nonadditive and performs no aggregation and supplies values directly from the fact table.
- Other semiadditive aggregation functions include:
    - The ByAccount function calculates the aggregation according to the aggregation function assigned to the account type for a member in an account dimension.
    - The AverageOfChildren function calculates the average of values for all non-empty child members.
    - The FirstChild function retrieves the value of the first child member.
    - The LastChild function retrieves the value of the last child member.
    - The FirstNonEmpty function retrieves the value of the first non-empty child member.
    - The LastNonEmpty function retrieves the value of the last non-empty child member.

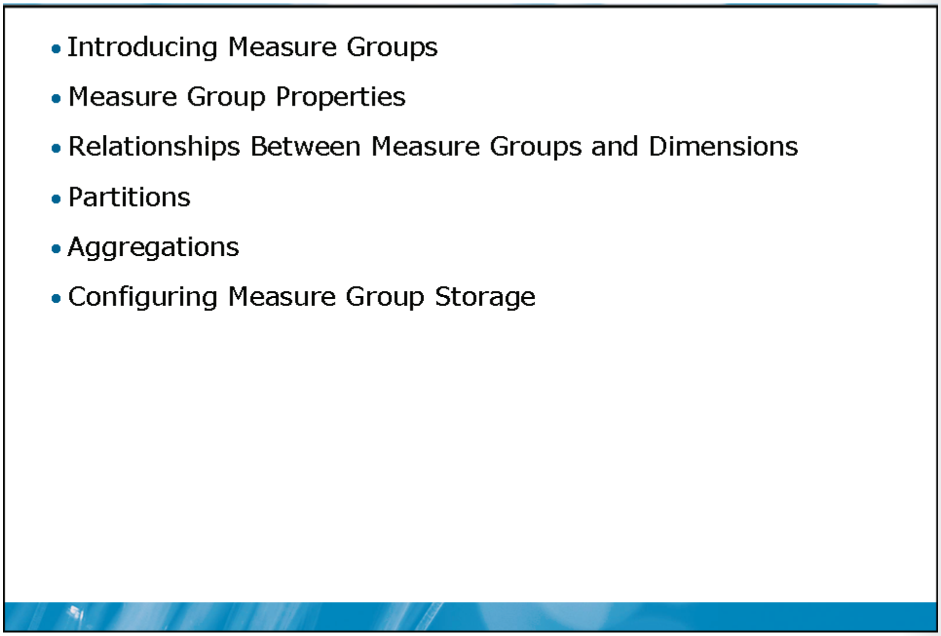
Demonstration: Using Measures



Question: When would you use the DisplayFolder property?

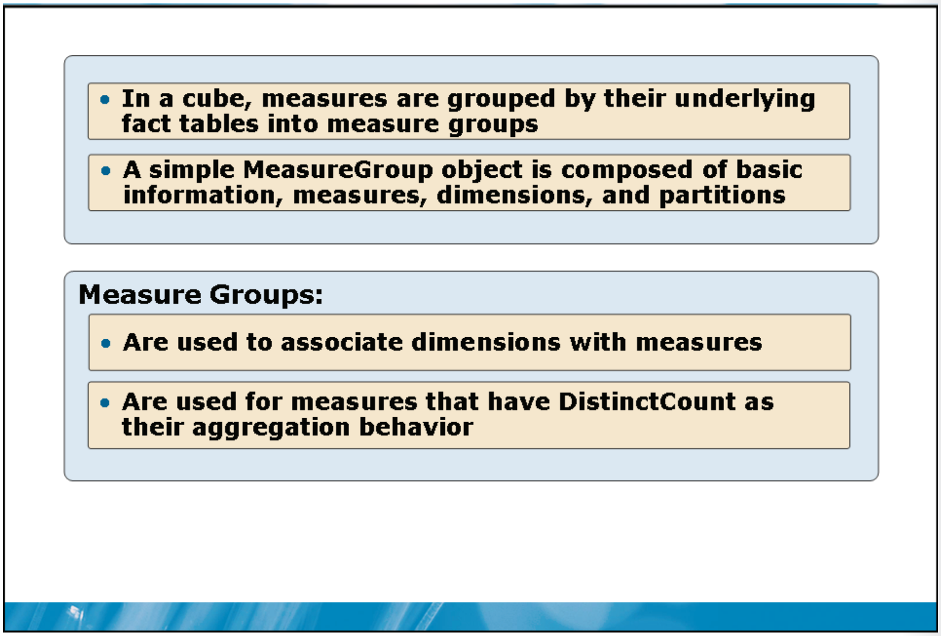
Question: What happens if no value can be read from the fact table for a member that has an AggregateFunction set to None?

Lesson 2
Working with Measure Groups



Measure Groups are used to associate dimensions with measures, Partitions are containers for a portion of the measure group data, and Aggregations are precalculated summaries of data from leaf cells that improve query response times. In this lesson you will learn about measure group properties, relationships between measure groups and dimensions, partitions, aggregations, and configuring Measure Group storage.

Introducing Measure Groups



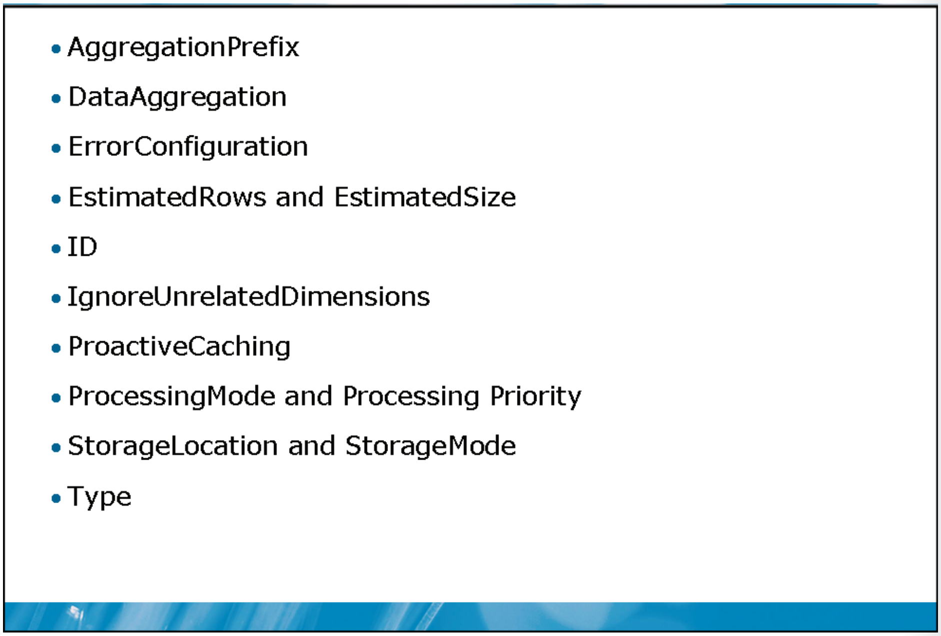
Key Points

In a cube, measures are grouped by their underlying fact tables into measure groups.

- Measure groups are used to associate dimensions with measures.
- Measure groups are also used for measures that have distinct count as their aggregation behavior.

- A simple MeasureGroup object is composed of:
  - Basic information that includes the name of the measure group, the type of measures, the storage mode, and the processing mode.
  - The actual set of measures that compose the measure group.
  - A subset of cube dimensions that will be used to create the processed measure group.
  - The collection of physical splits of the processed measure group.
- You define one or more measure groups in Business Intelligence Development Studio by using the Cube Wizard and then add and configure measure groups by using Cube Designer.

Measure Group Properties



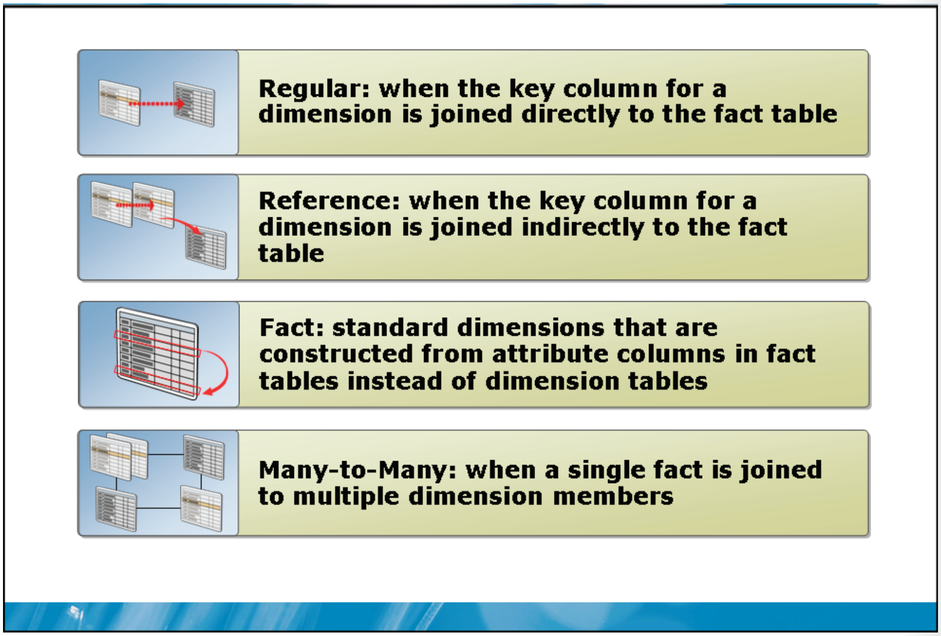
Key Points

Measure group properties determine behaviors for the entire measure group and set the default behaviors for certain properties of measures within a measure group.

- **AggregationPrefix** specifies the common prefix that is used for aggregation names.
- **DataAggregation** determines whether Analysis Services can aggregate persisted data or cached data for the measure group.
- **ErrorConfiguration** provides configurable error handling settings for handling of duplicate keys, unknown keys, null keys, error limits, action upon error detection, and the error log file.
- **EstimatedRows** specifies the estimated number of rows in the fact table.
- **EstimatedSize** specifies the estimated size (in bytes) of the measure group.
- **ID** specifies the identifier of the object.

- **IgnoreUnrelatedDimensions** determines whether unrelated dimensions are forced to their top level when members of dimensions that are unrelated to the measure group are included in a query.
- **ProactiveCaching** provides configurable error handling settings for handling of duplicate keys, unknown keys, null keys, error limits, action upon error detection, and the error log file.
- **ProcessingMode** indicates whether indexing and aggregating should occur during or after processing.
- **ProcessingPriority** determines the processing priority of the cube during background operations, such as lazy aggregations and indexing.
- **StorageLocation** is the file system storage location for the measure group. If none is specified, the location is inherited from the cube that contains the measure group.
- **StorageMode** determines the storage mode for the measure group.
- **Type** specifies the type of the measure group.

Relationships Between Measure Groups and Dimensions

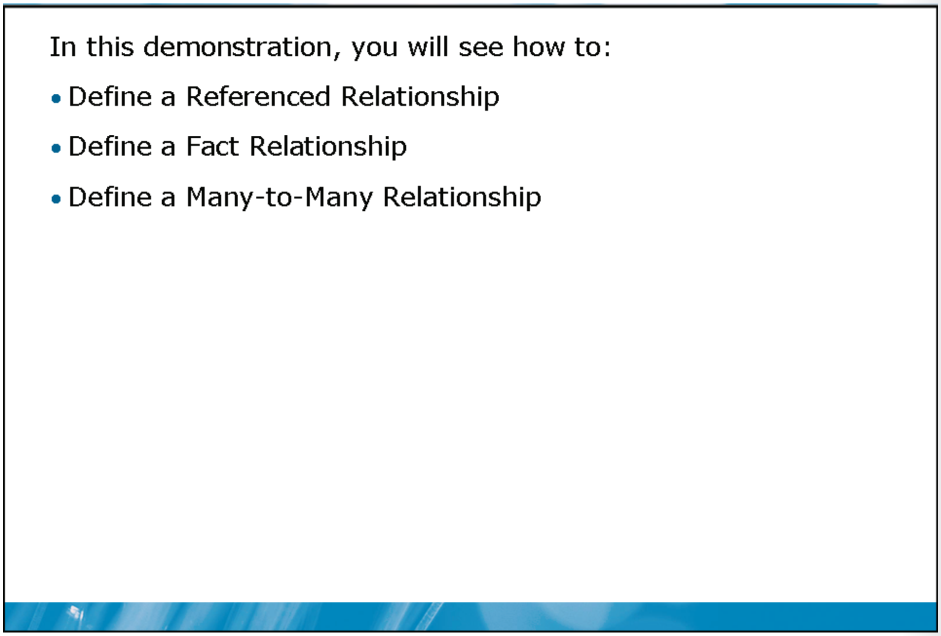


Key Points

A relationship between a dimension and a measure group consists of the dimension and fact tables participating in the relationship and a granularity attribute that specifies the granularity of the dimension in the particular measure group.

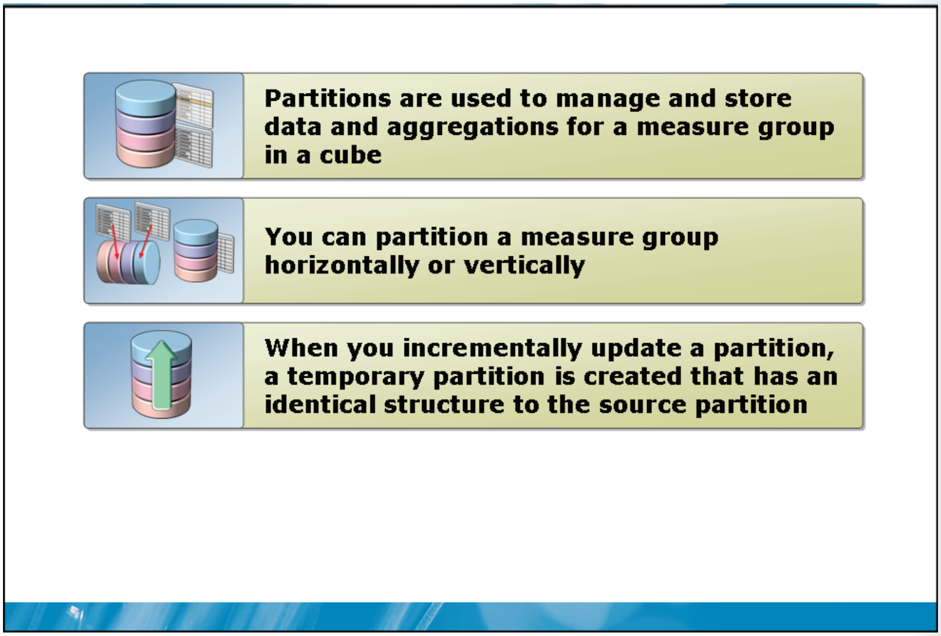
- A regular dimension relationship represents the relationship between dimension tables and a fact table in a traditional star schema design.
- A reference dimension relationship represents the relationship between dimension tables and a fact table in a snowflake schema design.
- Useful dimensional data is sometimes stored in a fact table to reduce duplication.
- It is frequently useful to join a single fact to multiple dimension members as a many-to-many relationship.

Demonstration: Defining Relationships Between Dimensions and Measure Groups



Question: When would it be useful to join a single fact to multiple dimension members?

Partitions



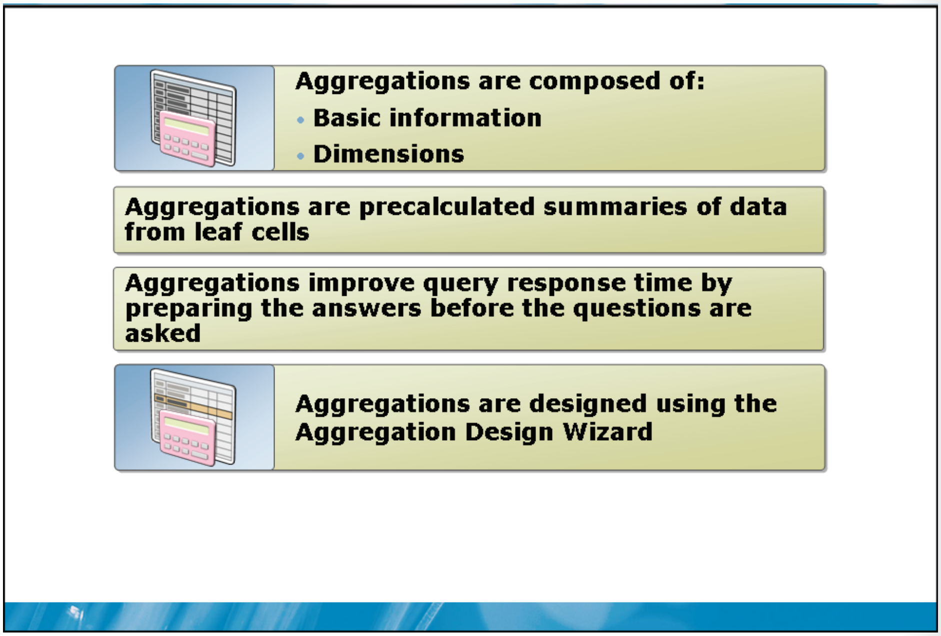
Key Points

A partition is a container for a portion of the measure group data.

- Partitions are used by Microsoft SQL Server Analysis Services to manage and store data and aggregations for a measure group in a cube.
- Partitions allow the source data and aggregate data of a cube to be distributed across multiple hard drives and among multiple server computers.
- Each partition in a measure group can have a different fact table, and these fact tables can be from different data sources.

- A simple Partition object is composed of:
  - Basic information that includes the name of the partition, the storage mode, and the processing mode.
  - A slicing definition that is an MDX expression specifying a tuple or a set.
  - An aggregation design that is a collection of aggregation definitions which can be shared across multiple partitions.
- Typically, you can partition a measure group horizontally or vertically.
  - In a horizontally partitioned measure group, each partition in a measure group is based on a separate table.
  - In a vertically partitioned measure group, a measure group is based on a single table, and each partition is based on a source system query that filters the data for the partition.

Aggregations



Key Points

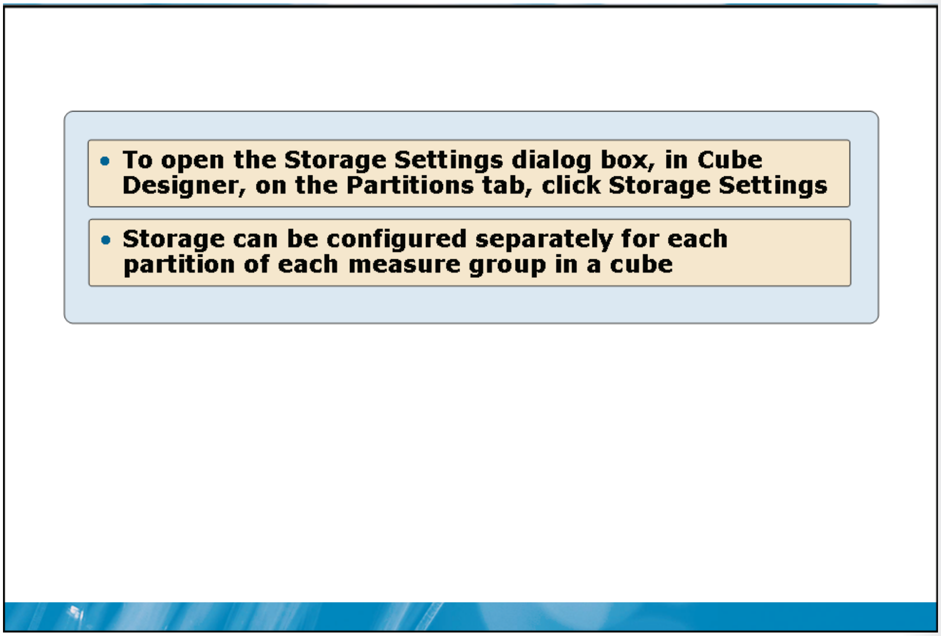
Aggregations are precalculated summaries of data from leaf cells that improve query response time by preparing the answers before the questions are asked.

- A simple Aggregation object is composed of:
  - Basic information that includes the name of the aggregation, the ID, annotations, and a description.
  - A collection of AggregationDimension objects that contain the list of granularity attributes for the dimension.

- The Aggregation Design Wizard provides options for you to specify storage and percentage constraints on the algorithm to achieve a satisfactory tradeoff between query response time and storage requirements.
- Microsoft SQL Server Analysis Services incorporates a sophisticated algorithm to select aggregations for precalculation so that other aggregations can be quickly computed from the precalculated values.

Question: When are aggregations calculated?

Configuring Measure Group Storage



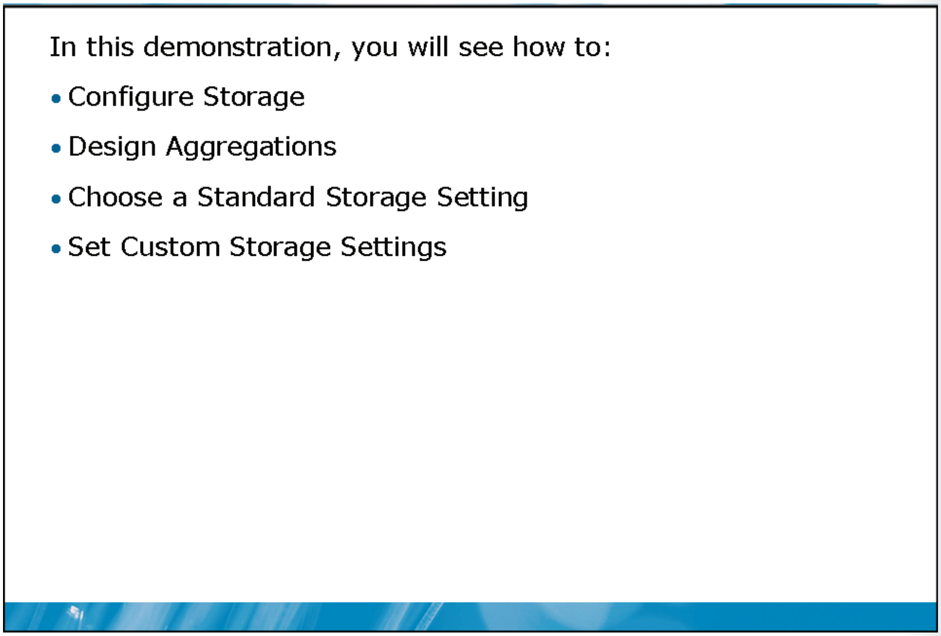
Key Points

You can use the Storage Settings dialog box to configure storage.

- Storage settings for partitions are set in the Storage Options dialog box, which can be opened in one of several different ways.
- Configuring Default Storage Settings for new Measure Groups Added to a Cube:
  - On the **Cube Builder** tab in the Cube Wizard, click the cube object in either the **Measures** or **Dimensions** pane, and then in the **Properties** window, click the browse **(...)** button for the **ProactiveCaching** property setting
- Configuring Default Storage Settings for new Partitions Added to a Measure Group:
  - On the **Partitions** tab in the Cube Wizard, expand the measure group and click the **Storage Settings** link for that measure group. This displays the **Storage Settings** dialog box for the selected measure group.

- Configuring Storage for an Existing Partition:
  - On the **Partitions** tab in the Cube Wizard, expand the measure group, and use one of the following methods:
    - Right-click the partition, and then click **Storage Settings**.
    - Click the partition, and then on the **Cube** menu, click **Storage Settings**.
    - Click the partition, and then on the toolbar, click the **Set Proactive-cache Settings** buttons.
    - Click the partition, and then in the **Properties** window, click the browse **(...)** button for the **ProactiveCaching** property setting.

Demonstration: Defining Partition Storage and Aggregations



Question: What are the different settings you can configure with the Storage Settings dialog box in Cube Designers?

Question: What goal should be kept in mind when designing aggregations with the Aggregation Design Wizard?

Lab: Configuring Measures and Measure Groups

- Exercise 1: Configuring Measures
- Exercise 2: Defining Dimension Usage and Relationships
- Exercise 3: Configuring Measure Group Storage

Logon information

Virtual machine	NY-SQL-01
User name	Administrator
Password	Pa$$w0rd

Estimated time: 60 minutes

Exercise 1
Configuring Measures

Scenario

You are refining the OLAP cube for your company. You want to modify the formatting for some measures in your cube. You have found that several relationships are needed for users to be able to retrieve the data that they need. You will add the relationships that are required to support these users. You also want to configure measure group storage for when the OLAP solution is deployed to testing.

In this exercise you will configure format strings and display folders for measures.

The main tasks for this exercise are as follows:

1. Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator.
2. Create a new SQL Server user.
3. Configure measure display format using the FormatString property.
4. Configure measure display name using the Name property.
5. Deploy the Analysis Services project and review formatting changes.

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

- Start 6234A-NY-SQL-01 and logon as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

- Use **SQL Server Management Studio** to create a new user on the **AdventureWorksDW2008** database.
- Name the user **sqlserver**, set the default schema to **dbo**, grant the user database role membership in **db_owner** and **db_securityadmin**.

▶ Task 3: Configure measure display format using the FormatString property

- Use the **Adventure Works** cube contained in project **Mod04Lab**.
- Change the properties of **Unit Price**, **Extended Amount**, **Discount Amount**, **Product Standard Cost**, **Total Product Cost**, **Sales Amount**, **Tax Amt**, and **Freight** measures to display them as **Currency**.
- Set the **FormatString** property for **Unit Price Discount Pct** to **Percent**.

▶ Task 4: Configure measure display name using the Name property

- Rename the **Unit Price Discount Pct** measure to **Unit Price Discount Percent**.
- Rename the **Tax Amt** measure to **Tax Amount**.

▶ Task 5: Deploy the Analysis Services project and review formatting changes

- Deploy **Module04Lab**.
- Review the measures changed in task 3 of this exercise by adding them to the data area to ensure your changes have been made.

Results: After this exercise, you should have configured format strings and display names for several measures. You should have also deployed and reviewed the Analysis Services project.

Exercise 2
Defining Dimension Usage and Relationships

Scenario

Now that you have configured the measure properties for your company's OLAP cube, you will need to add the relationships required to support the company's users.

In this exercise you will create a referenced relationship and a fact relationship.

The main tasks for this exercise are as follows:

1. Define a Referenced Relationship.
2. Define a Regular Relationship.

▶ Task 1: Define a Referenced Relationship

- Define the intermediate dimension attribute and the referenced dimension relationship.
- Dimension **Reseller Sales** by **Geography**.

▶ Task 2: Define a Regular Relationship

- Define the **Internet Sales Order Details** dimension.
- Define a Regular relationship for the **Internet Sales Order Details** dimension using **Order Number** as the Granularity attribute.

Results: After this exercise, you should have defined dimension usage and relationships for measure groups by defining a referenced relationship and defining a regular relationship.

Exercise 3
Configuring Measure Group Storage

Scenario

With the measure display configured and relationships created, it is now time to configure measure group storage for when the OLAP solution is deployed.

In this exercise you will configure the storage mode of a measure group and design aggregations for a measure group.

The main tasks for this exercise are as follows:

1. Configure the storage mode of the Internet Sales measure group.
2. Design aggregations for the Internet Sales measure group.

▶ Task 1: Configure the storage mode of the Internet Sales measure group

- Configure default storage settings for new measure groups.
- Configure storage for the existing **Internet Sales** measure group.

▶ Task 2: Design aggregations for the Internet Sales measure group

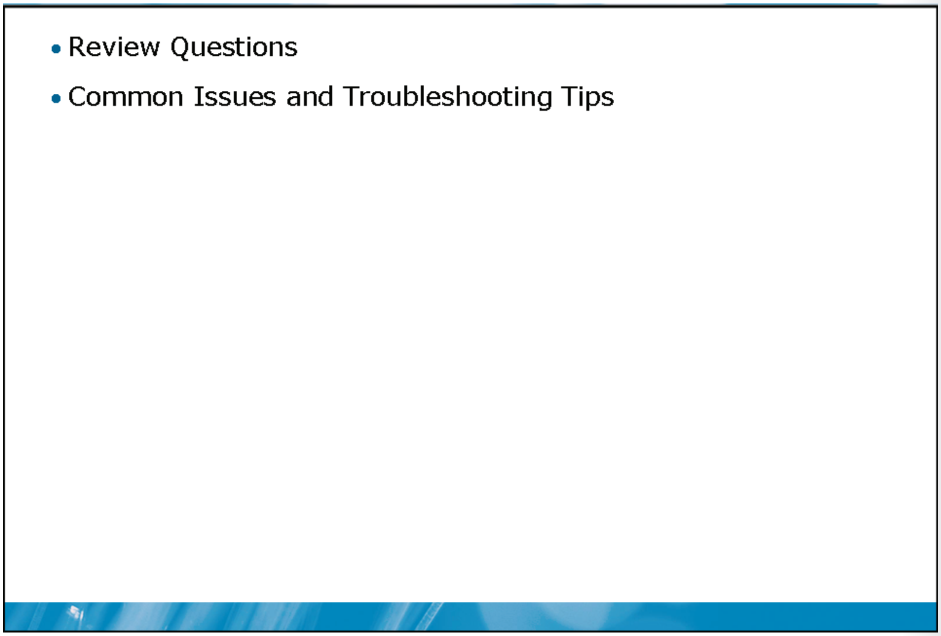
- Use the **Aggregation Design Wizard** to specify a performance gain limit for aggregations.
- Review the aggregations designed with the wizard.

Results: After this exercise, you should have defined aggregations and storage modes for the Internet Sales Measure group.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-01 virtual machine and discard any changes.

Module Review and Takeaways



Review Questions

1. What is a MeasureGroup object composed of?
2. What are the two ways to partition a measure group?
3. How can aggregations be used to improve query response time

Common Issues Related to Defining Relationships

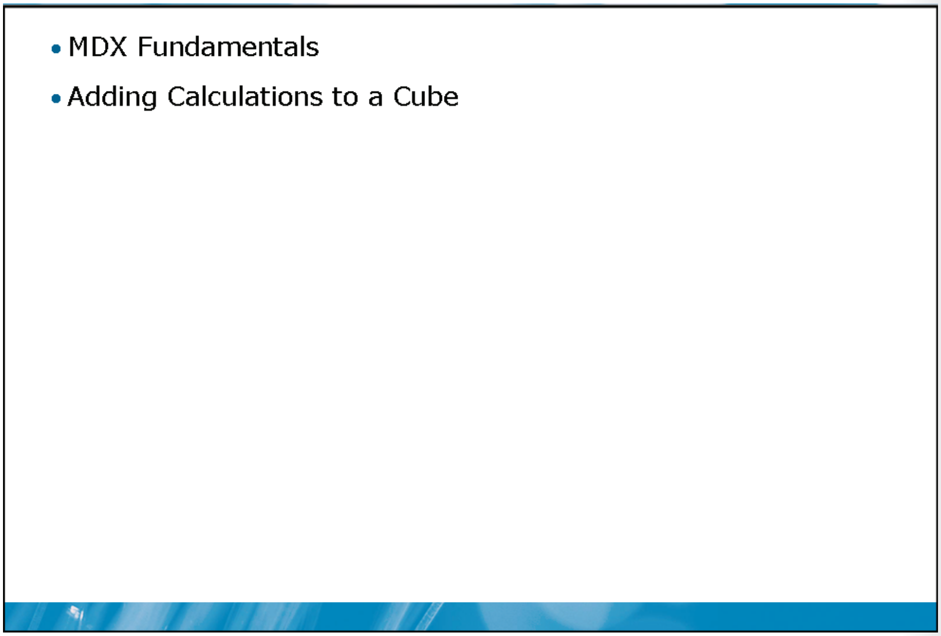
Issue	Troubleshooting tip
Dimension relationship information is unavailable after the measure group has been updated.	
User cannot select the correct intermediate measure group when trying to create a many-to-many dimension relationship.	

Module 5

Querying Multidimensional Analysis Solutions

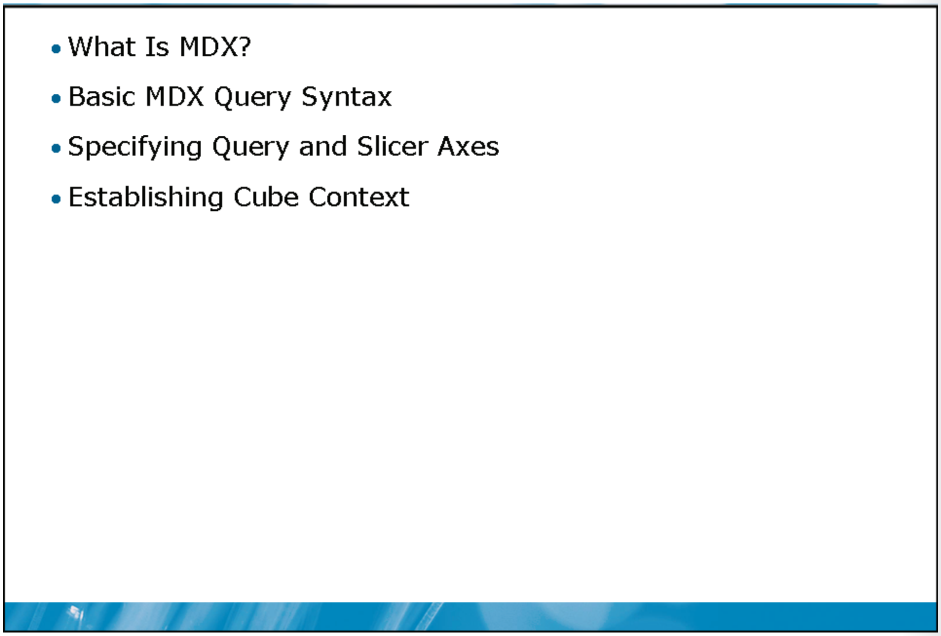
Contents:

Module Overview



Multidimensional Expressions (MDX) is the query language that you use to work with and retrieve multidimensional data in Microsoft® SQL Server® 2008 Analysis Services (SSAS). In this module, you will learn the fundamentals of Multidimensional Expressions. You will also learn about building calculations, such as calculated members and named sets, which are both

Lesson 1
MDX Fundamentals



You can use MDX to query multidimensional data or to create MDX expressions for use within a cube, but first you need to understand the fundamentals of MDX. In this lesson you will learn about Multidimensional Expressions. You will learn what MDX is, discuss basic MDX query syntax and how to specify query and slicer axes, and finally discuss the process involved in establishing cube context.

What Is MDX?

- Use Multidimensional Expressions (MDX) to query multidimensional data or create MDX expressions
- A *cell* in a cube is the space that exists at the intersection of a member of the measures dimension member and a member from each attribute hierarchy in a cube
- A *tuple* uniquely identifies a cell, based on a combination of attribute members
- A *set* is an ordered set of tuples with the same dimensionality

```
SELECT
   {
      ([Measures].[Reseller Sales Amount],
         [Date].[Calendar Year].[CY 2003]),
      ([Measures].[Reseller Sales Amount],
         [Date].[Calendar Year].[CY 2004])
   } ON COLUMNS
FROM [Adventure Works]
```

Key Points

MDX queries and expressions can be used to return data to a client application from a SQL Server Analysis Services cube, format query results, perform cube design tasks such as the definition of calculated members and named sets, and also perform administrative tasks including dimension and cell security.

- MDX utilizes expressions composed of identifiers, values, statements, functions, and operators that SSAS can evaluate to retrieve an object or scalar value.
- MDX is similar in many ways to the SQL syntax typically used with relational databases, however, MDX is not an extension of the SQL language.

Basic MDX Query Syntax

SELECT Statement Syntax

```
[ WITH <SELECT WITH clause> [ , <SELECT WITH clause> ... ] ]
SELECT [ * | ( <SELECT query axis clause>
    [ , <SELECT query axis clause> ... ] ) ]
FROM <SELECT subcube clause>
[ <SELECT slicer axis clause> ]
[ <SELECT cell property list clause> ]
```

SELECT Statement Example

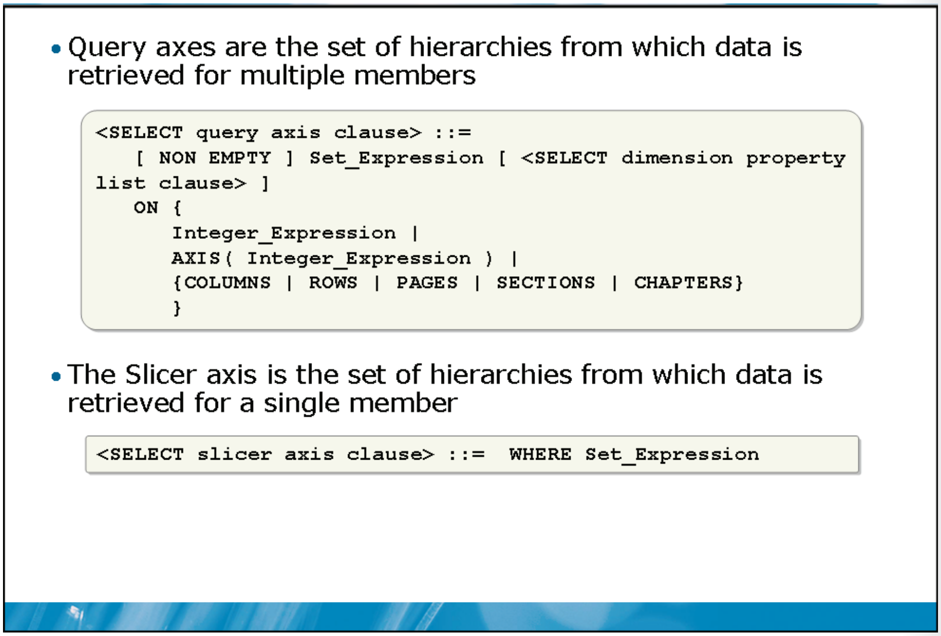
```
SELECT
    { [Measures].[Sales Amount],
        [Measures].[Tax Amount] } ON COLUMNS,
    { [Date].[Fiscal].[Fiscal Year].&[2002],
        [Date].[Fiscal].[Fiscal Year].&[2003] } ON ROWS
FROM [Adventure Works]
WHERE ( [Sales Territory].[Southwest] )
```

Key Points

The basic MDX query is the SELECT statement—the most frequently used query in MDX.

- In MDX, the SELECT statement specifies a result set that contains a subset of multidimensional data that has been returned from a cube.
- To identify the query axes, the cube that sets the context of the query, and the slicer axis, the MDX SELECT statement uses the following clauses:
  - A SELECT clause that determines the query axes of an MDX SELECT statement.
  - A FROM clause that determines which multidimensional data source to use when extracting data to populate the result set of the MDX SELECT statement.
  - A WHERE clause that optionally determines which dimension or member to use as the slicer axis that restricts the extracting of data to a specific dimension or member.

Specifying Query and Slicer Axes



Key Points

Query axes and slicer axes are each a set of hierarchies from which data is retrieved.

- Query axes specify the edges of a cellset returned by a Multidimensional Expressions (MDX) SELECT statement.
- To explicitly specify a query axis, use the <SELECT query axis clause> as shown in the example here.
- The slicer axis filters the data returned by the MDX SELECT statement, restricting the returned data so that only data intersecting with the specified numbers will be returned.
- To explicitly specify a slicer axis, you using the <SELECT slicer axis clause> in MDX.
- If a member from a hierarchy within the cube is not explicitly included in a query axis, the default member from that hierarchy is implicitly included in the slicer axis.

Establishing Cube Context

- FROM Clause Syntax

```
<SELECT subcube clause> ::=
   Cube_Identifier |
   (SELECT [
      * |
      ( <SELECT query axis clause> [ , <SELECT query axis
clause> ... ] ) ]
   FROM <SELECT subcube clause> <SELECT slicer axis clause> )
```

- Refining the Context

LookupCube

```
Numeric expression syntax
LookupCube(Cube_Name, Numeric_Expression)

String expression syntax
LookupCube(Cube_Name, String_Expression)
```

Filter

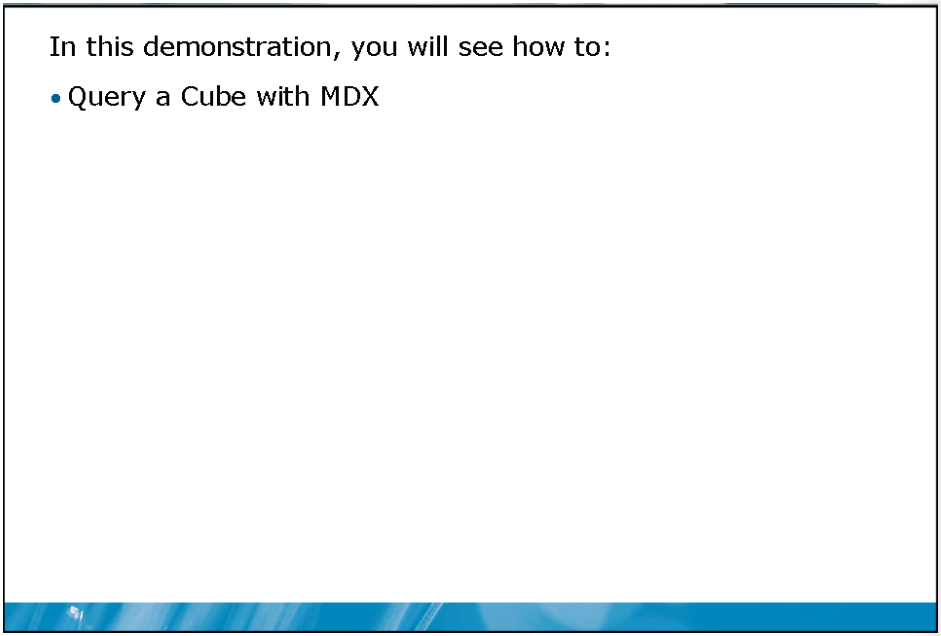
```
Filter(Set_Expression, Logical_Expression)
```

Key Points

Every MDX query runs within a specified cube context. This context defines the members that are evaluated by the expressions within the query.

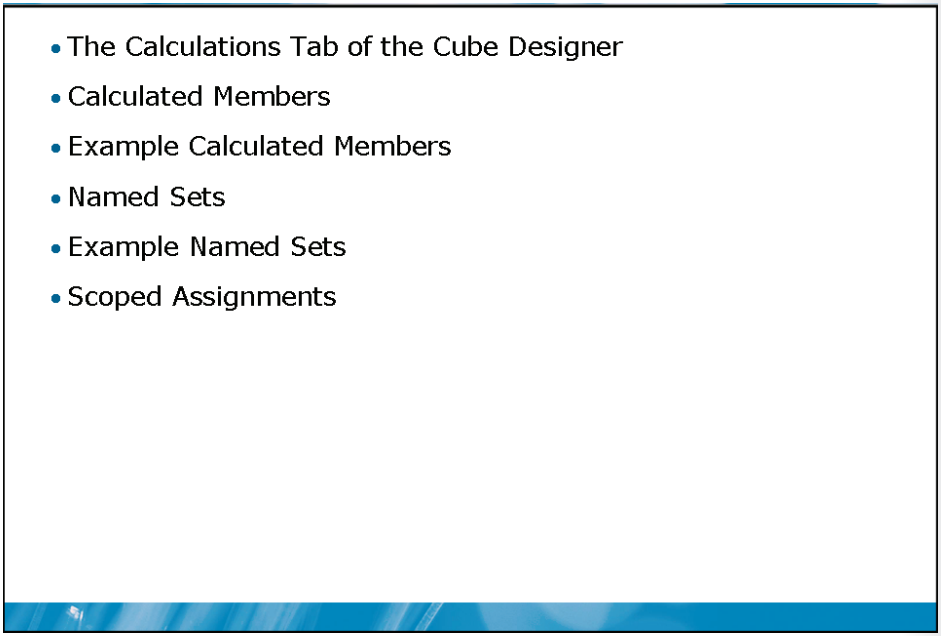
- In the SELECT statement, the FROM clause determines the cube context.
- Although the FROM clause specifies the cube context as within a single cube, this does not have to limit us from working with data from more than one cube at a time.

Demonstration: Querying a Cube with MDX



Question: Why would you use a tuple in a query?

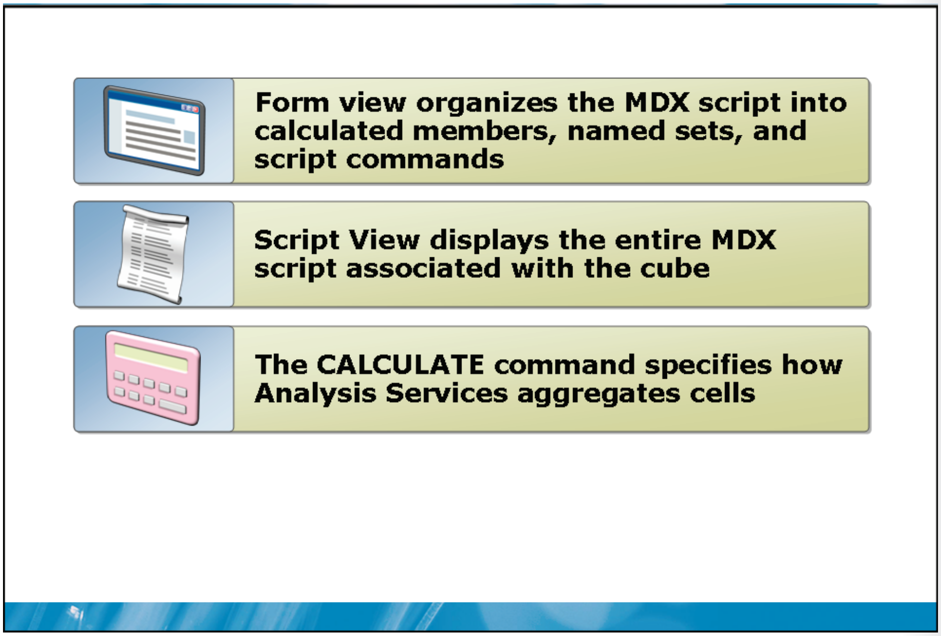
Lesson 2
Adding Calculations to a Cube



In SQL Server Analysis Services, calculations include calculated members and named sets. Calculated members and named sets are combinations of cube data, arithmetic operators, numbers, and functions. In Multidimensional Expressions (MDX), a calculated member is a member that is resolved by calculating an MDX expression to return a value, while a named set is a set of dimension members or expressions that are grouped together for reuse in MDX queries. The ability to construct and use calculated members and named sets in an MDX query provides a great deal of manipulation capability for multidimensional data.

In this lesson, you will learn about adding calculations to a cube. You will review the Calculations Tab of the Cube Designer and introduce the concepts of calculated members and named sets. Finally, you will review some examples calculated members and named sets.

The Calculations Tab of the Cube Designer

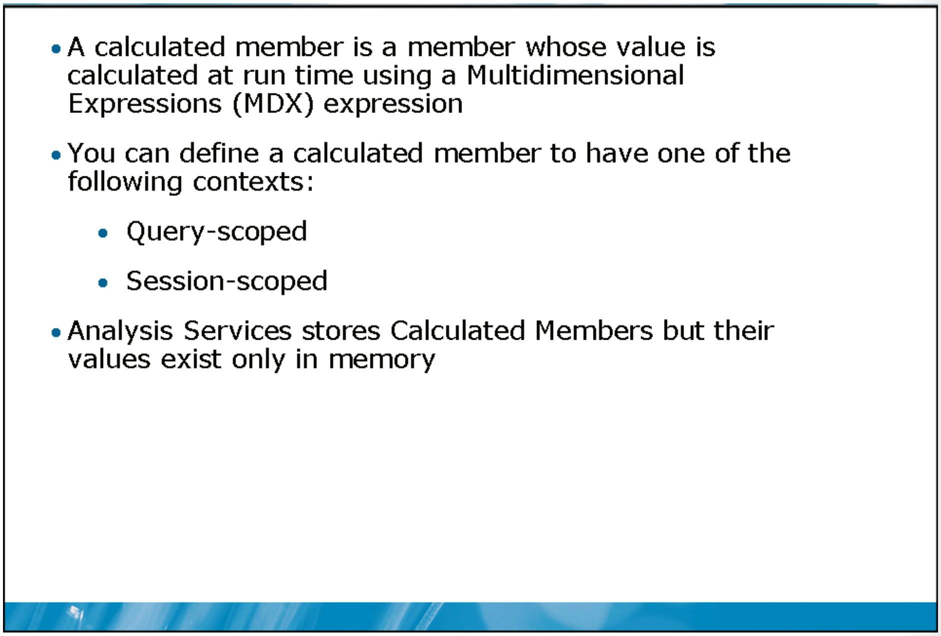


Key Points

To create a calculated member, use the New Calculated Member icon on the toolbar of the Calculations Tab of Cube Designer. This command displays a form to specify several options for the calculated member.

- Use the Script Organizer pane in form view to display the contents of the cube script in an ordered format.
- Use the Calculation Tools pane in both form view and script view to display metadata, functions, and tools available to the cube.
- Use the Script Editor pane in script view to edit the entire cube script and in form view to edit script commands contained in the cube script.
- Use the Calculated Member Form Editor pane in form view to edit calculated members in the cube script.
- Use the Named Set Form Editor pane in form view to edit named sets in the cube script.

Calculated Members



Key Points

A calculated member is a member whose value is calculated at run time using a Multidimensional Expressions (MDX) expression that is specified when you define the calculated member. A calculated member is available to business intelligence applications just like any other member.

- Calculated members do not increase the size of the cube because only the definitions are stored in the cube; values are calculated in memory as required to answer a query.
- Although calculated members are typically based on data that already exists in the cube, you can create complex expressions by combining data with arithmetic operators, numbers, and functions.

- You can define a calculated member to have one of two contexts:
  - Query-scoped: a calculated member that is defined as a part of an MDX query and whose scope is limited to that query.
  - Session-scoped: a calculated member whose scope is wider than the context of the query and whose scope is the lifetime of the MDX session.

Example Calculated Members

- Example Query-Scoped Calculated Member

```
WITH
   MEMBER [Measures].[Special Discount] AS
   [Measures].[Discount Amount] * 1.5
SELECT
   [Measures].[Special Discount] on COLUMNS,
   NON EMPTY [Product].[Product].MEMBERS  ON Rows
FROM [Adventure Works]
WHERE [Product].[Category].[Bikes]
```

- Example Session-Scoped Calculated Member

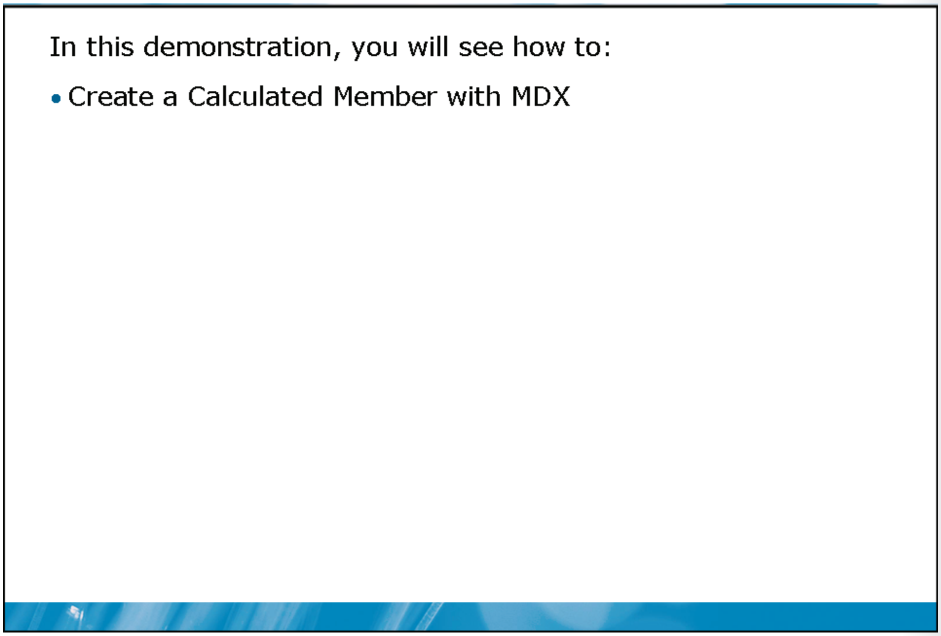
```
Create Session Member [Store].[Measures].LastFourStores as
sum(([Stores].[ByLocation].Lag(3) :
[Stores].[ByLocation].NextMember), [Measures].[Units Sold])
```

Key Points

Here are two example calculated members:

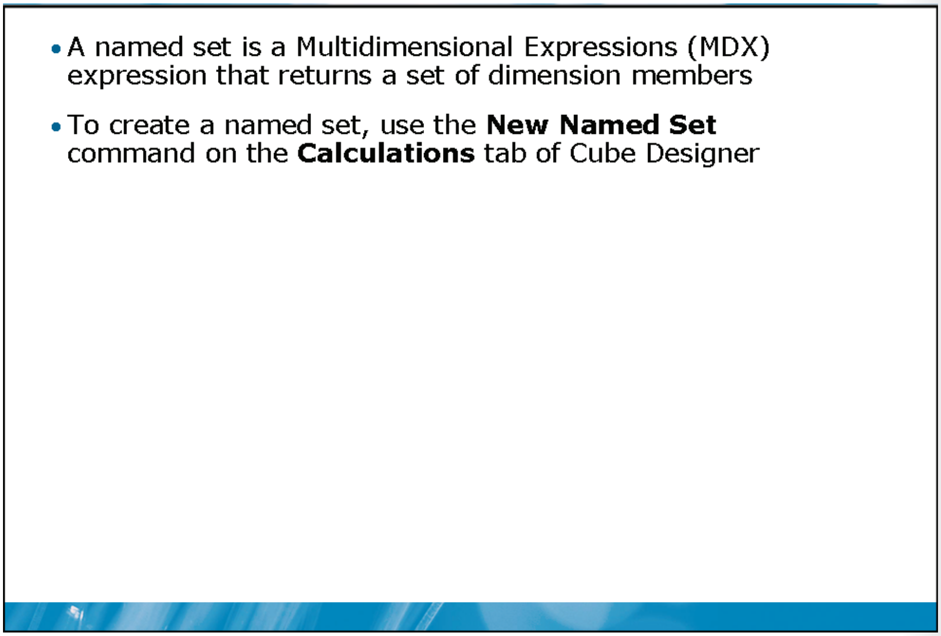
- If a calculated member is only required for a single Multidimensional Expressions (MDX) query, you can define that calculated member by using the WITH keyword.
- To create a calculated member that is available throughout a Multidimensional Expressions (MDX) session, use the CREATE MEMBER statement.

Demonstration: Using Calculations in a Cube



Question: What can calculated members be based on?

Named Sets



Key Points

A named set is a Multidimensional Expressions (MDX) expression that returns a set of dimension members. You can define named sets and save them as part of the cube definition; you can also create named sets in client applications.

- Named sets can be used by users in MDX queries in client applications and can also be used to define sets in subcubes.
- Named sets simplify MDX queries and provide useful aliases for complex, typically used, set expressions.

Question: How are named set definitions stored?

Example Named Sets

- Example Query-Scoped Named Set

```
WITH SET [ChardonnayChablis] AS
   'Filter([Product].Members, (InStr(1,
[Product].CurrentMember.Name, "chardonnay") <> 0) OR
(InStr(1, [Product].CurrentMember.Name, "chablis") <> 0))'
SELECT
   [ChardonnayChablis] ON COLUMNS,
   {Measures.[Unit Sales]} ON ROWS
FROM Sales
```

- Example Session-Scoped Named Set

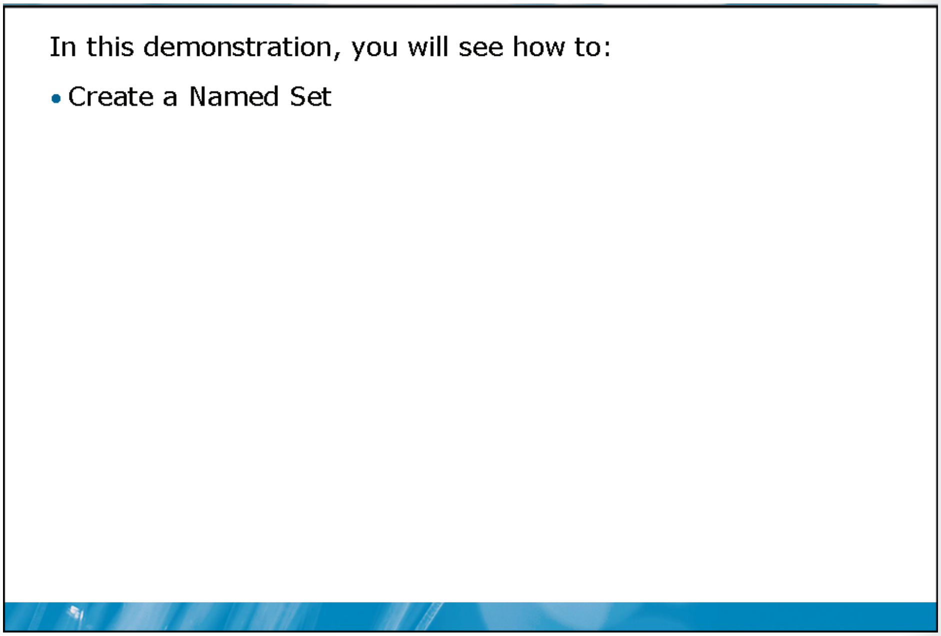
```
create Session set [Store].[SetCities_2_3] as
{[Data Stores].[ByLocation].[State].&[CA].&[City 02],
[Data Stores].[ByLocation].[State].&[NH].&[City 03]}
```

Key Points

Here are two example named sets.

- If a named set is only required for a single Multidimensional Expressions (MDX) query, you can define that named set by using the WITH keyword.
- To create a named set that is available throughout a Multidimensional Expressions (MDX) session, use the CREATE SET statement. A named set that is created by using the CREATE SET statement will not be removed until after the MDX session closes.

Demonstration: Using Named Sets in a Cube



Question: How can you make MDX queries easier to maintain with named sets?

Scoped Assignments

```
Scope
 (  [Date].[Fiscal Year].&[2005],
    [Date].[Fiscal].[Fiscal Quarter].Members,
    [Measures].[Sales Amount Quota] ) ;
    This = ParallelPeriod
          (
             [Date].[Fiscal].[Fiscal Year], 1,
             [Date].[Fiscal].CurrentMember
          ) * 1.35 ;

/*-- Allocate equally to months in FY 2002 ------------------
-----------*/

  Scope
  (  [Date].[Fiscal Year].&[2002],
     [Date].[Fiscal].[Month].Members   ) ;
    This = [Date].[Fiscal].CurrentMember.Parent / 3 ;
  End Scope ;
End Scope ;
```

Key Points

The SCOPE statement limits the scope of specified Multidimensional Expressions (MDX) statements to a specified subcube.

- A scoped assignment uses the SCOPE command to define a subcube. MDX statements can then be applied to this subcube including the CALCULATE statement. You can use the THIS function to refer to the subcube.
- SCOPE statements will create subcubes that expose "holes" regardless of the MDX Compatibility setting. For example, the statement, `Scope( Customer.State.members )`, can include the states in countries or regions that do not contain states, but for which otherwise invisible placeholder members were inserted.

Lab: Querying a Cube

- Exercise 1: Querying a Cube by Using MDX
- Exercise 2: Creating a Calculated Member
- Exercise 3: Defining a Named Set

Logon information

Virtual machine	NY-SQL-01
User name	Administrator
Password	Pa$$w0rd

Estimated time: 60 minutes

Exercise 1: Querying a Cube by Using MDX

Scenario

You are the database administrator for the Adventure Works company. The sales management team has made several requests for information that you must fulfill. The senior database administrator has asked you to fulfill these requests using Multidimensional Expressions (MDX) as a test of your ability.

In this exercise you will write multiple MDX queries to obtain the necessary information for the sales management team.

The main tasks for this exercise are as follows:

1. Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator.
2. Create a new SQL Server user.
3. Write MDX queries to show Internet Sales amounts and Reseller Sales amounts.
4. Write an MDX query to show Reseller Sales amounts by Product Line.

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

- Start 6234A-NY-SQL-01 and logon as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

- Use SQL Server Management Studio to create a new user on the AdventureWorksDW2008 database.
- Name the user **sqlserver**, set the default schema to **dbo**, grant the user database role membership in **db_owner** and **db_securityadmin**.

▶ Task 3: Write MDX queries to show Internet Sales amounts and Reseller Sales amounts

- Use the Adventure Works cube contained in project **Mod05Lab**.
- To retrieve the fully qualified key of an object, drag the object from the Metadata tab to the query pane.
- Write one MDX query that returns Internet Sales amounts.
- Write another MDX query that returns Reseller Sales amounts.

▶ Task 4: Write an MDX query to show Internet Sales amounts across Customer geography

- Continue using the Adventure Works cube contained in project **Module05Lab**.
- Write an MDX query that returns internet sales amounts by customer geography.

Results: After this exercise, you should have written an MDX query to show Internet Sales amounts, written an MDX query to show Reseller Sales amounts, and written an MDX query to show Internet Sales amounts by across Customer geography.

Exercise 2: Creating a Calculated Member

Scenario

Now that you have written several MDX queries to obtain information for the sales management team, the senior database administrator has requested additional information of you, and has asked you to create calculated members in order to better support the needs of the sales management team.

In this exercise you will create a calculated member to aggregate sales information and two additional calculated members that return gross profit margins.

The main tasks for this exercise are as follows:

1. Define calculations to aggregate physical measures.
2. Define gross profit margin calculations.
3. Browse the new calculated members.

▶ Task 1: Define calculations to aggregate physical measures

- Use the Adventure Works cube contained in project **Module05Lab**.
- Create a calculated member named **[Total Sales Amount]** that aggregates internet **[Sales Amount]** and **[Sales Amount - Fact Reseller Sales]**.
- Define the **Non-empty behavior** option to improve performance.

▶ Task 2: Define gross profit margin calculations

- Create a new calculated member named **[Internet GPM]** that first subtracts internet **[Sales Amount]** from internet **[Total Product Cost]** and then divides by internet **[Sales Amount]** to find a percentage.
- Create a new calculated member named **[Reseller GPM]** that also finds a percentage as above but uses the **[Sales Amount - Fact Reseller Sales]** and **[Total Product Cost - Fact Reseller Sales]** measures.

▶ Task 3: Browse the new calculated members

- Deploy the updated cube.
- Add the **Total Sales Amount**, **Internet GPM**, and **Reseller GPM** measures to the **Drop Totals of Detail Fields Here** area, and then review the results.

Results: After this exercise, you should have created several calculated members and then browsed the calculated members to review your work.

Exercise 3: Defining a Named Set

Scenario

The sales management team is now requesting a final set of information which require the writing of a named set to obtain. The senior database administrator has asked you to write this named set and also to demonstrate to the sales management team how to browse the cube using this new named set.

In this exercise you will create a named set to obtain information vital to the sales management team.

The main tasks for this exercise are as follows:

1. Define a Core Products named set.
2. Browse the cube using the new named set.

▶ Task 1: Define a Core Products named set

- Create a new named set named **[Core Products]**.
- Add **Bikes** from **English Product Category Name** in the **Dim Product Category** dimension to the **[Core Products]** named set.

▶ Task 2: Browse the cube using the new named set

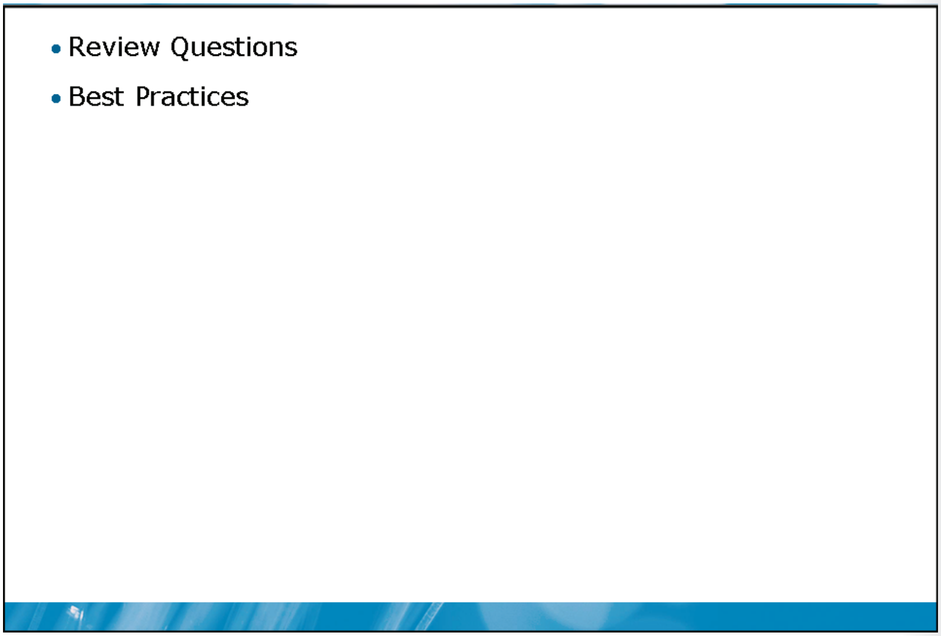
- Deploy the updated cube.
- Add **Sales Amount - Fact Reseller Sales** measure to the data area and then add the **Product Categories** user-defined hierarchy to the row area.
- Add the **Core Products** named set to the subcube area to review how it affects the cube data.

> **Results**: After this exercise, you should have defined a named set and browsed the cube using the new named set.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-01 virtual machine and discard any changes.

Module Review and Takeaways



Review Questions

1. What are MDX queries and expressions in SQL Server used for?
2. What are the three panes of the Calculations tab and what purpose does each pane serve?
3. How can named sets be created?

Best Practices Related to Specifying Query and Slicer Axis

Supplement or modify the following best practices for your own work situations:

- **Define a default member before working with slicer axes**. The default member of an attribute hierarchy is used to evaluate expressions when an attribute hierarchy is not included in a query. To avoid name resolution problems, define the default member in the cube's MDX script in the following situations: if the cube refers to a database dimension more than once, if the dimension in the cube has a different name than the dimension in the database, or if you want to have different default members in different cubes.
- **An MDX query cannot skip query axes**. That is, a query that includes one or more query axes must not exclude lower-numbered or intermediate axes. For example, a query cannot have a ROWS axis without a COLUMNS axis, or have COLUMNS and PAGES axes without a ROWS axis.

Best Practices Related to Defining Calculations

Supplement or modify the following best practices for your own work situations:

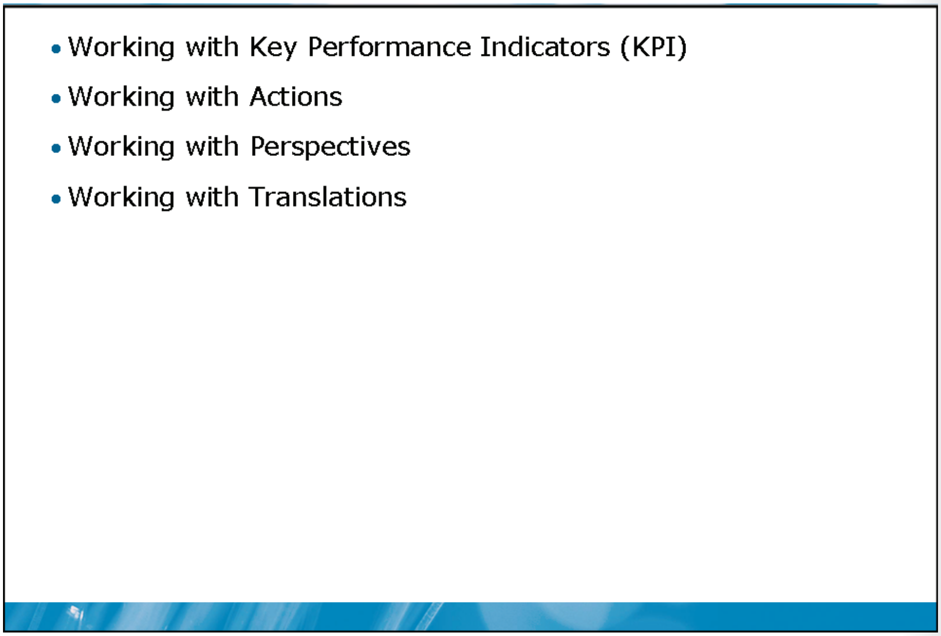
- **Calculations are solved in the order listed in the Script Organizer pane**. You can reorder the calculations by right-clicking any calculation and then clicking Move Up or Move Down on the shortcut menu. You can also click a calculation and then click the Move Up or Move Down button on the toolbar.
- **For best performance with cell calculations, specify only a single member when possible**. The Non-empty Behavior option stores the names of measures used to resolve NON EMPTY queries in MDX. If this property is blank, the calculated member must be evaluated repeatedly to determine whether a member is empty. If this property contains the name of one or more measures, the calculated member is treated as empty if all the specified measures are empty. This property is an optimization hint to Analysis Services to return only non-NULL records. Returning only non-NULL records improves the performance of MDX queries that utilize the NON EMPTY operator or the NonEmpty function, or that require the calculation of cell values.

Module 6

Customizing Cube Functionality

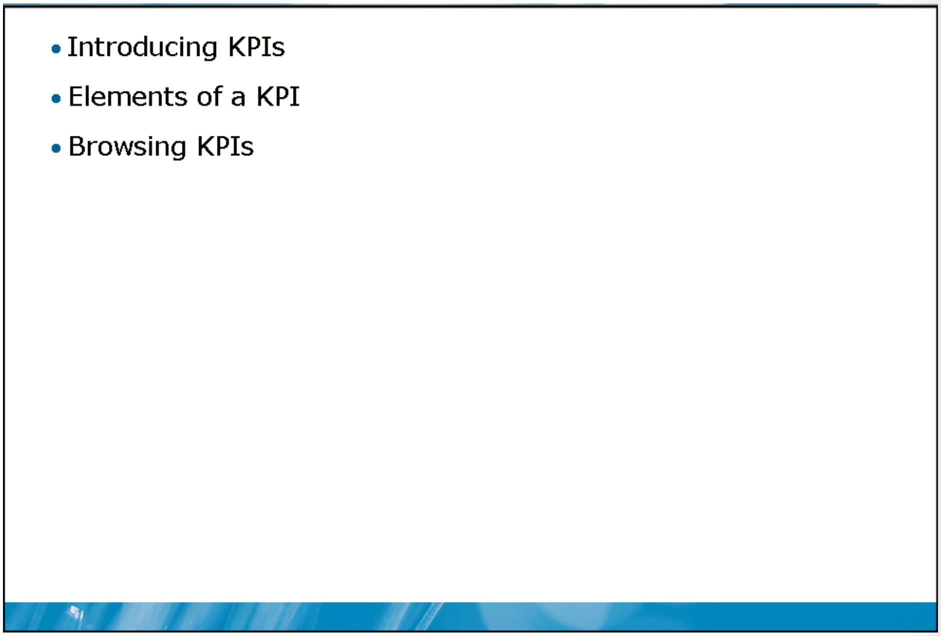
Contents:

Module Overview



In this module you will learn how to customize cube functionality by using several technologies available to you in Microsoft® SQL Server® 2008 Analysis Services. You will learn about Key Performance Indicators and how they can be used to obtain a quick and accurate historical summary of business success. You will learn about Actions and how they allow end-users to go beyond traditional analysis to initiate solutions to discovered problems and deficiencies. You will learn about Perspectives and how they allow users to see the cube in a simpler way, enabling users to focus on the most relevant data to them. Finally you will learn about Translations, and how they can be used to translate various elements of a cube so that users from all over the world can view and understand cube and dimension data.

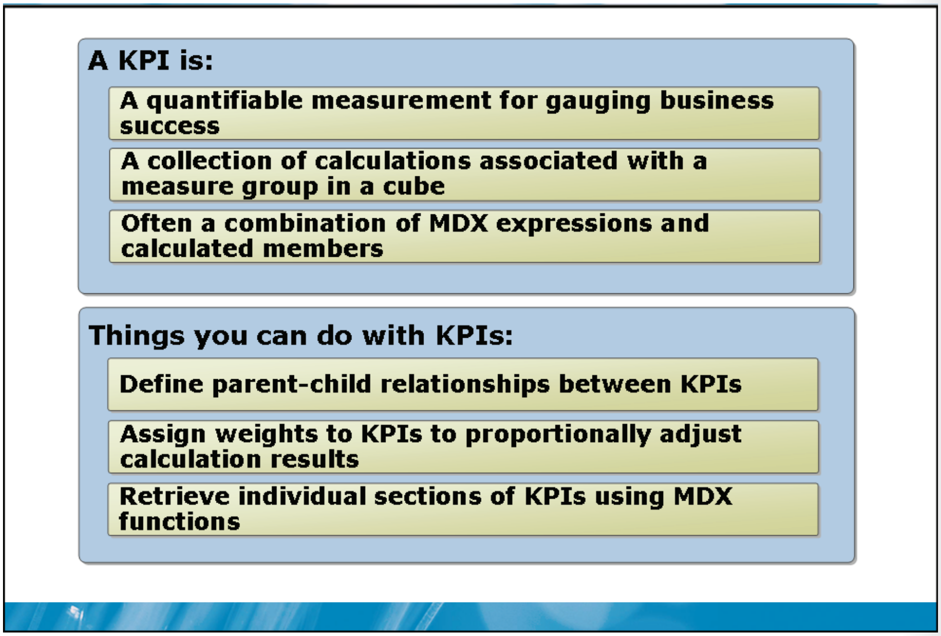
Lesson 1
Working with Key Performance Indicators (KPI)



In business terminology, a Key Performance Indicator (KPI) is a quantifiable measurement for gauging business success. A KPI is frequently evaluated over time. For example, the sales department of an organization may use monthly gross profit as a KPI, but the human resources department of the same organization may use quarterly employee turnover. Each is an example of a KPI. Business executives frequently consume KPIs that are grouped together in a business scorecard to obtain a quick and accurate historical summary of business success.

In this lesson you will learn what KPIs are, the elements of a KPI, and how to browse KPIs. You will also learn about implementing key performance indicators (KPIs).

Introducing KPIs



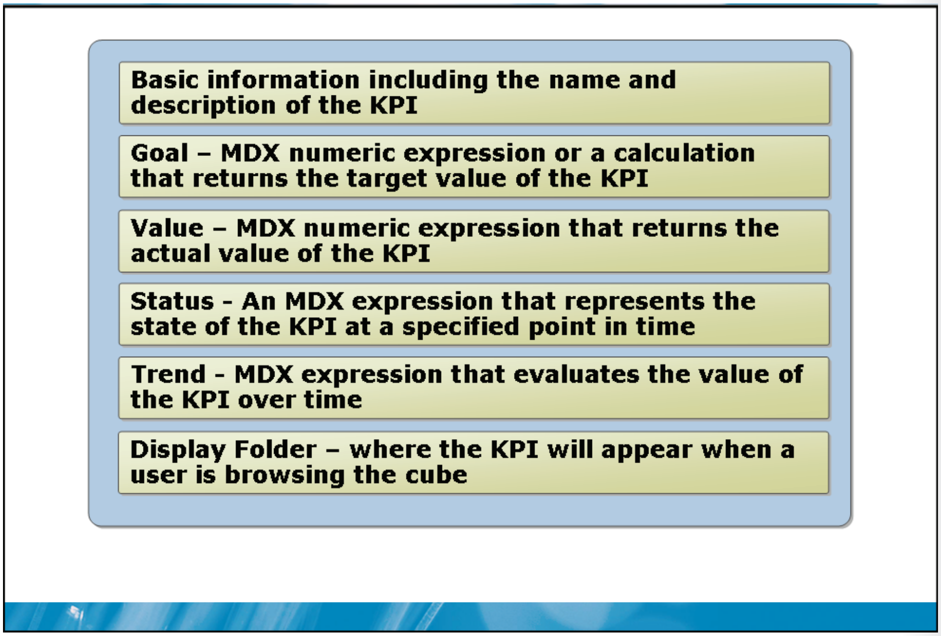
Key Points

In Microsoft SQL Server Analysis Services, a KPI is a collection of calculations, which are associated with a measure group in a cube, that are used to evaluate business success.

- A Key Performance Indicator (KPI) is a quantifiable measurement for gauging business success.
- A KPI handles information about a goal set, the actual formula of the performance recorded in the cube, and measurement to show the trend and the status of the performance.
- One key advantage of KPIs in Analysis Services is that they are server-based KPIs that are consumable by different client applications.

- A server-based KPI presents a single version of the truth, compared to separate versions of the truth from separate client applications.
- Analysis Services lets you define a parent-child relationship between KPIs.
- You can use MDX functions to retrieve individual sections of the KPI, such as the value or goal, for use in MDX expressions, statements, and scripts.

Elements of a KPI



Key Points

A simple KPI object is composted of basic information, the goal, the actual value achieved, a status value, a trend value, and a folder where the KPI is viewed.

- A value expression is a physical measure such as Sales, a calculated measure such as Profit, or a calculation that is defined within the KPI by using a Multidimensional Expressions (MDX) expression.
- A goal expression is a value, or an MDX expression that resolves to a value, that defines the target for the measure that the value expression defines. For example, the following Goal Expression has a value of 60 for Accessories and 30 for all other categories:

```
Case
    When [Product].[Category].CurrentMember Is
      [Product].[Category].[Accessories]
    Then .60
    Else .30
End
```

- A status expression is an MDX expression that Analysis Services uses to evaluate the current status of the value expression compared to the goal expression, which a generally a normalized value in the range of -1 to +1. -1 is very bad, and +1 is very good. The status expression displays with a graphic to help you easily determine the status of the value expression compared to the goal expression. For example, the following code returns a value of 1 if you achieve the goal and a value of -1 if you do not:

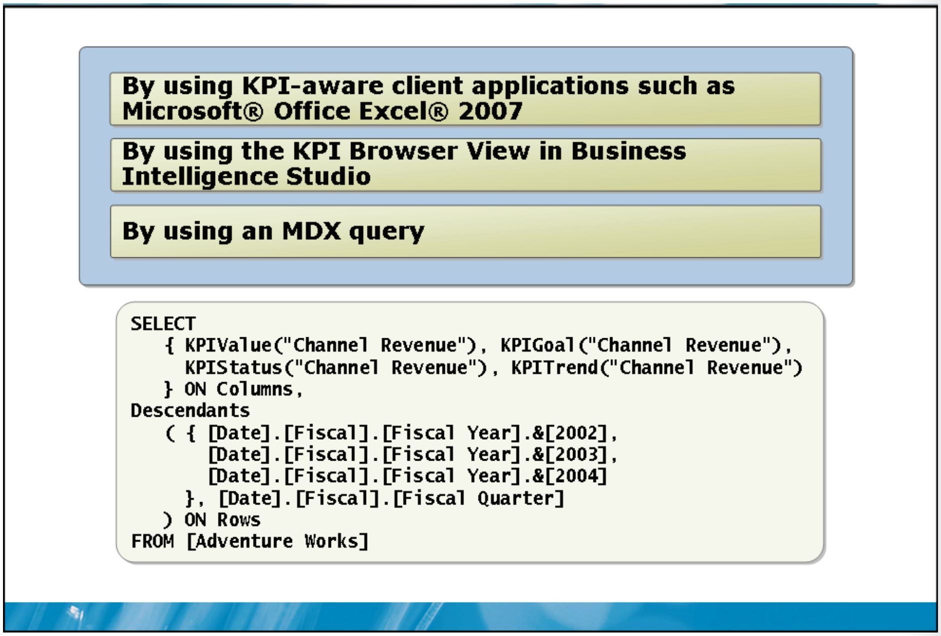
```
Case
    When KpiValue( "Gross Profit Margin" ) /
         KpiGoal ( "Gross Profit Margin" ) >= 1
    Then 1
    Else -1
End
```

- A trend expression is an MDX expression that Analysis Services uses to evaluate the current trend of the value expression compared to the goal expression. The trend expression helps the business user to quickly determine whether the value expression is getting better or worse relative to the goal expression. You can associate one of several graphics with the trend expression to help business users be able to quickly understand the trend. For example, the following code returns a value of 1 if the Gross Profit Margin is increasing over the last year, -1 if it is decreasing, and 0 if it is the same:

```
Case
    When KpiValue( "Gross Profit Margin" ) >
         (KpiValue( "Gross Profit Margin" ),
          ParallelPeriod
          ( [Date].[Fiscal Time].[Fiscal Year], 1,
            [Date].[Fiscal Time].CurrentMember ))
    Then 1
    When KpiValue( "Gross Profit Margin" ) <
         (KpiValue( "Gross Profit Margin" ),
          ParallelPeriod
          ( [Date].[Fiscal Time].[Fiscal Year], 1,
            [Date].[Fiscal Time].CurrentMember ))
    Then -1
    Else 0
End
```

- Additional Properties include:
  - **Display Folder**. This is the folder in which the KPI will appear if you are browsing the cube.
  - **Parent KPI**. This defines the parent of the current KPI. This causes the browser to display the KPI as a child and also allows the parent to access the values of the child. This allows you to have KPIs based on other KPIs. You can also apply a Weight to adjust the importance of a child KPI against its siblings.
- **Current Time Member**. This is an MDX expression that defines the current time member.

Browsing KPIs



Key Points

A key advantage of KPIs in Analysis Services is that they are server-based KPIs that are consumable by different client applications.

- Business Intelligence Development Studio can be used to build and browse KPI information.
- You can use MDX functions to retrieve individual sections of the KPI, such as the value or goal, for use in MDX expressions, statements, and scripts.
- The example here uses several MDX functions to return the KPI value, KPI goal, KPI status, and KPI trend for the channel revenue measure of the Adventure Works database for the descendants of three members of the Fiscal Year attribute hierarchy.

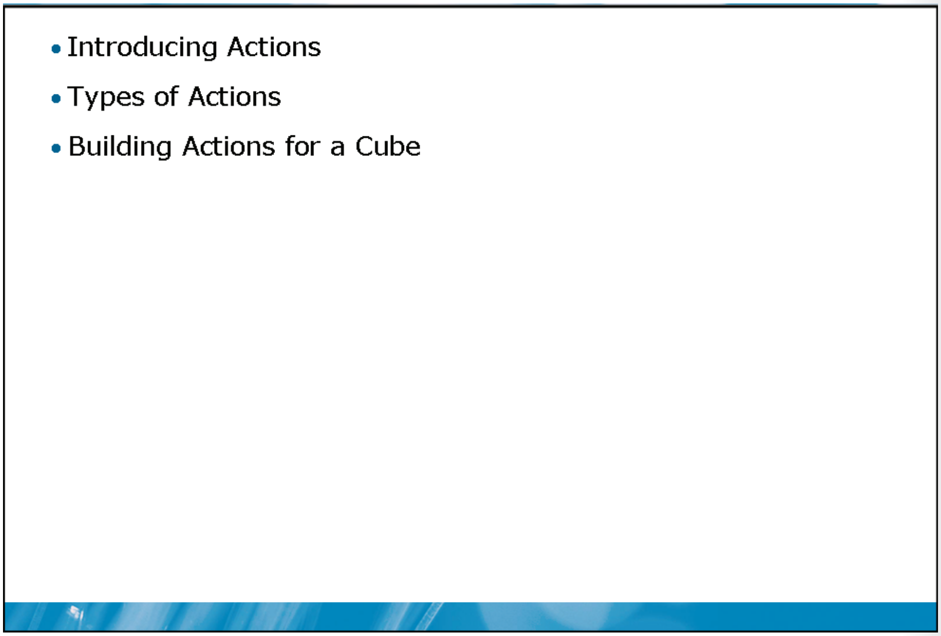
Demonstration: Using KPIs

In this demonstration, you will see how to:

- Define a KPI
- Browse a cube using a KPI

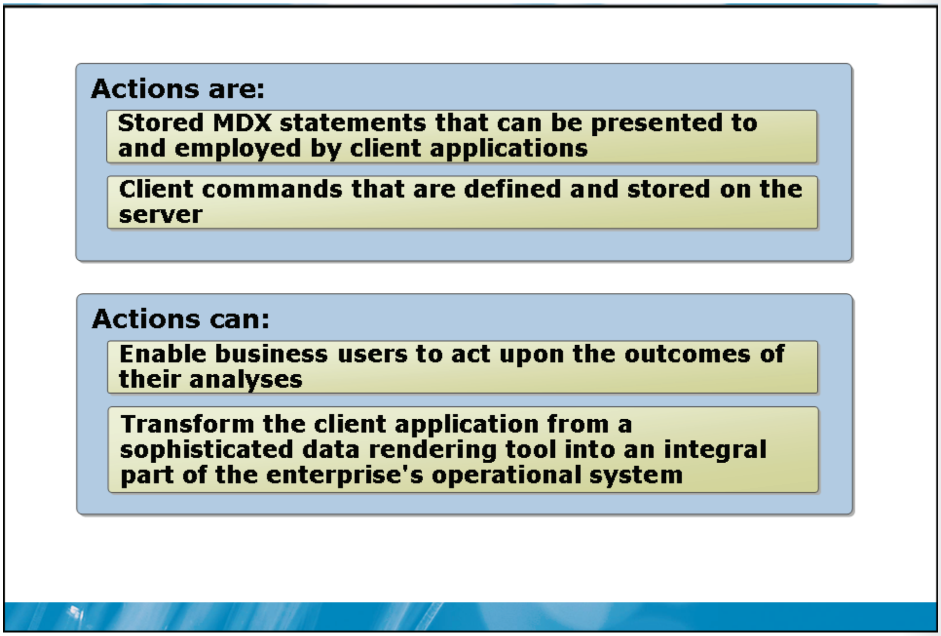
Question: How could you use KPIs in Analysis Services to your business?

Lesson 2
Working with Actions



In Microsoft SQL Server Analysis Services, an action is a stored MDX statement that can be presented to and employed by client applications. In other words, an action is a client command that is defined and stored on the server. In this lesson you will be introduced to the Analysis Services concept known as Actions. You will learn what Actions are, discuss the types of Actions available in SSAS, and review the many elements of an Action.

Introducing Actions



Key Points

An action is a stored MDX statement that can be presented to and employed by client applications.

- A simple Action object is composed of: basic information, the target where the action is to occur, a condition to limit the action scope, and the type.
  - Basic information includes the name of the action, the description of the action, the caption suggested for the action, and others.
  - The target is the actual location in the cube where the action is to occur and is composed of a target type and a target object.
  - Target type could be level members, cells, hierarchy, hierarchy members, or others.
  - The target object is a specific object of the target type; if the target type is hierarchy, then the target object is any one of the defined hierarchies in the cube.

- Actions enable business users to act upon the outcomes of their analyses.
- By saving and reusing actions, end users can go beyond traditional analysis, which typically ends with presentation of data, and initiate solutions to discovered problems and deficiencies, thereby extending the business intelligence application beyond the cube.
- You can exercise flexibility when you create actions: for example, an action can launch an application, or retrieve information from a database.
- Actions can be configured to be triggered from almost any part of a cube, including dimensions, levels, members, and cells, or create multiple actions for the same portion of a cube.

Types of Actions

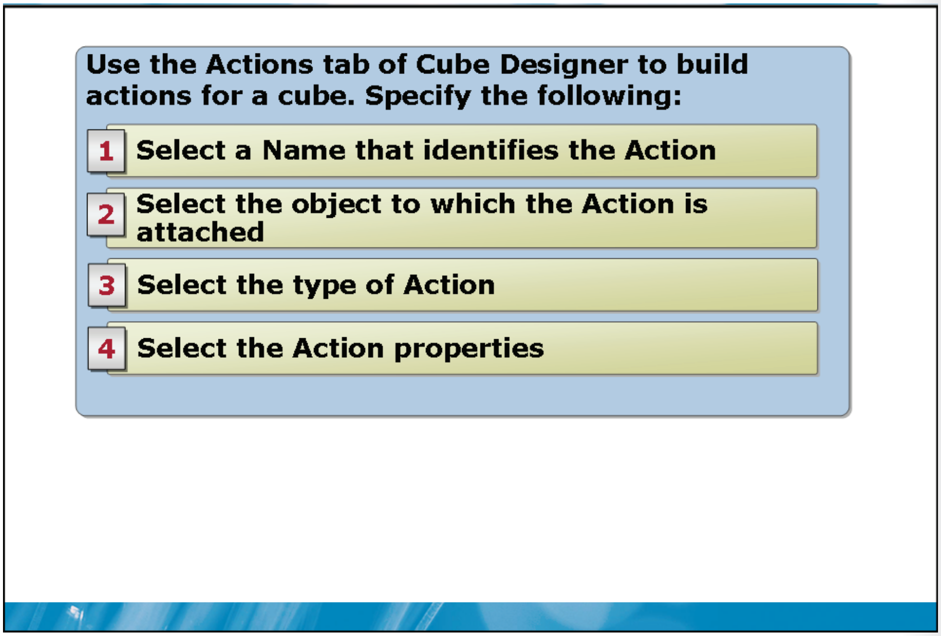
Action Type	Description
CommandLine	Executes a command
Dataset	Returns a dataset
Drillthrough	Returns a drillthrough statement as an expression
Html	Executes an HTML script
Proprietary	Performs an operation
Report	Submits a parameterized URL-based request
Rowset	Returns a rowset
Statement	Runs an OLE DB command
URL	Displays a dynamic Web page

Key Points

Actions can be of different types and they have to be created accordingly.

- Actions can be:
  - Drillthrough actions, which return the set of rows that represents the underlying data of the selected cells of the cube where the action occurs.
  - Reporting actions, which return a report from Reporting Services that is associated with the selected section of the cube where the action occurs.
  - Standard actions, which return the action element (URL, HTML, DataSet, RowSet, and other elements) that is associated with the selected section of the cube where the action occurs.
- To create a reporting action, on the Cube menu, click New Reporting Action, and then provide a server name, server path, and report format.
- To create a new drillthrough action, on the Cube menu, click New Drillthrough Action, and then specify drillthrough columns.

Building Actions for a Cube



Key Points

Use the Actions tab of Cube Designer to build actions for a cube and specify the following:

- **Name**: select a name that identifies the action.
- **Action Target**: select the object to which the action is attached. For Target type, select from the following objects:
  - Attribute members
  - Cells
  - Cube
  - Dimension members

    - Hierarchy
    - Hierarchy members
    - Level
    - Level members
- **Action Content**: select the type of action from the following:

Type	Description
Data Set	Retrieves a dataset.
Proprietary	Performs an operation by using an interface other than those listed here.
Row Set	Retrieves a rowset.
Statement	Runs an OLE DB command.
URL	Displays a variable page in an Internet browser.

- **Additional Properties**: select additional action properties from the following:

Property	Description
Invocation	Specifies how the action is run. Interactive, the default, specifies that the action is run when a user accesses an object. The possible settings are: • Batch • Interactive • On Open
Application	Describes the application of the action.
Description	Describes the action.
Caption	Provides a caption that is displayed for the action. If the caption is MDX, specify True for Caption is MDX.
Caption is MDX	Specify True if the caption is MDX or False if it is not.

- To create a reporting action, on the Cube menu, click New Reporting Action, and then specify the Server name, Server path, and Report format.
- To create a new drillthrough action, on the Cube menu, click New Drillthrough Action and then specify the drillthrough columns required by the action.

Question: How could you use Actions to enable your business users?

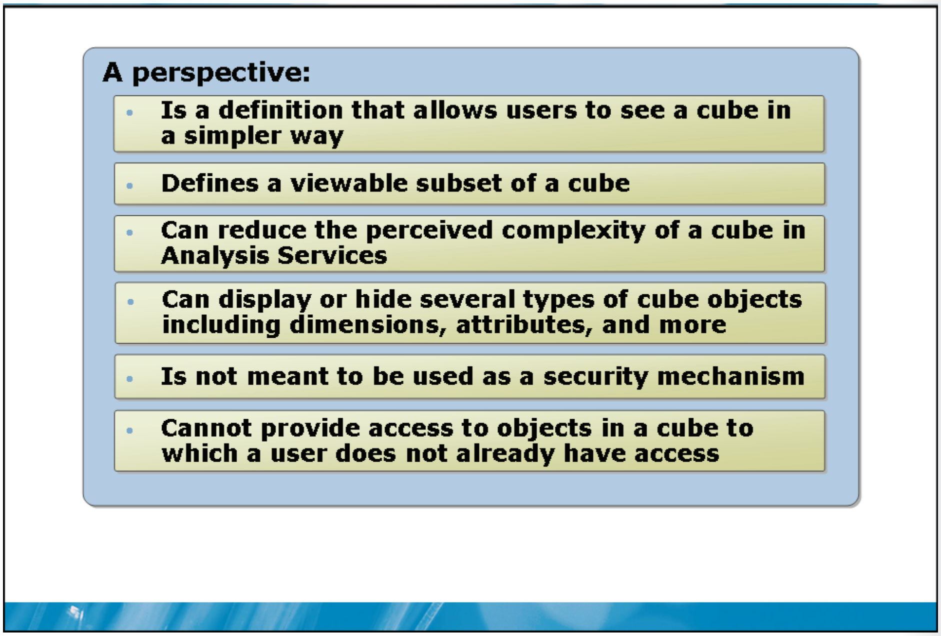
Lesson 3
Working with Perspectives

- Introducing Perspectives
- Defining Perspectives in Cube Designer

Cubes can be very complex objects for users to explore in Microsoft SQL Server Analysis Services. A single cube can represent the contents of a complete data warehouse, with multiple measure groups in a cube representing multiple fact tables, and multiple dimensions based on multiple dimension tables. Such a cube can be very complex and powerful, but daunting to users who may only need to interact with a small part of the cube in order to satisfy their business intelligence and reporting requirements.

In Microsoft SQL Server Analysis Services, you can use a perspective to reduce the perceived complexity of a cube in Analysis Services. In this lesson you will learn the concepts behind Perspectives. You will learn about what perspectives are and identify how to work with perspectives to allow you to improve the cube viewing experience for business users.

Introducing Perspectives



Key Points

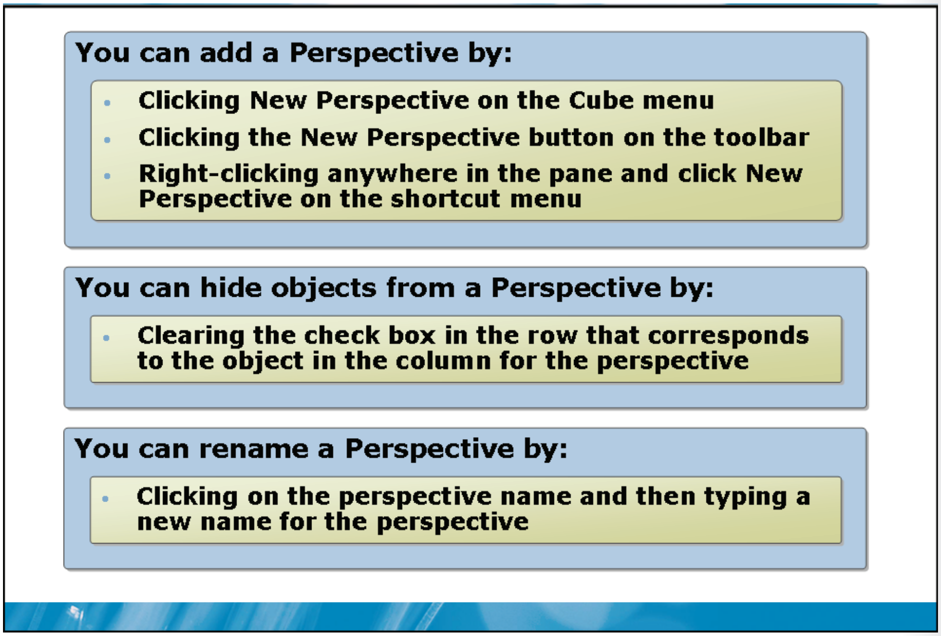
A perspective enables administrators to create views of a cube, helping users to focus on the most relevant data for them., and contains subsets of all objects from a cube.

- A simple Perspective object is composed of: basic information, dimensions, measure groups, calculations, KPIs, and actions.
    - Basic information includes the name and the default measure of the perspective.
    - The dimensions are a subset of the cube dimensions.
    - The measure groups are a subset of the cube measure groups.
    - The calculations are a subset of the cube calculations.
    - The KPIs are a subset of the cube KPIs.
    - The actions are a subset of the cube actions.

- Objects in a cube that are not visible to the user through a perspective can still be directly referenced and retrieved using XML for Analysis (XMLA), Multidimensional Expressions (MDX), or Data Mining Extensions (DMX) statements.
- Perspectives do not restrict access to objects in a cube and should not be used as such; instead, perspectives are used to provide a better user experience while accessing a cube.

Question: Why are perspectives not meant to be used as a security mechanism?

Defining Perspectives in Cube Designer



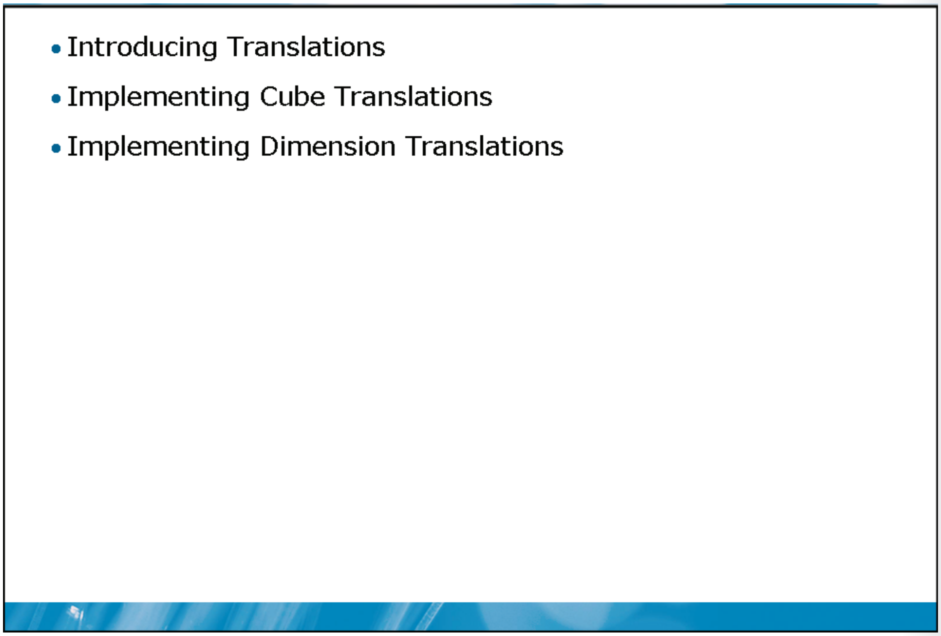
Key Points

The first column of the Perspectives tab in Cube Designer is the Cube Objects column, which lists all the objects in the cube.

- You can add a perspective to the Perspectives tab by clicking New Perspective on the Cube menu. You can also click the New Perspective button on the toolbar, or right-click anywhere in the pane and click New Perspective on the shortcut menu.
- To remove a perspective, first click any cell in the column for the perspective that you want to delete. Then, on the Cube menu, click Delete Perspective. You can also click the Delete Perspective button on the toolbar, or right-click any cell in the perspective you want to delete, and then click Delete Perspective on the shortcut menu.

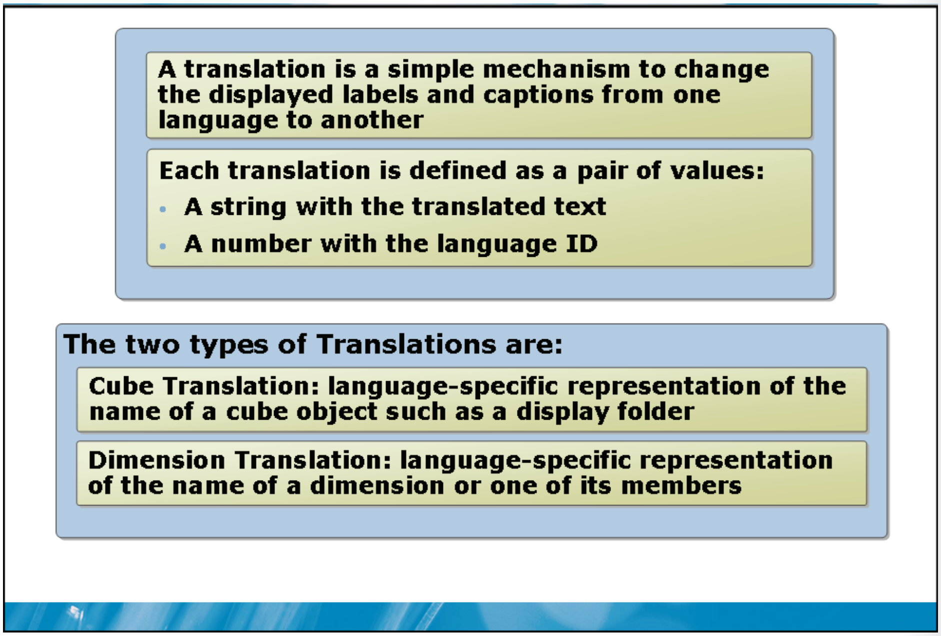
- When you create a perspective, the name is initially Perspective (followed by an ordinal number, starting with 1, if there is already a perspective called Perspective). You can click the name to edit it.
- To hide an object from a perspective, clear the check box in the row that corresponds to the object in the column for the perspective.

Lesson 4
Working with Translations



Multilanguage support in Microsoft SQL Server Analysis Services is accomplished by using translations. In this lesson you will be introduced to the concepts involved in using translations and discuss how to implement cube translations and dimension translations.

Introducing Translations



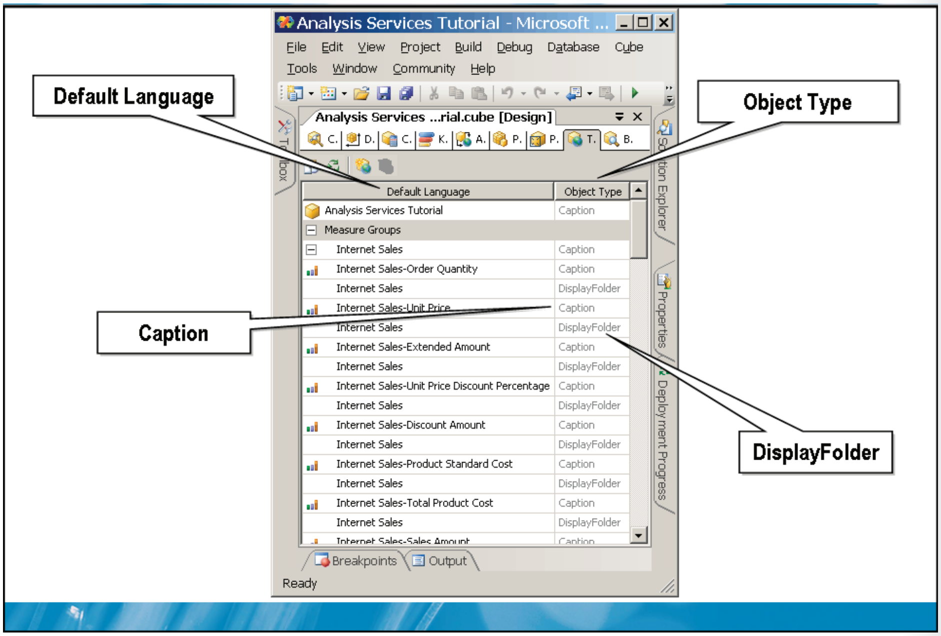
Key Points

A translation is a simple mechanism to change the displayed labels and captions from one language to another. Translations are available for all objects in Analysis Services.

- A simple Translation object is composed of: language ID number, and translated caption.
  - The language ID number is an Integer with the language ID.
  - The translated caption is the translated text.
- In Microsoft SQL Server Analysis Services, a cube translation is a language-specific representation of the name of a cube object, such as a caption or a display folder.
- Translations provide server support for client applications that can support multiple languages.
- It is useful to be able to translate various elements of a cube into a different language so that these users can view and understand the cube's metadata.

- The collation and language information for the client computer is stored in the form of a locale identifier (LCID).
  - Upon connection, the client passes the LCID to the instance of Analysis Services.
  - The instance uses the LCID to determine which set of translations to use when providing metadata for Analysis Services objects to each business user.
  - If an Analysis Services object does not contain the specified translation, the default language is used to return the content back to the client.

Implementing Cube Translations



Key Points

Click the New Translation button on the toolbar of the Translations tab in Cube Designer to create a new translation.

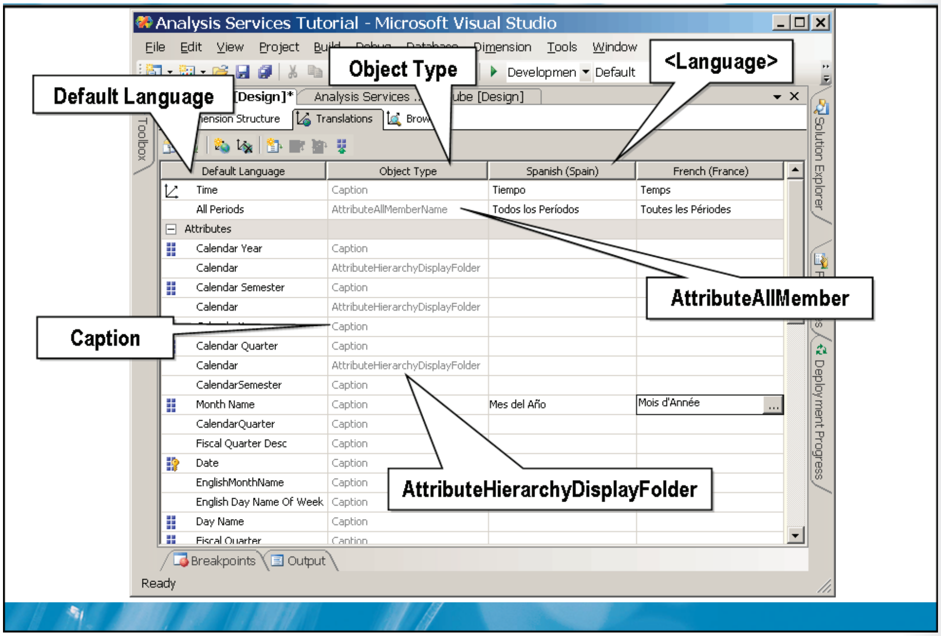
- You can define the following parts of a cube translation:
  - **Default Language**

    Sets the names of the cube objects in the default language.
  - **Object Type**

    Displays the property that will be translated.
  - **<Language>**

    Type the value of the property for the cube object in the selected language.
  - Select New Translation to display the Select Language dialog box and create a new translation.
  - Select Delete Translation to delete the selected translation.

Implementing Dimension Translations



Key Points

Use the Translations tab in Dimension Designer to define and manage translations for the dimension.

- You can define the following parts of a dimension translation:
  - **Default Language**

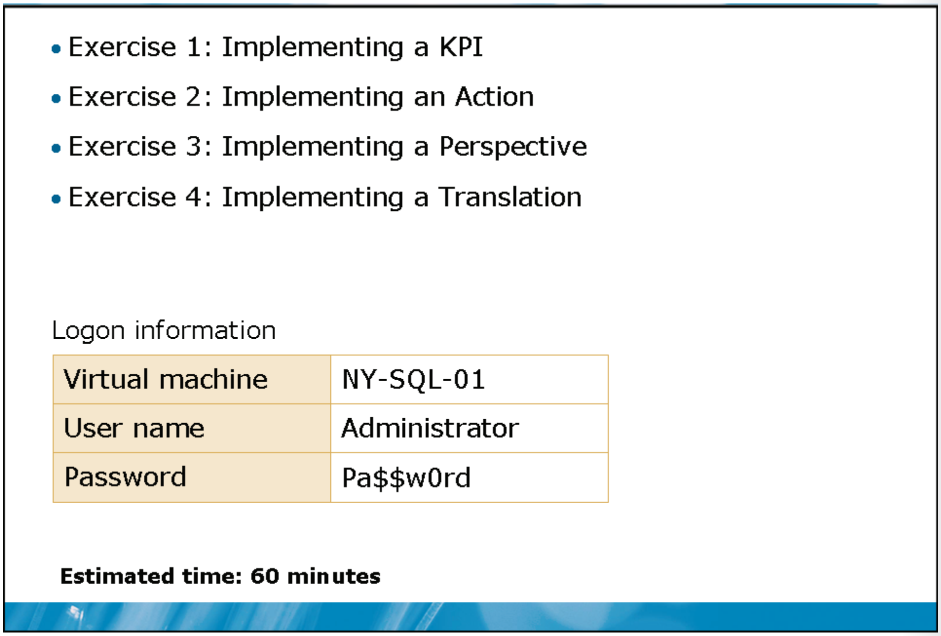
    Sets the names of the dimension objects in the default language.
  - **Object Type**

    Displays the property that will be translated.
  - **<Language>**

    Type or select the property value of the dimension object in the selected language.

- Select New Caption Column to display the Attribute Data Translation dialog box and define a new caption column when you modify an attribute in the Translation Details grid.
- Select Edit Caption Column to display the Attribute Data Translation dialog box and modify an existing caption column when you modify an attribute in the Translation Details grid.

Lab: Customizing a Cube



Exercise 1: Implementing a KPI

Scenario

You have been asked to enhance the revenue information cube for your company. First, you will need to add a KPI to measure how actual reseller sales compare to sales quotas for reseller sales. You will then browse the cube using the new KPI to review you work.

The main tasks for this exercise are as follows:

1. Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator.
2. Create a new SQL Server user.
3. Define the Reseller Revenue KPI.
4. Browse the cube using the Reseller Revenue KPI.

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

- Start 6234A-NY-SQL-01 and logon as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

- Use SQL Server Management Studio to create a new user on the AdventureWorksDW2008 database.
- Name the user **sqlserver**, set the default schema to **dbo**, grant the user database role membership in **db_owner** and **db_securityadmin**.

▶ Task 3: Define the Reseller Revenue KPI

- Use the Adventure Works cube contained in project **Mod06Lab**.
- You will need to use the **Sales Amount - Fact Reseller Sales** measure and the **Sales Amount Quota** measure along with the **Gauge** status graphic in order to properly create the KPI.
- The **status expression** is found in **E:\MOD06\Labfiles\Democode\UsingKPI_1.txt**.
- The **trend expression** is found in **E:\MOD06\Labfiles\Democode\UsingKPI_2.txt**.

▶ Task 4: Browse the cube using the Reseller Revenue KPI

- Use the KPI Browser pane to view reseller sales.
- Create a filter that shows data for Sales Territory 2, 3, and 4 during month 1 of the calendar year.

> **Results:** After this exercise, you should have defined the reseller revenue KPI and browsed the cube using the reseller revenue KPI.

Exercise 2: Implementing an Action

Scenario

In this exercise you will enhance the revenue information cube for your company by creating a drillthrough action to return order detail information for sales to customers over the Internet.

The main tasks for this exercise are as follows:

1. Define the Drillthrough Action properties.
2. Use the Drillthrough Action.

▶ Task 1: Define the Drillthrough Action properties

- Create a drillthrough action named **Internet Sales Details Drillthrough Action** using the **Internet Sales** measure group.
- Define the action to return **Sales Order Number** for a maximum of **10** items.

▶ Task 2: Use the Drillthrough Action

- Add the **Sales Amount** measure of the **Internet Sales** measure group to the data area.
- Create a filter using the **Customer Name** user-defined hierarchy of the **Customer** dimension that shows data for total sales to **Adam Powell**.
- Apply the drillthrough created in Task 1 to the filtered data.

> **Results**: After this exercise, you should have defined a drillthrough action and used the drillthrough action to return sales information.

Exercise 3: Implementing a Perspective

Scenario

In this exercise you will further enhance the revenue information cube by implementing a perspective for the Internet Marketing group.

The main tasks for this exercise are as follows:

1. Define a Sales Summary perspective.
2. Browse the cube using the Sales Summary perspective.

▶ Task 1: Define a Sales Summary perspective

- Create a new perspective named **Sales Summary.**
- Define the perspective to show users only the **Fact Internet Sales Count** and **Fact Reseller Sales Count** measures of the cube.

▶ Task 2: Browse the cube using the Sales Summary perspective

- Clear all measures and hierarchies from the **Data** pane and all dimensions from the **Filter** pane before beginning.
- Select the **Sales Summary** perspective and expand the **Internet Sales** and **Reseller Sales** measures to review your work.

Results: After this exercise, you should have defined a perspective and browsed the cube using the new perspective.

Exercise 4: Implementing a Translation

Scenario

In this exercise you will add the final enhancement to the revenue information cube by specifying translations for the French speakers in your company.

The main tasks for this exercise are as follows:

1. Specify translations for the Time Dimension metadata.
2. Specify translations for the Adventure Works cube metadata.
3. Browse the cube using the new translations.

▶ Task 1: Specify translations for the Time Dimension metadata

- Using **Dimension Designer** for the **Date** dimension, create a **French (France)** translation.
- For the **Caption** object of the **Date** dimension, under the **French (France)** column, enter **Temps**.
- For the **Caption** object of the **Month Number of Year** attribute, under the **French (France)** column, enter **Mois d'Année**.

▶ Task 2: Specify translations for the Adventure Works cube metadata

- Using **Cube Designer** for the **Adventure Works** cube, create a new **French (France)** translation.
- For the **Caption** object of the **Internet Sales** measure group, under the **French (France)** column, enter **Ventes D'Internet**.
- For the **Caption** object of the **Internet Sales-Sales Amount** measure, under the **French (France)** column, enter **Quantité de Ventes d'Internet**.

▶ Task 3: Browse the cube using the new translations

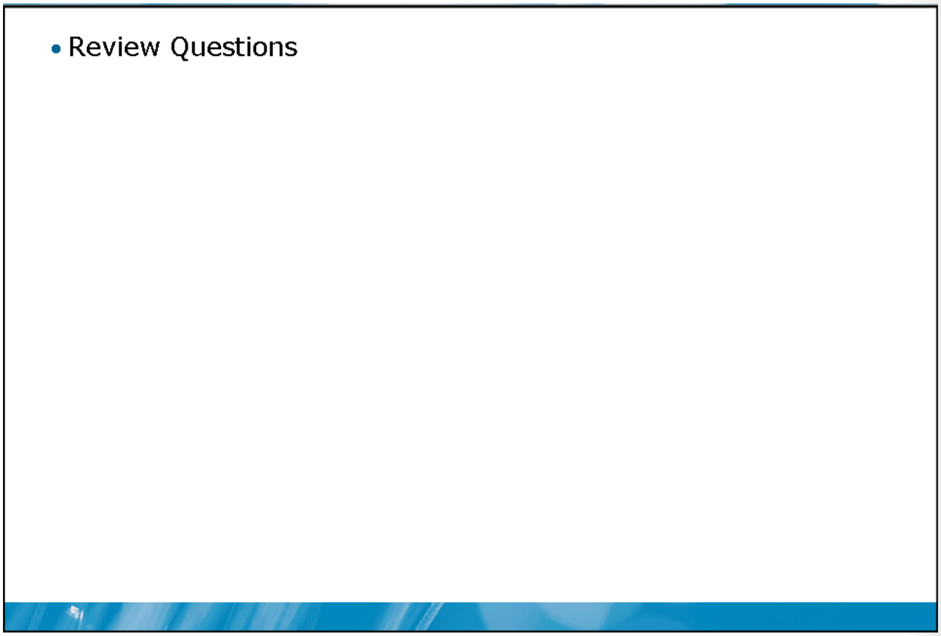
- Clear all measures and hierarchies from the **Data** pane and all dimensions from the **Filter** pane before beginning.
- Select **French (France)** in the **Language** list and review **Measures** in the **Metadata** pane to review the translations.

> **Results**: After this exercise, you should have specified translations for the time dimension metadata, specified translations for the Adventure Works cube metadata, and browsed the cube using the new translations.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-01 virtual machine and discard any changes.

Module Review and Takeaways



Review Questions

1. How are KPIs useful to the business?
2. What value do Actions add to the business?
3. What objects can be displayed or hidden in a perspective?

Module 7

Deploying and Securing an Analysis Services Database

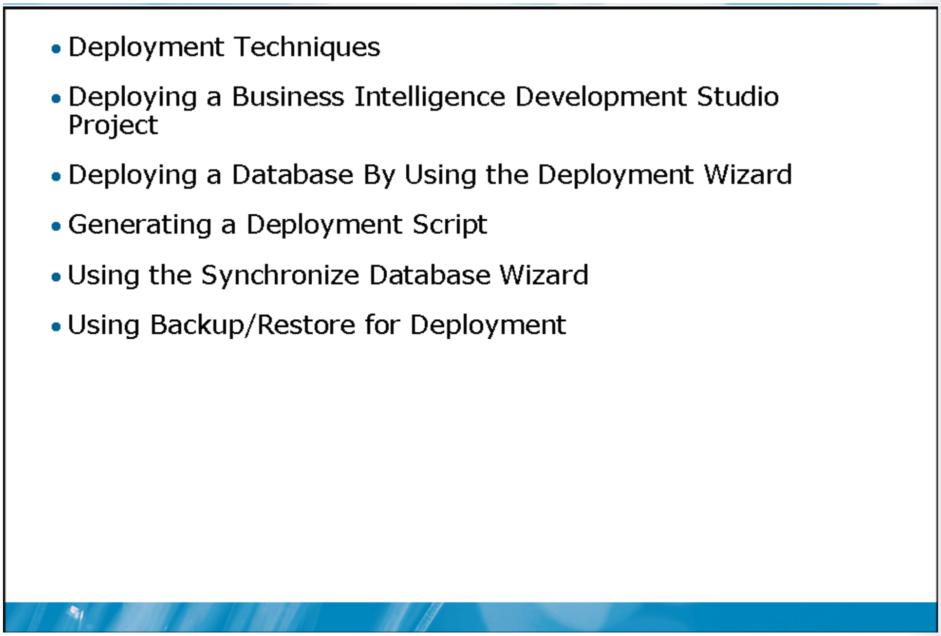
Contents:

Module Overview

- Deploying an Analysis Services Database
- Securing an Analysis Services Database

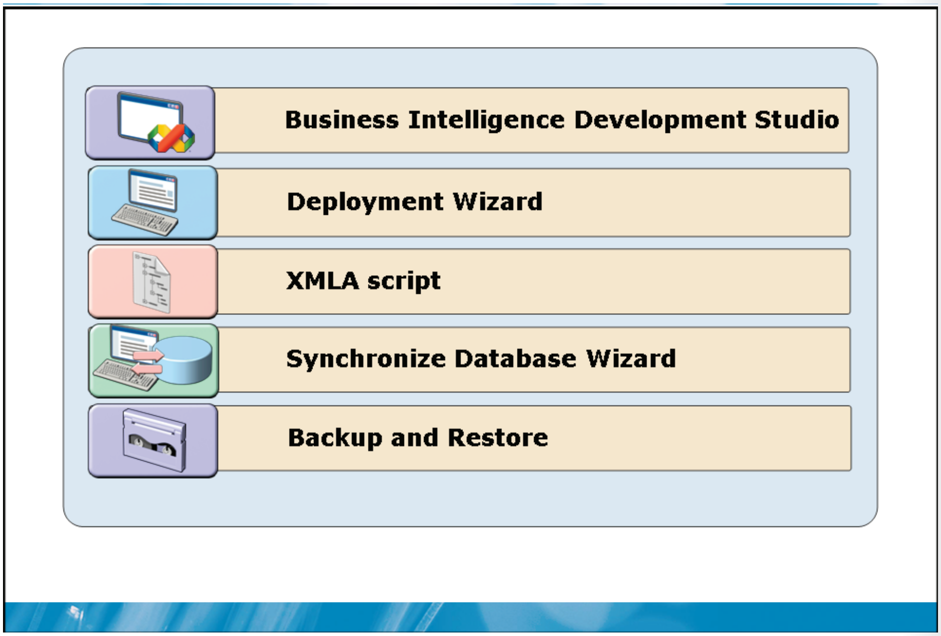
After you have completed the development of a Microsoft® SQL Server® Analysis Services project in Business Intelligence Development Studio and have deployed and tested the project in your development environment, you are ready to deploy the Analysis Services database to test and production servers. In addition, the process of securing SQL Server Analysis Services occurs at multiple levels. You must secure each instance of Analysis Services and its data sources to make sure that only authorized users have read or read/write permissions to selected cubes, dimensions, cells, mining models, and data sources, and to prevent unauthorized users from maliciously compromising sensitive business information.

Lesson 1:
Deploying an Analysis Services Database



This lesson will explain how to deploy an Analysis Services database by providing information about the planning and deployment of an Analysis Services project to test or production servers. There are five possible deployment methods that can be used to deploy an instance of Analysis Services. Each of the methods should be considered based on the needs of the scenario and environment.

Deployment Techniques



Key Points

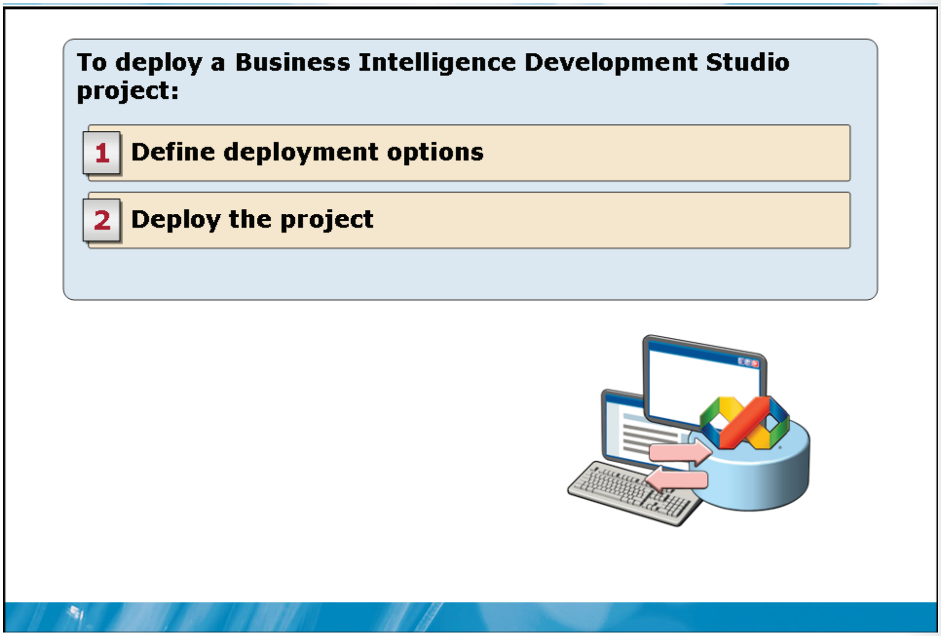
There are five deployment techniques available for Analysis Services databases and each of these has advantages based on the complexity of deployment and whether synchronization or incremental updates are required.

- You can use Business Intelligence Development Studio to deploy a solution during development.
- You can use the Deployment Wizard to deploy SQL Server Analysis Services databases.
- You can create XMLA scripts to generate the database metadata by using SQL Server Management Studio.

- You can use the Synchronize Database Wizard in Management Studio to synchronize Analysis Services databases on separate instances.
- You can use backup and restore in Management Studio, which provides a basic method of deploying and Analysis Services database.

Question: What methods do you use to deploy an Analysis Services database in your organization today?

Deploying a Business Intelligence Development Studio Project



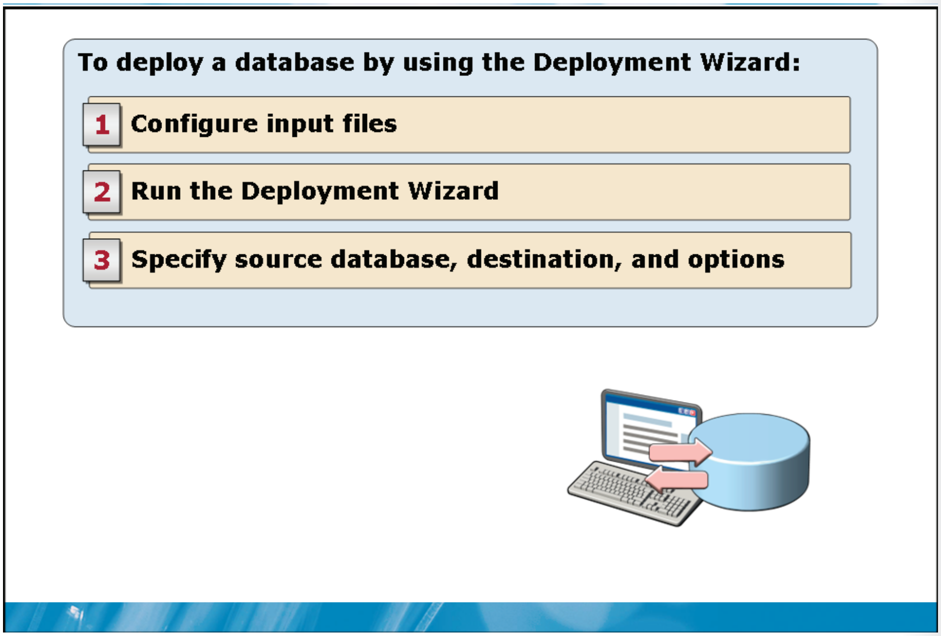
Key Points

While you are in the process of developing an Analysis Services project, you will need to build and deploy the project to test its functionality. You can use Business Intelligence Development Studio to configure the deployment options and deploy the project.

- After you have set the deployment options, you can deploy the project by clicking Build projectname on the Build menu, or by right-clicking the project in Solution Explorer and then clicking Build.

Question: Why is it not a good idea to use Business Intelligence Development Studio to deploy production databases?

Deploying a Database by Using the Deployment Wizard



Key Points

The Deployment Wizard uses the input files created in an Analysis Services project as settings to deploy a database.

To start the Deployment Wizard, click Start, point to All Programs, point to Microsoft SQL Server 2008, point to Analysis Services, and then click Deployment Wizard.

There are several steps to complete the Deployment Wizard:

1. Specify the source Analysis Services database.
2. Specify the installation target.
3. Specify whether you want to replace partitions and roles or keep existing ones.

4. Specify the configuration properties, including configuration settings, optimization settings, and impersonation settings.
5. Select whether you want to process the database and whether processing should occur in a single transaction.
6. Confirm the deployment or store the results in a deployment script for later execution.

Generating a Deployment Script

Key Points

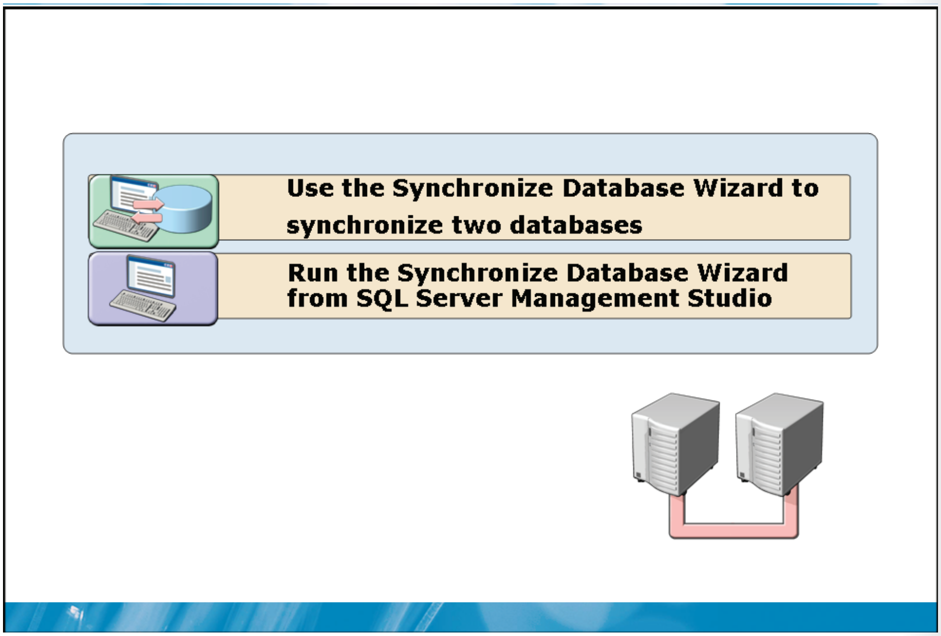
You can also use the Deployment Wizard to generate a deployment script, which you can use to deploy a database later. You can edit the scripts to customize a deployment.

- You generate a deployment script by using the Create deployment script option in the Deployment Wizard, or if you are running the Deployment Wizard from the command prompt, you can use the /o switch.
  - You can edit the deployment script.

You can run the XMLA script from Management Studio to recreate the database objects defined in the script. You can also create a script in Management Studio.

Question: Under what circumstances would you want to edit the deployment script?

Using the Synchronize Database Wizard

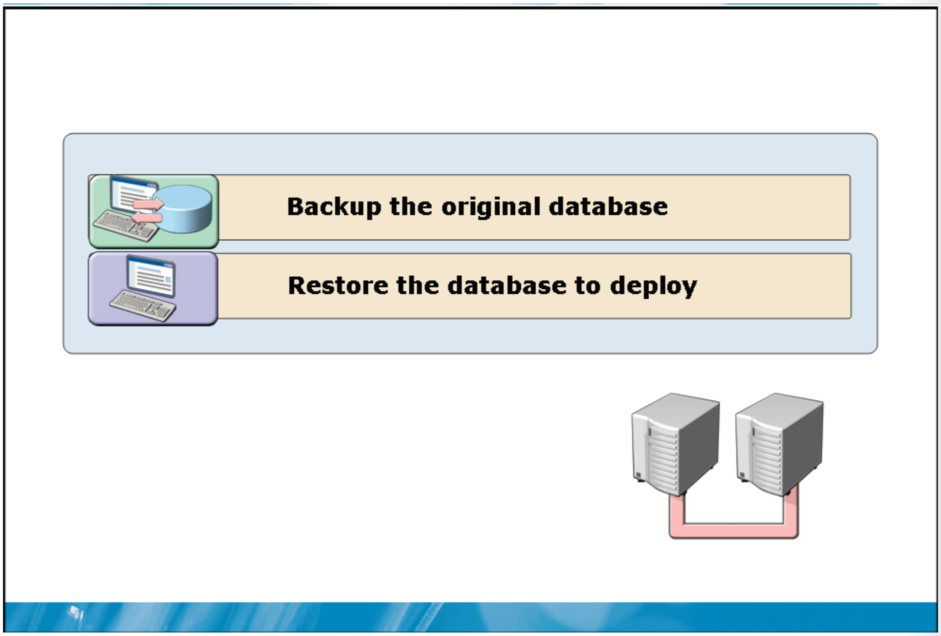


Key Points

You can use the Synchronize Database Wizard to deploy a database or to synchronize previously deployed databases.

- The Synchronize Database Wizard provides a one-way synchronization between two Analysis Services databases.
- To use the Synchronize Database Wizard, right-click the Databases folder in SQL Server Management Studio, and then click Synchronize.

Using Backup/Restore for Deployment



Key Points

You can use backup and restore in Management Studio, which provides a basic method of deploying and Analysis Services database. There is no ability to modify the target database before deployment and no ability to incrementally update objects.

Question: What are some scenarios in which you might want to use Backup and Restore to deploy an Analysis Services Database?

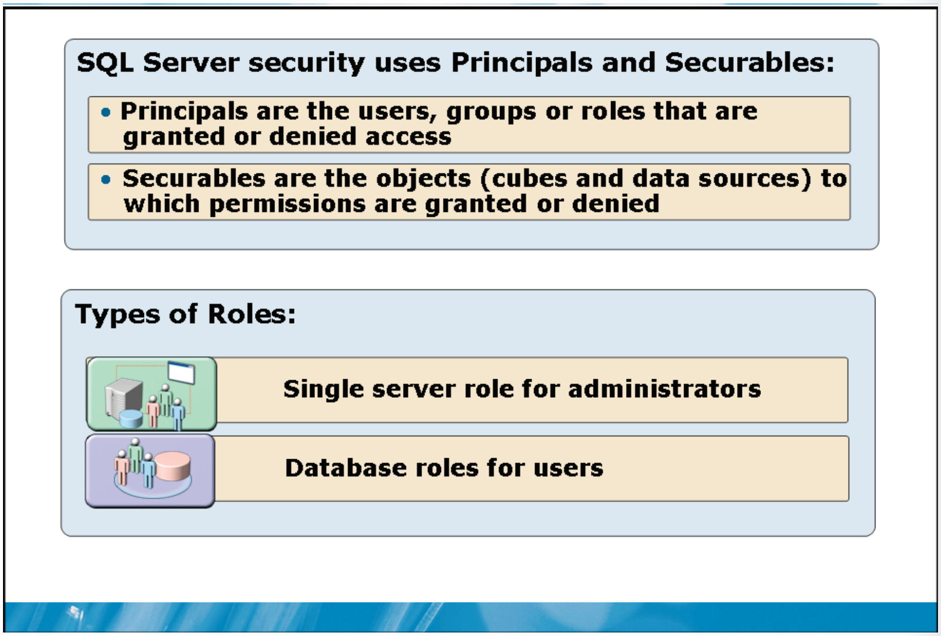
Lesson 2: Securing an Analysis Services Database

- Analysis Services Security Model
- Adding Members to the Server Role
- Creating Database Roles
- Granting Data Source and Cube Permissions
- Granting Cell Permissions
- Granting Dimension and Dimension Data Permissions
- Testing Security Permissions

After you install an instance of SQL Server Analysis Services, only the members of the server role have server-wide permissions to perform any task within the instance of Analysis Services. By default, no other users have any access permissions to the objects in the instance. Members of the Analysis Services server role can grant other users access to server and database objects by using SQL Server Management Studio, Business Intelligence Development Studio, or an XMLA script.

This lesson describes how a member of the Analysis Services server role can grant other users access to Analysis Services.

Analysis Services Security Model



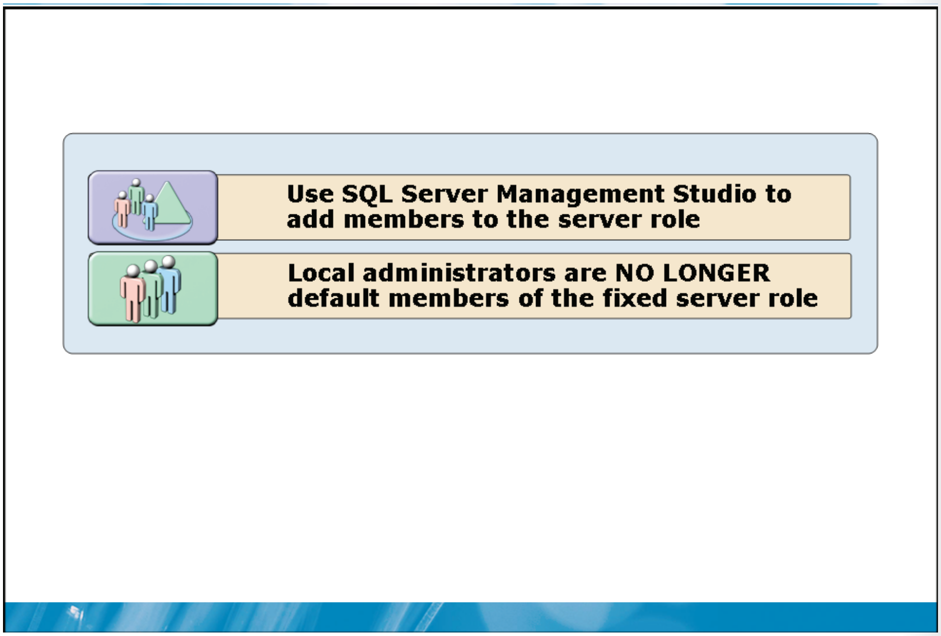
Key Points

Analysis Services relies on Microsoft® Windows® Authentication to authenticate users. After a user has been authenticated, Analysis Services controls permissions within the databases based on the user's role membership.

- Principals are entities that can request SQL Server resources.
- Securables are the resources to which the SQL Server Database Engine authorization system regulates access.

- Analysis Services has a single fixed server role for administrators. Members of this role have full permissions to perform any action in the instance.
- Administrators can create database roles for users. You can grant these roles user access and database administrator rights. Role permissions are database-specific.
    - You can grant database role permissions for database and cube dimensions, individual dimension members, cubes, individual cells within a cube, mining structures, mining models, data sources, and stored procedures.

Adding Members to the Server Role



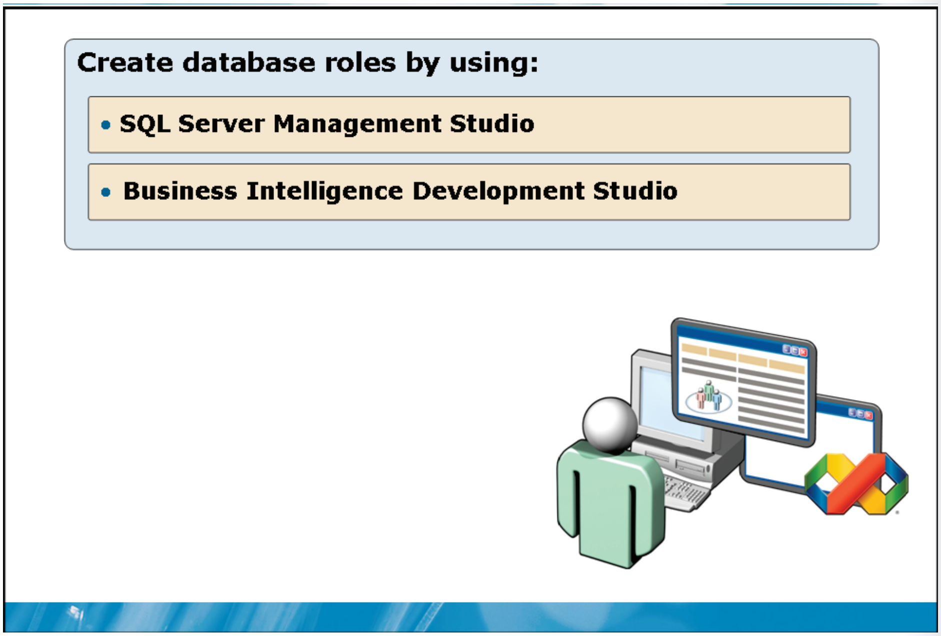
Key Points

The server role is a fixed role at the server level. You cannot change its permissions and you cannot add further server roles.

- Use Management Studio to add members to the server role. You can add members to the server role on the Security page of the Analysis Server Properties dialog box.
- In SQL Server 2008, the local Windows Group BUILTIN\Administrator is no longer included in the SQL Server sysadmin server role.

Question: What is meant when referring to a policy of "Least Privileged Authority"?

Creating Database Roles



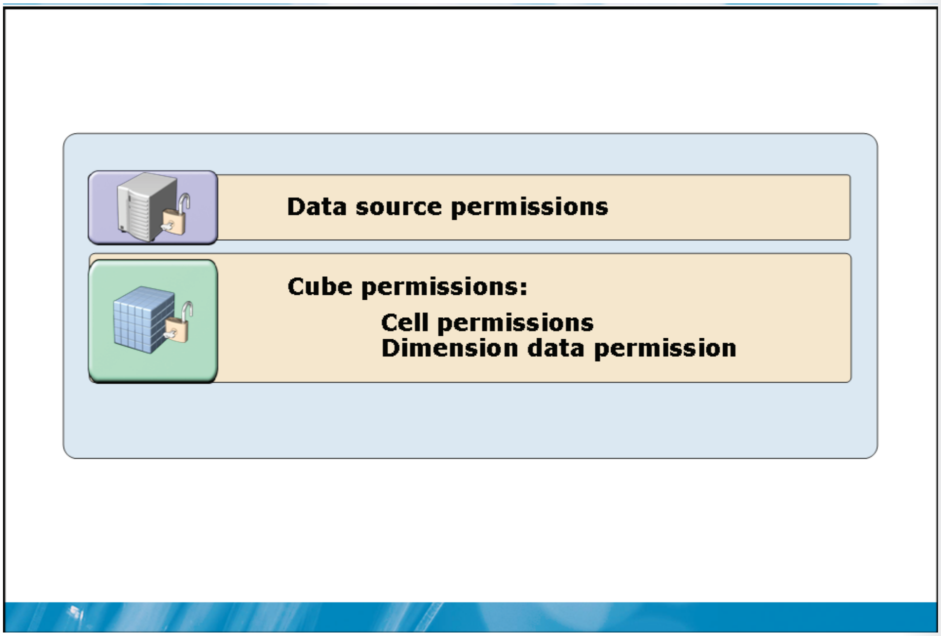
Key Points

To assign permissions to users, you should create database roles.

- You can create database roles by using Management Studio or Business Intelligence Development Studio.
  - In SQL Server Management Studio, expand the database and right-click the Roles folder to add a new role.
- In Business Intelligence Development Studio, roles are displayed in Solution Explorer and you can add them from the Project menu.

Question: How might your organization benefit from better defining database roles?

Granting Data Source and Cube Permissions



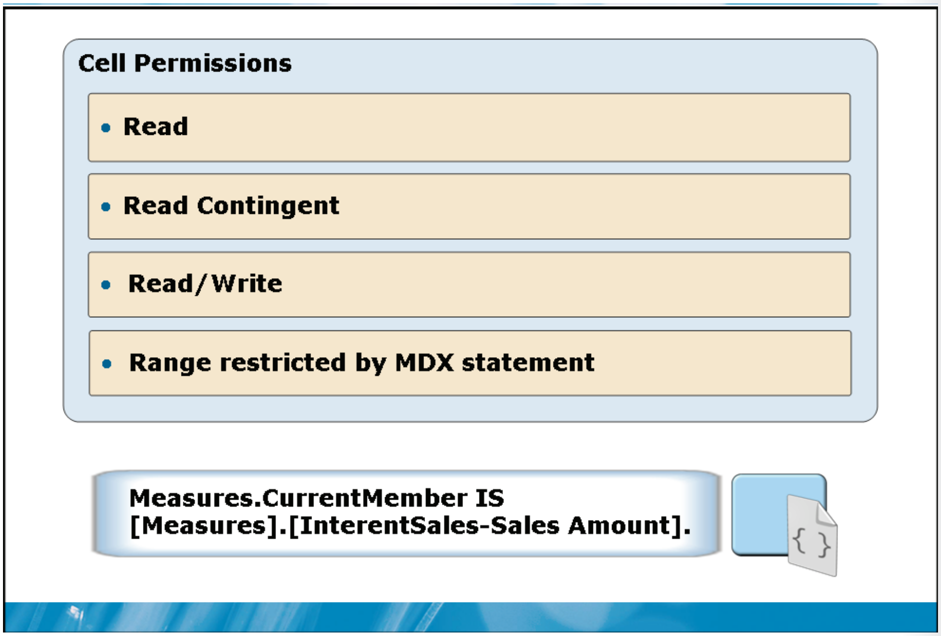
Key Points

To access data within the database users must be members of a role that has permissions to access the cube. Permissions to access a data source are not required to access cube data.

- Users do not typically need access to data sources.
- You can grant roles read or read/write permission to cubes within the database. By default a role with permissions for the cube has access to the individual cells within the cube. By default a role with permissions for the cube will also have read permissions on the dimensions and dimension members that the cube contains.

Question: When would you restrict access to a cell versus a dimension?

Granting Cell Permissions



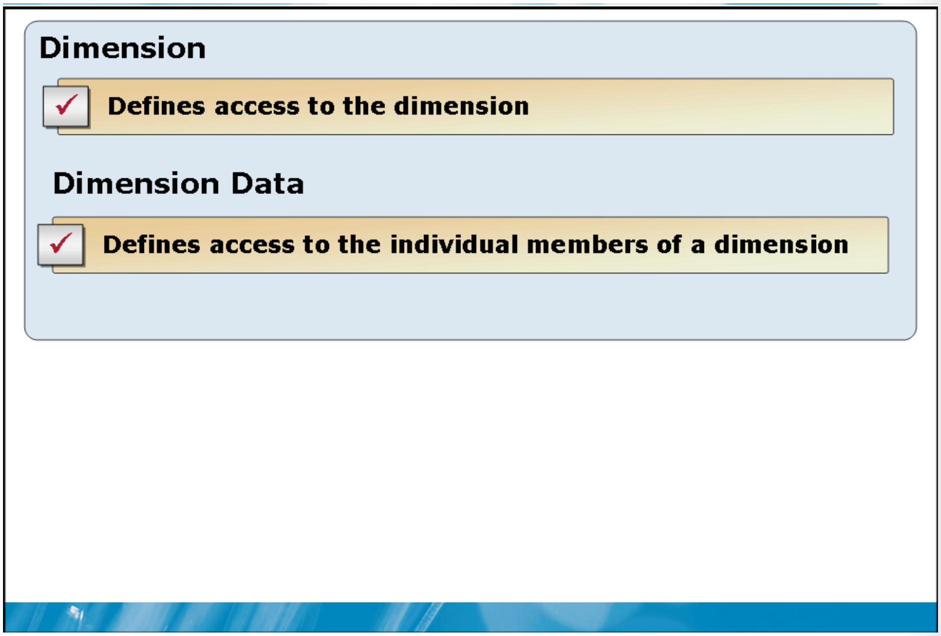
Key Points

A role with read-only or read and write permissions is granted permissions on the individual cells until you apply cell security. If you grant access to one or more cells, the role is denied access to all other cells until you grant permissions on them.

- Read access allows members of the role to read data in the cells.
- Read-contingent access allows members of the role to read data in the cells.
- Read/write access allows members to read and write to the cells.
- By default, cell permissions apply to all cells in the cube. Use an MDX statement to limit the range of cells to which the permissions apply.

Question: When would you use read contingent permissions?

Granting Dimension and Dimension Data Permissions



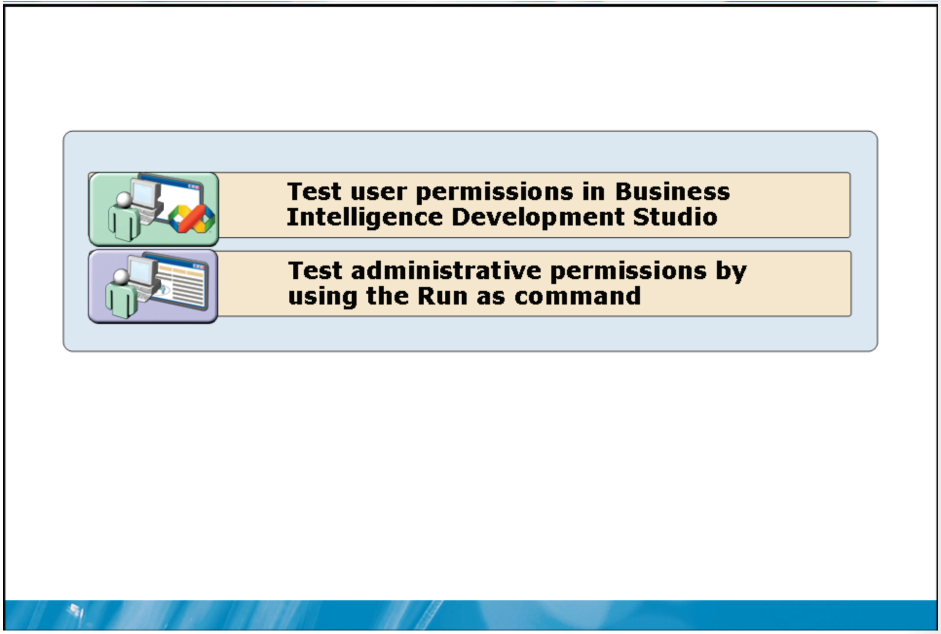
Key Points

By default, a role that has access to a cube has read access to all of the dimensions in the cube. You can modify this access and deny access to one or more members in the dimension.

- You can change dimension access to read/write access and grant permission to allow members of the role to process the dimension. By default, dimension access is set to read-only.
- You can change dimension data access to define access to individual members of the dimension. By default, members of the role have access to all members of the dimension.
- You can use an MDX statement for an allowed member set, a denied member set, or a default member for the role.

Question: When might you allow users to process a dimension?

Testing Security Permissions

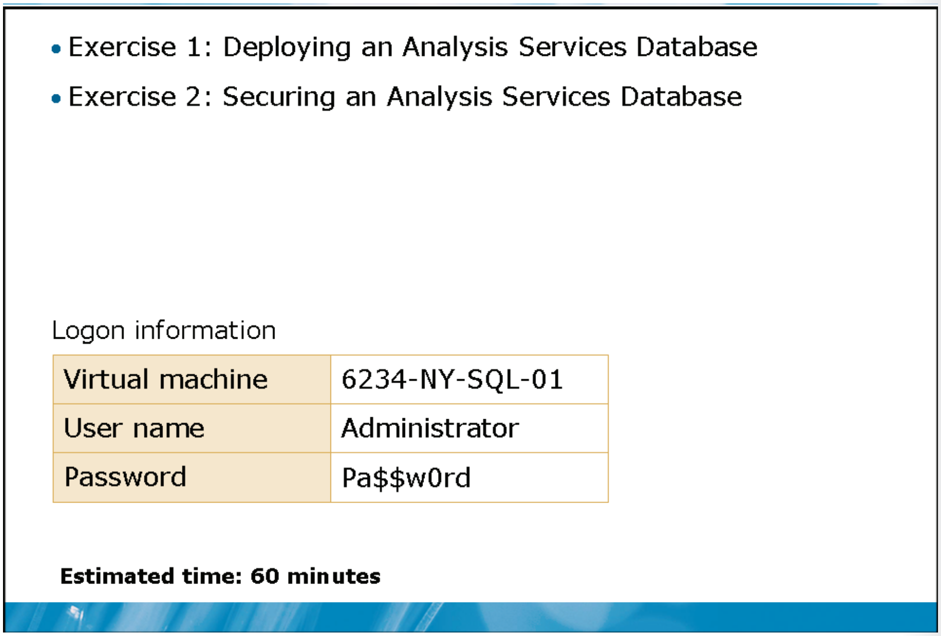


Key Points

After you have configured security for an Analysis Services database, it is important to test your security settings and permissions.

- You can use Business Intelligence Development Studio to test user permissions.
  - On the Cube Designer Browser tab, you can change user to an individual or to a role to test security by clicking the Change User button.
  - On the Role Designer Cell Data tab, you can click the Test cube security hyperlink. This has the advantage that it saves objects if necessary and defaults to the current role.
- You can use the Windows Run as feature to start an application such as SQL Server Management Studio, and test administrative permissions such as those granted through membership in the server role or the database level Full Control (Administrator) permission.

Lab: Deploying and Securing an Analysis Services Database



Exercise 1: Deploying an Analysis Services Database

Scenario

You need to deploy the Analysis Services solution that analyzes sales information for your company. You must create a deployment script and test it.

This exercise's main tasks are:

1. Start the 6434A-NY-SQL-01 virtual machine and log on as Student.
2. Install the Adventure Works Data Warehouse database.
3. Use the Deployment Wizard to create a deployment script.
4. Review and run the deployment script.

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Student

- Start 6234A-NY-SQL-01, and log on as **Student** with the password of **Pa$$w0rd**.

▶ Task 2: Install the Adventure Works Data Warehouse database

- Open **E:\Mod07\Labfiles\Starter\AdventureWorks Data Warehouse**, click **instawdwdb.sql** in **SQL Server Management Studio** and execute it to install the Adventure Works Data Warehouse database.

▶ Task 3: Use the Deployment Wizard to create a deployment script

- Use **E:\Mod07\Labfiles\Starter\Adventure Works OLAP \bin\Adventure Works OLAP.asdatabase** as the source database.
  - The script should deploy the database to **localhost** with the name **Adventure Works OLAP Copy**.
- Select the default options on the **Specify Options for Partitions and Roles**, **Specify Configuration Properties**, and **Select Processing Options** pages.

▶ Task 4: Review and run the deployment script

- Use SQL Server Management Studio to execute the script you generated in the previous step.
- Use the Object Browser to verify that the **Adventure Works OLAP Copy** database was created.

> **Results:** After this exercise, you should have successfully created, reviewed and executed the Adventure Works OLAP copy deployment script.

Exercise 2: Securing an Analysis Services Database

Scenario

Now that you have successfully deployed your solution, you must secure the sensitive information so that only those employees who should have access can see certain types of data.

This exercise's main tasks are:

1. Set up the lab environment.
2. Add a user to the Analysis Services Server role.
3. Add a new role with the ability to process the database.
4. Add a new role with specific cube and dimension permissions.
5. Deploy the solution.
6. Test user security by using Business Intelligence Development Studio.

▶ Task 1: Set up the lab environment

- Run **Setup.cmd** in the **E:\Mod07\Labfiles\Starter** folder.

▶ Task 2: Add a user to the Analysis Services Server role

- Use SQL Server Management Studio to connect to the default instance of Analysis Services on **NY-SQL-01**.
- In Object Explorer, view the properties of the server, and on the **Security** page, add the **NY-SQL-01\ASAdmin** user account to the server role.

▶ Task 3: Add a new role with the ability to process the database

- Use Business Intelligence Development Studio to open the **E:\Mod07\Labfiles\Starter\Adventure Works OLAP.sln** solution.
- Create a new role named **DB Process Role.role**.
  - Use the **Properties** pane to rename the file and the object if necessary.
- Add **NY-SQL-01\ProcessAdmin** to this role.
- Use the **General** page of the Role Designer to add the **Process Database** and **Read Definition** permissions to the new role.

▶ Task 4: Add a new role with specific cube and dimension permissions

- Create a new role named **Marketing Manager Role.role** in the Adventure Works OLAP.sln solution.
  - Add **NY-SQL-01\DBradley** to this role.
- On the **Cubes** tab of the Role Designer, grant **Read** access on the **Adventure Works OLAP** cube permissions to this role.
- Use the **Cell Data** tab to give this role access to all measures except **Sales Amount Quota**.
  - You can copy the MDX code to limit the cell access from E:\Mod07\Labfiles\Starter\CellDataMDX.txt.

▶ Task 5: Deploy the solution

- Use Business Intelligence Development Studio to deploy the Adventure Works OLAP.sln solution.
  - Deploy the solution to a new database named **Adventure Works Olap Secure**.

▶ Task 6: Test user security by using Business Intelligence Development Studio

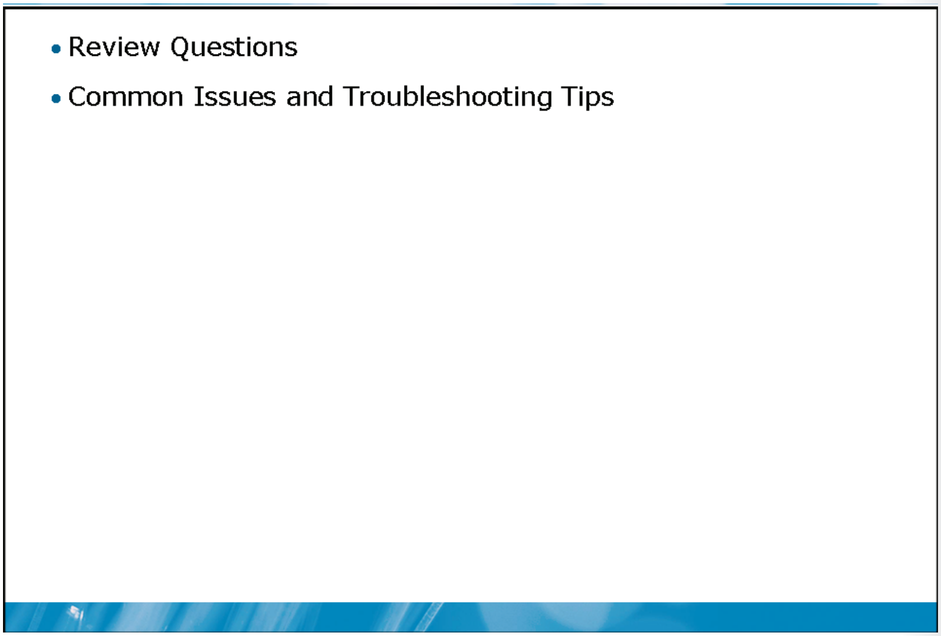
- Use the **Cell Data** tab of the Role Designer to test the cell level security that you defined.
- Use the **Browser** tab of the Cube Designer to test the cell level permissions for the **Marketing Manager Role** by using the **Change User** button.
  - Verify that the **Marketing Manager Role** can view the values of all measures except **Sales Amount Quota**.
- Use the **Change User** button on the **Browser** tab of the Cube Designer to verify that the **DB Process Role** cannot browse the cube.

Results: After this exercise, you should have successfully created and tested analysis database security roles.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-01 virtual machine and discard any changes.

Module Review and Takeaways



Review Questions

1. What are principals and securables and how do they relate to permissions?
2. Describe various scenarios in which organizations may benefit from the different deployment methods for an Analysis Services database.
3. Which installation methods can be used with scripting?
4. What is the difference between a server role and a database role?

Module 8

Maintaining a Multidimensional Solution

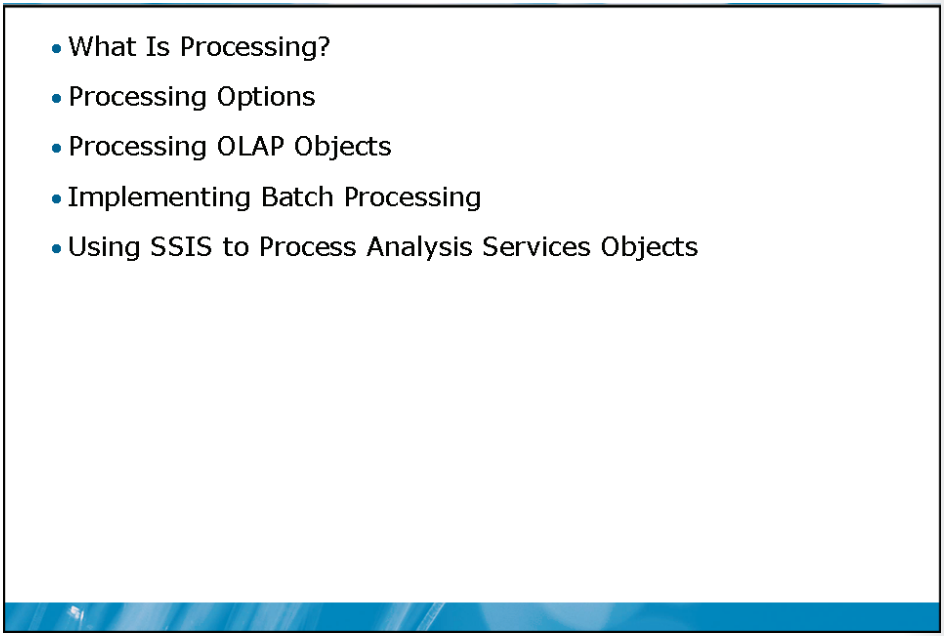
Contents:

Module Overview

- Configuring Processing Settings
- Logging, Monitoring, and Optimizing an Analysis Services Solution
- Backing Up and Restoring an Analysis Services Database

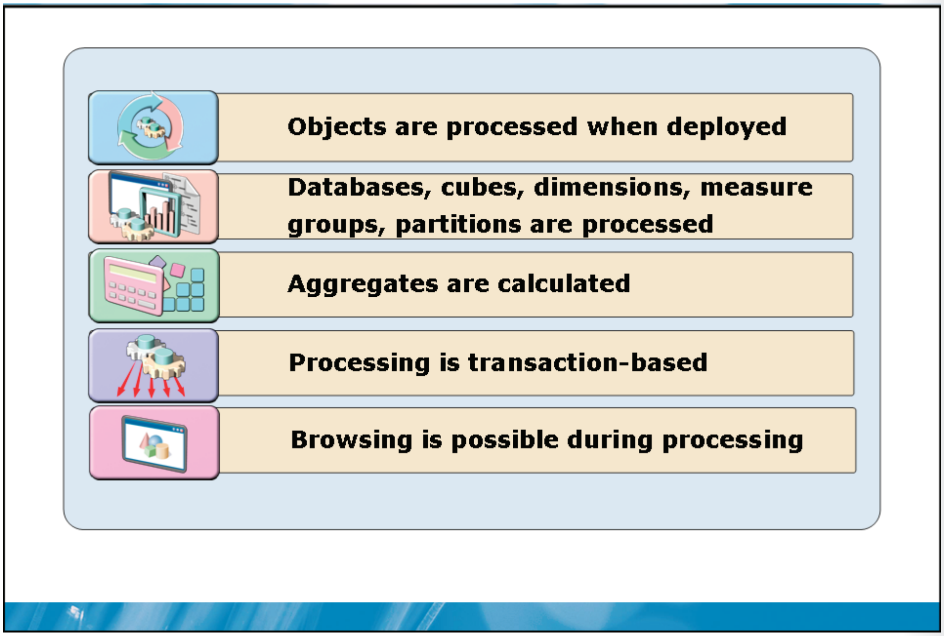
Administering Microsoft® SQL Server® Analysis Services includes managing all instances by backing up and restoring databases, processing objects, automating tasks, and monitoring activity within an Analysis Services instance. This module discusses the maintenance tasks associated with an Analysis Services solution, and describes how administrators can use the Analysis Services management tools to perform them.

Lesson 1: Configuring Processing Settings



As an administrator, you keep the SQL Server Analysis Services objects in the production databases current by processing them. Processing is the step, or series of steps, that populate Analysis Services objects with data from relational data sources. Processing is different depending on the type of object and the selection of processing options. In this lesson, you will learn about configuring processing of Analysis Service objects.

What Is Processing?



Key Points

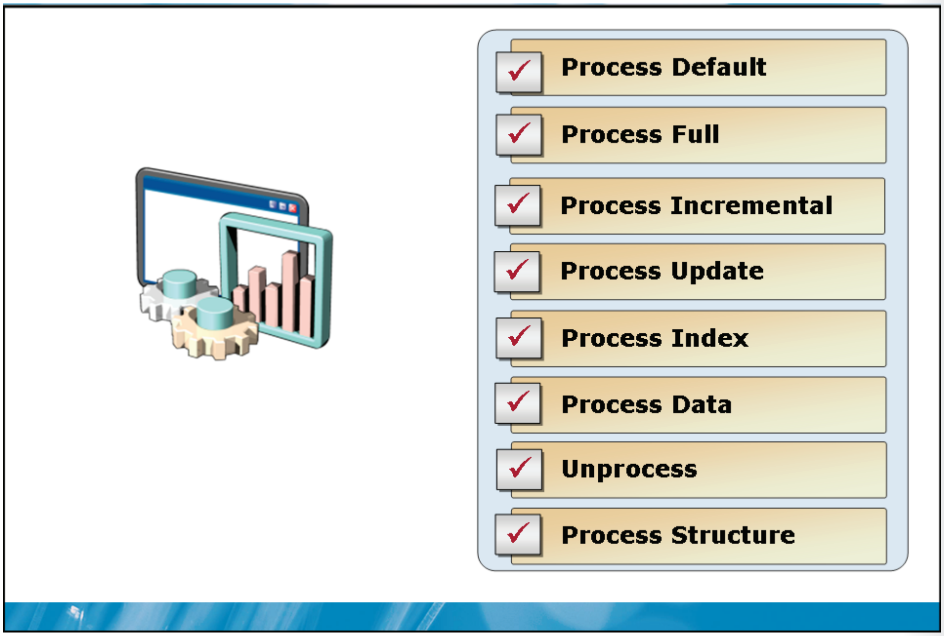
When multidimensional objects are created, they need to be deployed and processed before they are available for querying and browsing. Processing populates multidimensional objects with data from relational data sources. Processing varies depending on the object that is processed and the processing options that are selected.

- By default, all objects are processed when a project is first deployed. After deployment, you can process objects independently.
- Objects that are processed include OLAP databases, cubes, dimensions, measure groups, and partitions.

- For partitions and measure groups, processing calculates aggregates according to the aggregation scenario designed for that partition or measure group.
- Processing occurs within a transaction. If the process job fails, the transaction is rolled back.
- In Analysis Services, you can usually browse objects during processing. During the commit phase, the service takes an exclusive lock on the object and queries sent to the server will be held until the locks are freed.

Question: How would you determine when to process an object?

Processing Options



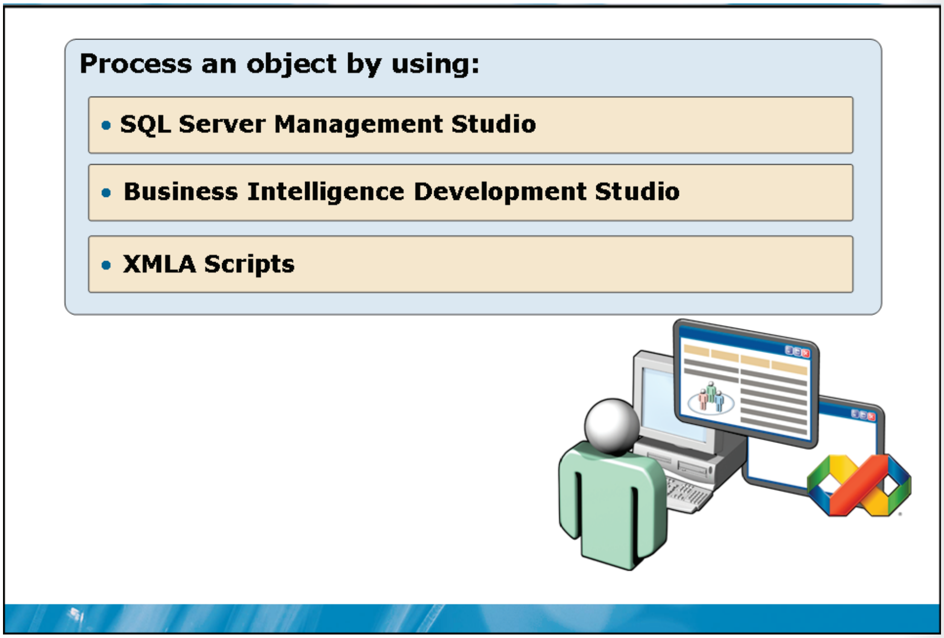
Key Points

There are different processing types available in Analysis Services. The processing options available are dependent on the type of object you are processing.

- The Process Default option detects the state of the object and executes the appropriate processing option to bring the object to a fully processed state.
- The Process Full option processes the object and all objects it contains. If the object already exists, the system will drop and rebuild the object.
- The Process Incremental option adds new fact data to a measure group or partition. Additional configuration settings are required to use the incremental option.
- The Process Update option forces a re-read of data and an update of dimension attributes.
- The Process Index option creates or rebuilds indexes and aggregations for all processed partitions.

- The Process Data option populates or removes and re-populates the data in the object, but it does not build indexes or aggregations.
- The Unprocess option drops the data associated with an object. It also drops the data from lower level objects associated with the object being processed. This option does not reload the data.
- The Process Structure option will create only cube definitions for previously processed cubes. For unprocessed cubes, Analysis Services will process all the cube's dimensions that require processing.

Processing OLAP Objects



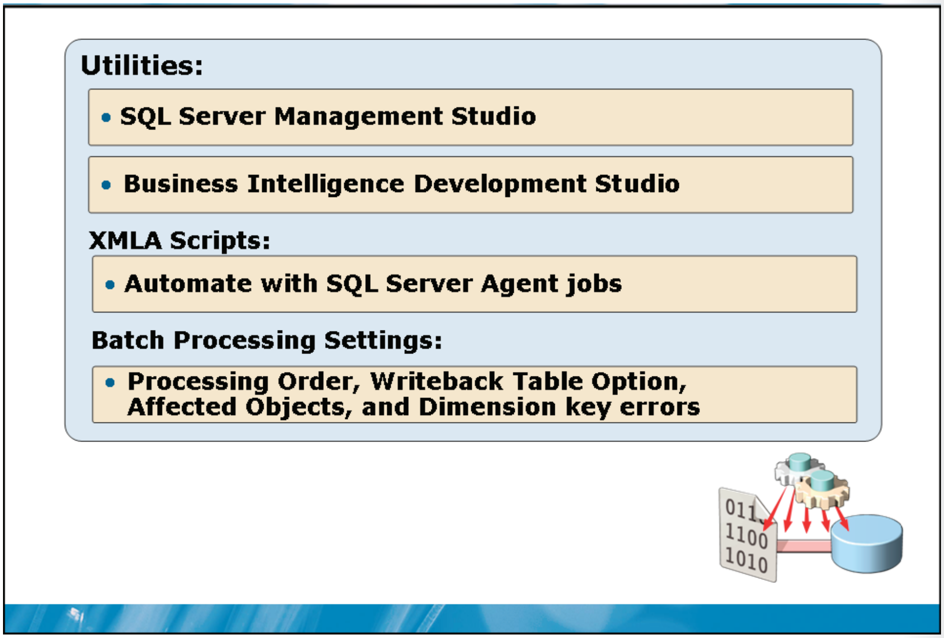
Key Points

You can use several tools and options to process OLAP objects.

- In SQL Server Management Studio, you can use Object Explorer to process objects.
- In Business Intelligence Development Studio, you can use both Solution Explorer and the individual Designers to process objects. For example, you can select a cube in Solution Explorer, right-click, and then click Process, or you can click the Process icon on the Cube Structure tab of the Cube Designer.
- You can also process objects by using XMLA Scripts. You can use the scripting option in SQL Server Management Studio to create and save processing scripts to use in the future.

Question: Which tool is used most frequently during development?

Implementing Batch Processing



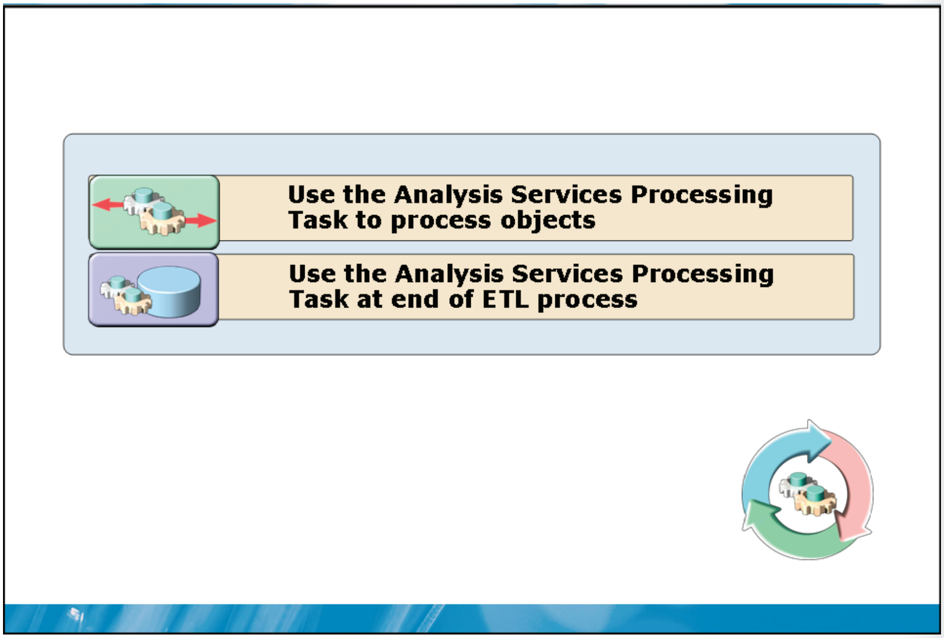
Key Points

When you need to process several objects at once, Analysis Services provides options for managing batch processing. You can define the order in which the objects process as well as whether they run as independent jobs or as part of a transaction.

- You can use both SQL Server Management Studio and Business Intelligence Development Studio to define batch processing. These two utilities provide similar functionality.
  - In Business Intelligence Development Studio, you must first deploy a project before you process objects.
- You can use XMLA scripts to implement batch processing. Additionally, you can create SQL Server Agent jobs to automate batch processing.

Question: Under what circumstances would you want to create a batch processing job?

Using SSIS to Process Analysis Services Objects



Key Points

You can use SQL Server Integration Services (SSIS) to process Analysis Services Objects as part of a package used to update data in the relational data source.

- The Analysis Services Processing task is available in Integration Services. This task will allow processing to run in batches.
- The Analysis Services Processing task is commonly found at the end of an Extract, Transform, and Load (ETL) process that includes some type of bulk insertion of data into the relational data source. You can use the Analysis Services Processing task to make this new data available in the Analysis Services database.

Question: What are the advantages of using SSIS to process an object?

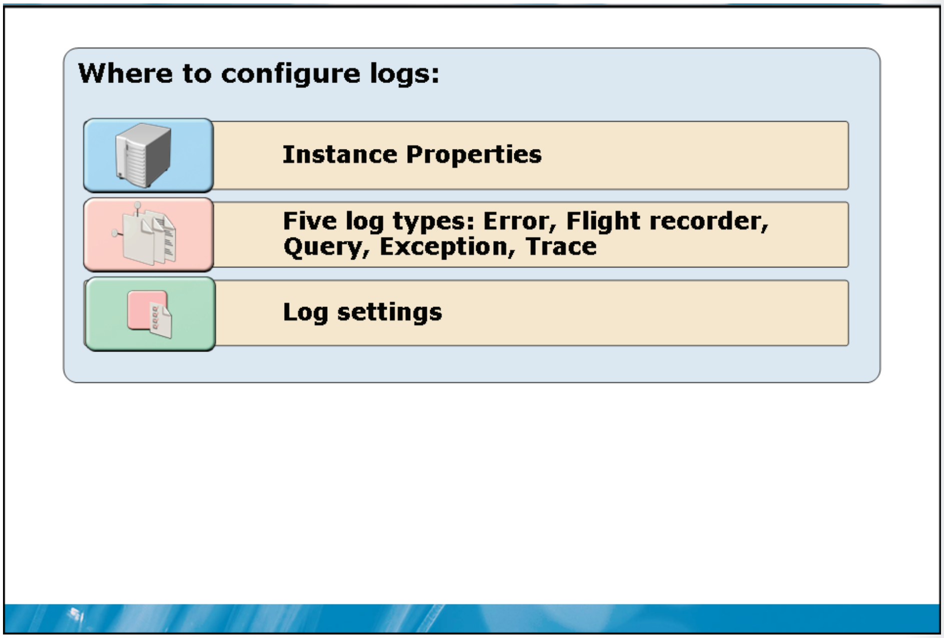
Lesson 2:
Logging, Monitoring, and Optimizing an Analysis Services Solution

- Configuring Logging
- Monitoring Analysis Services with SQL Server Profiler
- Monitoring Analysis Services with System Monitor
- Optimizing Aggregations

You can monitor the performance of SQL Server Analysis Services by using Logging and SQL Server Profiler or Performance, an application sometimes referred to as PerfMon. SQL Server Profiler provides you a window displaying the tasks that the server is performing during processing and query resolution. Performance gives you a view of product status as indexed through certain counters.

This lesson describes how to log, monitor, and optimize an Analysis Services solution.

Configuring Logging



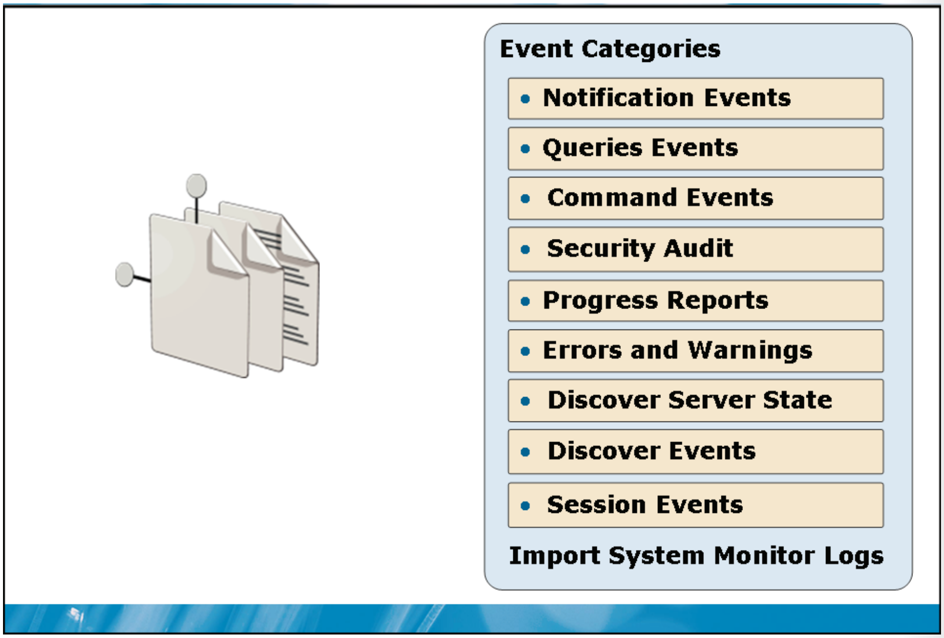
Key Points

Analysis Services provides logging options to track activity on each server instance.

- You can manage logging settings on the properties page of an Analysis Services instance in SQL Server Management Studio.
- Analysis Services includes the following types of logs:
  - **Error**. This log maintains errors that are configured and raised during processing and other operations.
  - **Flight recorder**. This log is a short-term log that tracks all activity on an Analysis Services instance.

- **Query**. This log records statistical information about running queries on an Analysis Services instance.
- **Exception**. This log should be used only with guidance from Microsoft Support.
- **Trace**. This log should be used only with guidance from Microsoft Support.

Monitoring Analysis Services with SQL Server Profiler



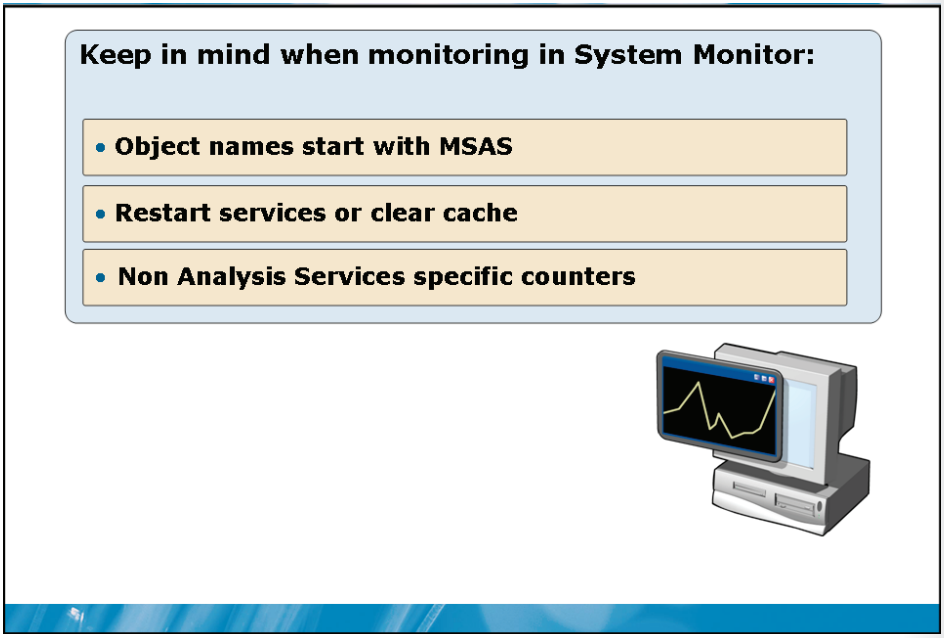
Key Points

Analysis Services provides events, event classes, and event categories in SQL Server Profiler to test the functionality and performance of MDX queries. SQL Server Profiler can capture data in a production environment. You can replay traces in a test environment to test and optimize changes to the system.

- SQL Server Profiler includes the following Analysis Services event categories:
  - **Notification events**. These provide helpful information for troubleshooting update problems in a real-time storage scenario.
  - **Query events**. These provide statistics on queries, including the start time, end time, processor time, and request parameters.
  - **Command events**. These provide statistics on commands, including the start time, end time, textdata, and EventSubclass event columns.

- **Security Audit events**. These provide auditing information on security events, including Audit Backup/Restore Event, Audit Login, Audit Logout, Audit Object Permission Event, and Audit Server Starts and Stops event classes.
- **Progress Reports events**. These provide statistics about progress reports. The Progress Report event category includes the Progress Report Begin, Progress Report End, Progress Report Current and Progress Report Error event classes.
- **Errors and Warnings events**. These report all error events since the trace began. The Error event class is the only event class in the Errors and Warnings event category.
- **Discover Server State events**. These report server-state XMLA discover events. The Discover Server State category includes the Server State Discover Begin, Server State Discover Data, and Server State Discover End classes.
- **Discover events**. These report information about XMLA Discover events. The XMLA Discover method retrieves information, such as a list of available databases, or detail about a specific object. The Discover Events category includes the Discover Begin and Discover End classes.
- **Session events**. These provide information about all sessions on the Analysis Services instance since the trace began. The Session Events category includes the Existing Connection, Existing Session, and Session Initialize event classes.

Monitoring Analysis Services with System Monitor



Key Points

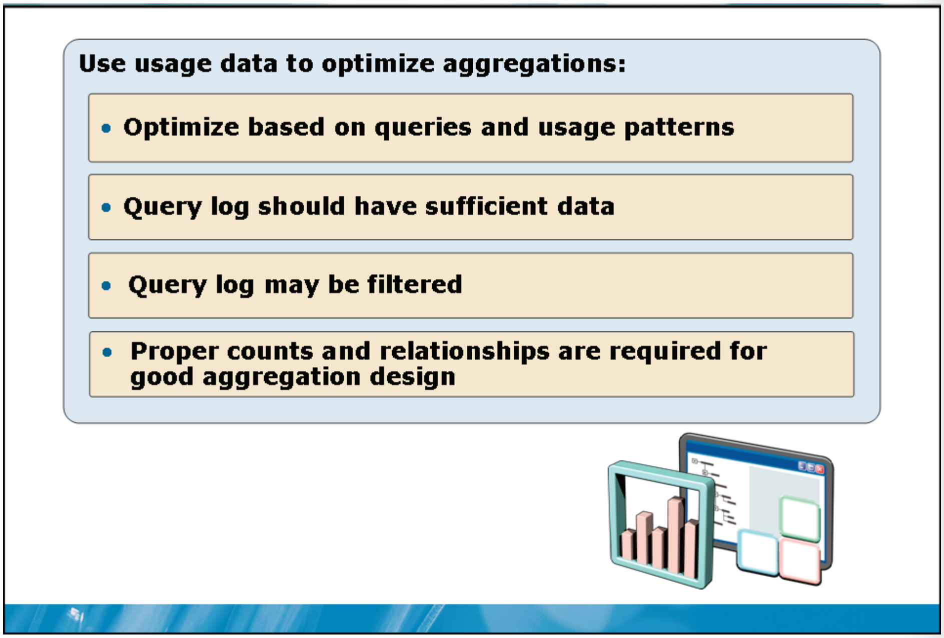
In addition to the standard hardware and operating system objects and counters, a large number of Analysis Services specific objects and counters exist with SQL Server 2008.

- There are a total of thirteen objects specifically designed for Analysis Services. Each of the objects that follow include a number of counters:
  - MSAS 2008: Cache
  - MSAS 2008: Connection
  - MSAS 2008: Data Mining Model
  - MSAS 2008: Data Mining Predictions
  - MSAS 2008: Locks
  - MSAS 2008: MDX
  - MSAS 2008: Memory

    - MSAS 2008: Proactive Caching
    - MSAS 2008: Proc Aggregations
    - MSAS 2008: Proc Indexes
    - MSAS 2008: Processing
    - MSAS 2008: Storage Engine Queries
    - MSAS 2008: Threads
- In addition to the Analysis Services specific counters, other objects such as Memory, Physical Disk, and Processor will also be helpful in optimizing and troubleshooting Analysis Services performance.

Question: What is the difference between a trace and a counter?

Optimizing Aggregations



Key Points

After an Analysis Services instance has been in production for some time, the optimization scenario that is being used for aggregations should be reviewed to see if it matches actual usage patterns. The Usage-Based Optimization Wizard allows administrators to base aggregation design on queries that occur on the server.

- The Usage-Based Optimization Wizard reads the information in the query log and will design the aggregations based on the usage patterns in the log.

Question: How can you ensure that the Usage Based Optimization Wizard bases optimizations on realistic usage information?

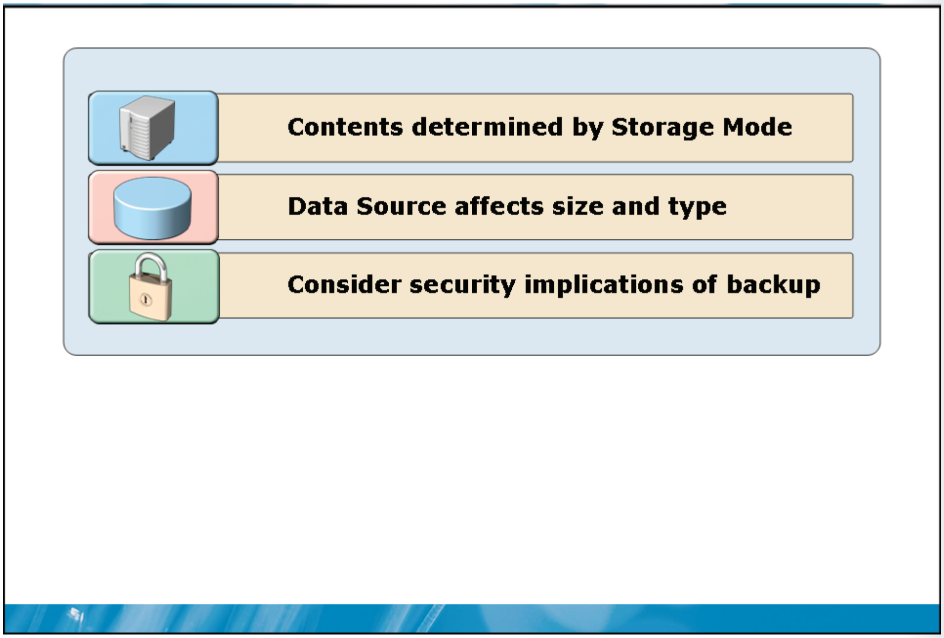
Lesson 3:
Backing Up and Restoring an Analysis Services Database

- Considerations for Backup and Restore
- Backing Up an Analysis Services Database
- Restoring an Analysis Services Database

Backing up lets administrators save a particular state of a SQL Server Analysis Services database and its objects. Restoring lets administrators restore an Analysis Services database to a previous state. The reasons for doing backups and restores include data recovery and preparation for audits.

In this lesson, you will learn how to backup and restore an Analysis Services database.

Considerations for Backup and Restore



Key Points

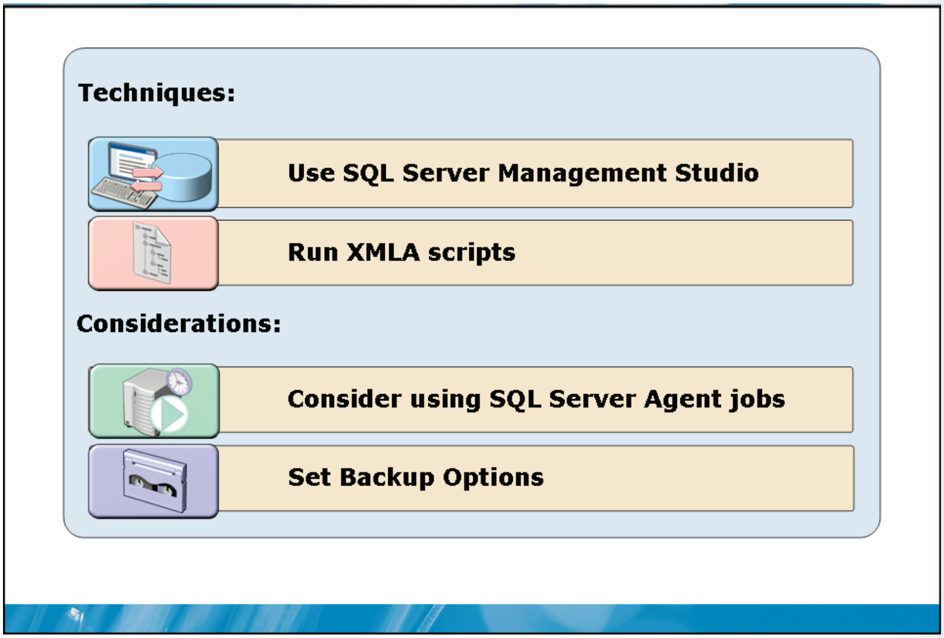
In any environment, it is important to be able to back up and restore your data. Analysis Services provides a backup and a restore command to protect the multidimensional data and structure.

- The Storage mode of your cubes and dimensions will affect what is backed up.
- You should also perform a backup of the relational data source on a regular basis. This is especially true for the ROLAP and HOLAP storage modes.
- Security considerations include the following:
  - The user that runs the backup command must have the appropriate file system permissions to write to the backup location. They must also either be a member of the Analysis Services server role, or have Full Control (Administrator) permissions on the database that they are backing up.
  - To encrypt a backup file, a password must be defined when the backup is created. This same password must be provided when the encrypted backup is restored.

- The user that runs the restore command must have the appropriate file system permissions to read from the location of the backup file. They must also be a member of the Analysis Services server role if the restore command creates a new database file. They can also be a member of a role with Full Control (Administrator) permissions on the database to be overwritten during the restore.

Question: What storage modes are used in your organization?

Backing Up an Analysis Services Database

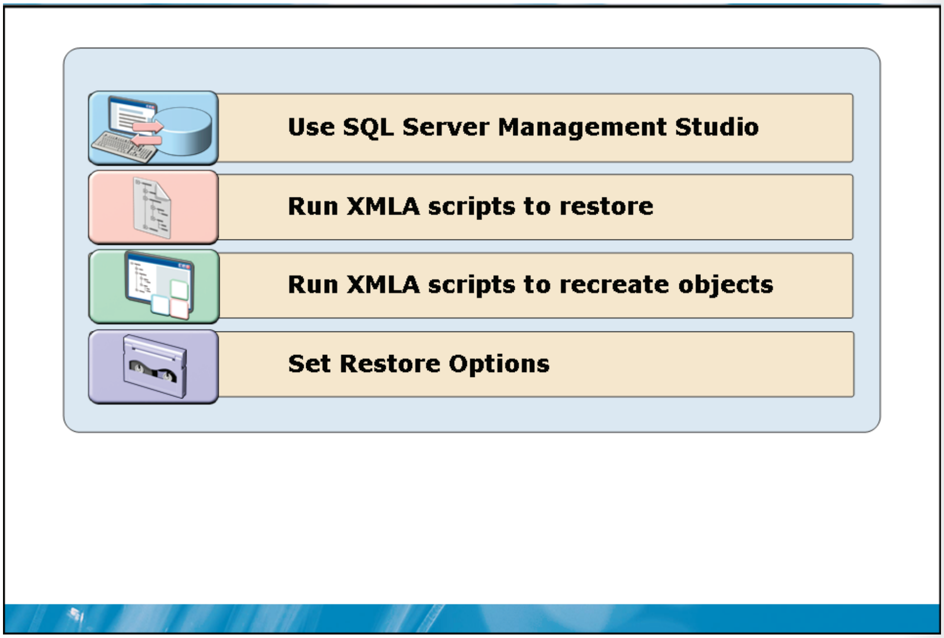


Key Points

Analysis Services provides several options to create backups.

- To back up a database by using SQL Server Management Studio, use Object Explorer to browse to the database that you want to back up, right-click that database, and then click Backup.
- You can automate Analysis Services backups by creating jobs that are run by the SQL Server Agent. You should define the job step as the SQL Server Analysis Services Command type, and you should include an XMLA backup script as the command.

Restoring an Analysis Services Database



Key Points

Similar to the backup process, there are several methods available to restore a database.

- To restore a database by using SQL Server Management Studio, use Object Explorer to browse to either the Databases folder or the database that you want to restore, right-click the object, and then click Restore.
- Analysis Services also provides the capability to create XMLA scripts to create the objects in your Analysis Services database. When the XMLA script is run, it will recreate the metadata only. You will then need to process the database to import data and build aggregations before users can query the database. You can use a script to recreate the metadata of a single object that was corrupted or lost.

- Additional restore options exist in both the Restore window in SQL Server Management Studio and the XMLA restore element that will define whether partitions should be restored to their original location or to a new location, as well as how remote partitions should be restored.

Question: How might you use automated backup and restore in a staging or test environment?

Lab: Maintaining a Multidimensional Solution

- Exercise 1: Configuring Processing
- Exercise 2: Implementing Logging and Monitoring
- Exercise 3: Backing Up and Restoring an Analysis Services Database

Logon information

Virtual machine	NY-SQL-01
User name	Administrator
Password	Pa$$w0rd

Estimated time: 60 minutes

Exercise 1: Configuring Processing

Scenario

You have been developing an OLAP solution to demonstrate the capabilities of Analysis Services to your company. You need to configure processing of cube data.

This exercise's main tasks are:

1. Start the 6234A-NY-SQL-01 virtual machine and log on as Student.
2. Install the Adventure Works Data Warehouse database and setup the lab environment.
3. Write a script to perform a full process of the Reseller Sales partition.
4. Update the data in the relational Reseller fact table and then perform a full process.

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Student

- Start 6234A-NY-SQL-01, and log on as **Student** with the password of **Pa$$w0rd**.

▶ Task 2: Install the Adventure Works Data Warehouse database and setup the lab environment

- Open **E:\Mod08\Labfiles\Starter\AdventureWorks Data Warehouse**, click **instawdwdb.sql** in **SQL Server Management Studio** and execute it to install the Adventure Works Data Warehouse database.
- Run **Setup.cmd** in the **E:\Mod08\Labfiles\Starter** folder.

▶ Task 3: Write a script to perform a full process of the Reseller Sales partition

- In SQL Server Management Studio, create an XMLA query to perform a full process on the **Reseller Sales** partition.
- Save the query to **E:\Mod08\Labfiles\Starter\ProcessPartition.xmla**.

▶ Task 4: Update the data in the relational Reseller fact table and then perform a full process

- Use SQL Server Management Studio to run the MDX query in **E:\Mod08\Labfiles\Starter\MDXQuery.mdx**.
    - This query displays the current aggregated value for the **Reseller Sales – Sales Amount** measure.
- Run the Transact-SQL query in **E:\Mod08\Labfiles\Starter\InsertNewFact.sql** to add a new row to the **FactResellerSales** table.

- Run the MDX query again and verify that aggregated value for the **Reseller Sales – Sales Amount** measure has not changed.
- Run the **ProcessPartition.xmla** script you created in the previous step to process the partition.
- Run the MDX query again and verify that aggregated value for the **Reseller Sales – Sales Amount** measure has been updated.

Results: After this exercise, you should have successfully created, reviewed and executed the ProcessPartition.xmla script.

Exercise 2: Implementing Logging and Monitoring

Scenario

You have been developing an OLAP solution to demonstrate the capabilities of Analysis Services to your company. You need to monitor your solution for performance problems and errors.

This exercise's main tasks are:

1. Configure Logging for an Analysis Services instance.
2. Simulate activity in the database.
3. Run usage-based optimization for the Internet Sales partition.

▶ Task 1: Configure Logging for an Analysis Services instance

- Use SQL Server Management Studio to edit the following properties of the NY-SQL-01 Analysis Services instance:
    - Set the **Log\QueryLog\CreateQueryLogTable** property to **true**.
    - Set the **Log\QueryLog\QueryLogConnectionString** property to connect to NY-SQL-01 by using **Windows Authentication** and connect to the **AdventureWorksDW** database.
    - Set the **Log\QueryLog\QueryLogSampling** property to **2**.

Note: Setting a QueryLogSampling rate of one in every two queries is too frequent for most production environments. This setting is used in the lab to overcome the low query volume on a test server.

- Restart the **Analysis Services** service.

▶ Task 2: Simulate activity in the database

- Use SQL Server Profiler to open and play back the **E:\Mod08\Labfiles\Starter\playbacktrace.trc** trace file.
- Close SQL Server Profiler when the replay has completed.

▶ Task 3: Run usage-based optimization for the Internet Sales partition

- In SQL Server Management Studio, use **Object Explorer** to browse to the **Internet Sales** partition in the **Adventure Works OLAP** database.
- Use the **Usage Based Optimization Wizard** to design aggregations based on the queries in the query log table. Optimize the aggregations to reach a 30 percent performance gain.

Results: After this exercise, you should have successfully configured logging, created activity in the database, and run usage-based optimization for a partition.

Exercise 3: Backing Up and Restoring an Analysis Services Database

Scenario

You have been developing an OLAP solution to demonstrate the capabilities of Analysis Services to your company. Before bringing the Analysis Services solution into your production environment, you need to successfully back up and restore the database for disaster recovery purposes.

This exercise's main tasks are:

1. Back up the Adventure Works OLAP database.
2. Create a restore script for the Adventure Works OLAP database.
3. Test a restore of the Adventure Works OLAP database.

▶ Task 1: Back up the Adventure Works OLAP database

- Use SQL Server Management Studio to back up the database.
- Back up the database to **E:\Mod08\Labfiles\Starter\Adventure Works OLAP.abf**.
- Use the **Allow file overwrite** option.
- Clear the **Encrypt backup file** option.

▶ Task 2: Create a restore script for the Adventure Works OLAP database

- Use the script option in the **Restore** window to create an XMLA script that will restore the Adventure Works OLAP database from the backup you created in the previous task.
- Select the **Allow database overwrite** option.
- Save the script to **E:\Mod08\Labfiles\Starter\AWOLAPRestore.xmla**.

▶ Task 3: Test a restore of the Adventure Works OLAP database

- In **Object Explorer**, delete the **Adventure Works OLAP** database.
- Use the **AWOLAPRestore.xmla** script created in the previous step to restore the **Adventure Works OLAP** database.
- Use **Object Explorer** to confirm the database has been restored successfully.
  - You might need to refresh the view to see the restored database.

> **Results:** After this exercise, you should have successfully backed up an Analysis Services database, created a restore script for the database, and tested restoring the database.

Note: The answers to the practices and labs are on the Student Materials CD.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-01 virtual machine and discard any changes.

Module Review and Takeaways

- Review Questions
- Common Issues and Troubleshooting Tips

Review Questions

1. What is processing?
2. Describe various scenarios in which organizations may benefit from optimizing an Analysis Services database.
3. Which processing methods can be used with scripting?
4. How does the storage mode affect backup and restore?

Common Issues and Troubleshooting Tips

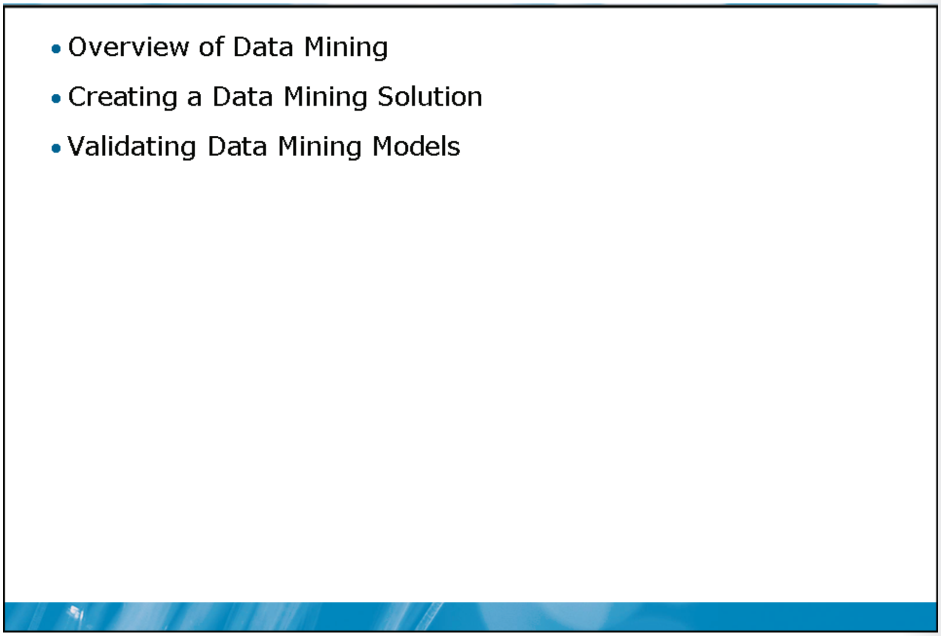
SQL Server Profiler is a very powerful tool for troubleshooting and optimizing Analysis Services.

Module 9

Introduction to Data Mining

Contents:

Module Overview



Data mining gives you access to the information that you need to make intelligent decisions about difficult business problems. Microsoft® SQL Server® Analysis Services provides tools for data mining with which you can identify rules and patterns in your data, so that you can determine why things happen and predict what will happen in the future. This module discusses the concepts of data mining, how to create a data mining solution, and how to validate data mining models.

Lesson 1: Overview of Data Mining

- What Is Data Mining?
- Data Mining Concepts
- Data Mining Algorithms

Data mining is frequently described as "the process of extracting valid, authentic, and actionable information from large databases." In this lesson, you will learn how to describe data mining, the scenarios and concepts associated with data mining, and the algorithms used to build prediction models.

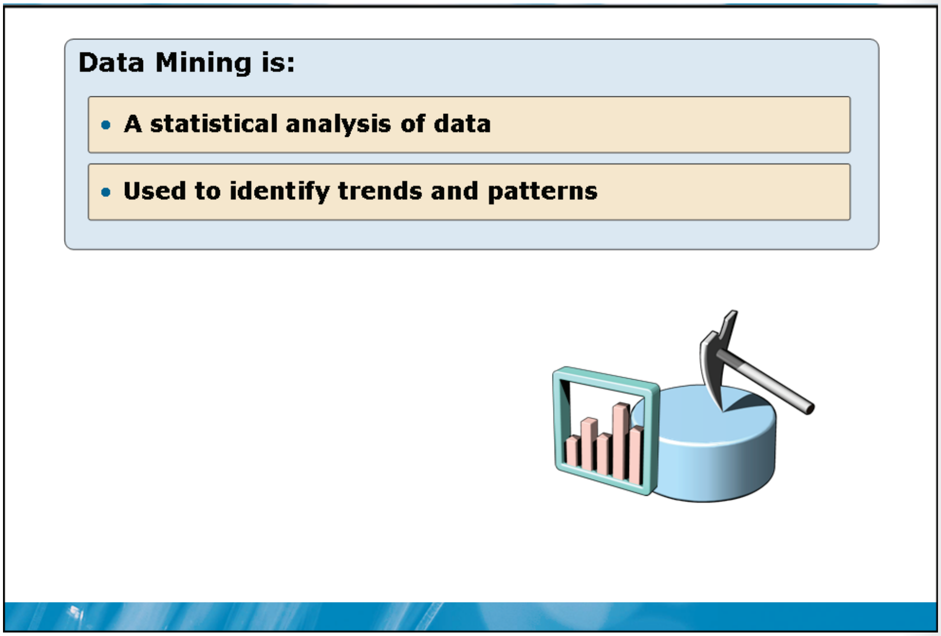
Discussion: Why Mine Data?

- Forecasting Sales
- Targeted Advertising
- Credit Ratings

Key Points

Discuss the data mining scenarios with the classroom, led by your instructor.

What Is Data Mining?



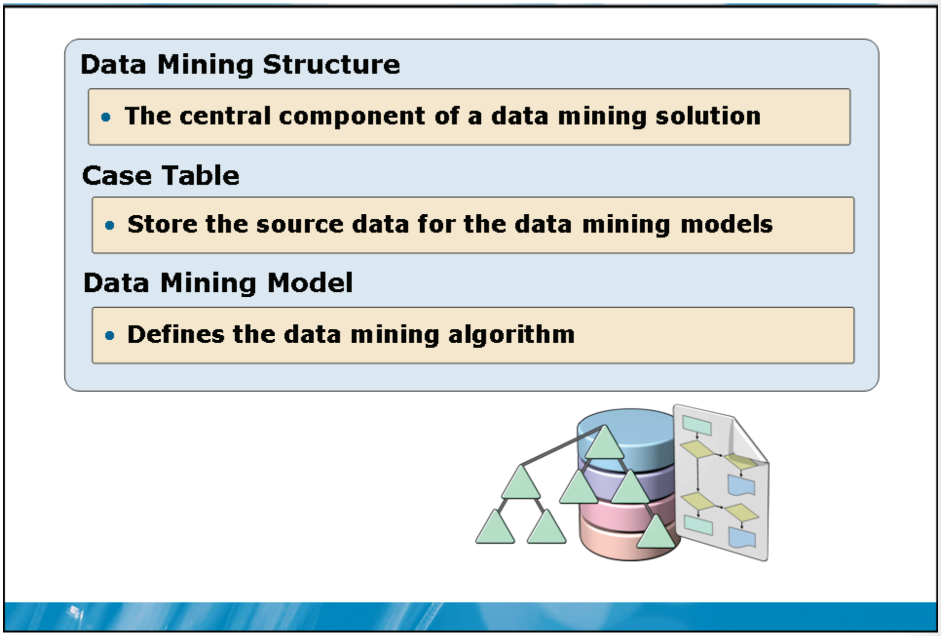
Key Points

Data mining is the process of searching through data to extract patterns and trends by using various algorithms. For example, a retail organization might use data mining to determine whether there is a correlation between location and unit price. They might discover that customers in large urban areas are more likely to buy goods that are in a high price range, and that rural customers are more likely to buy low-priced goods.

- Data mining statistically analyzes your data and can use large volumes of data that would be very difficult to manually analyze. Data mining enables business users to access information that would otherwise only be available to statisticians.
- Data mining can identify trends and patterns that may not be immediately obvious. You can use data mining to predict unknown values based on statistics and patterns in previously analyzed sets of data. This is useful when you are trying to predict and plan for future events.

Question: Which of the following is an example of data mining: querying an address directory to find people with a particular last name in a zip code; aggregating the counts of last names by zip code; or predicting the zip codes that might contain a higher concentration of a particular last name?

Data Mining Concepts



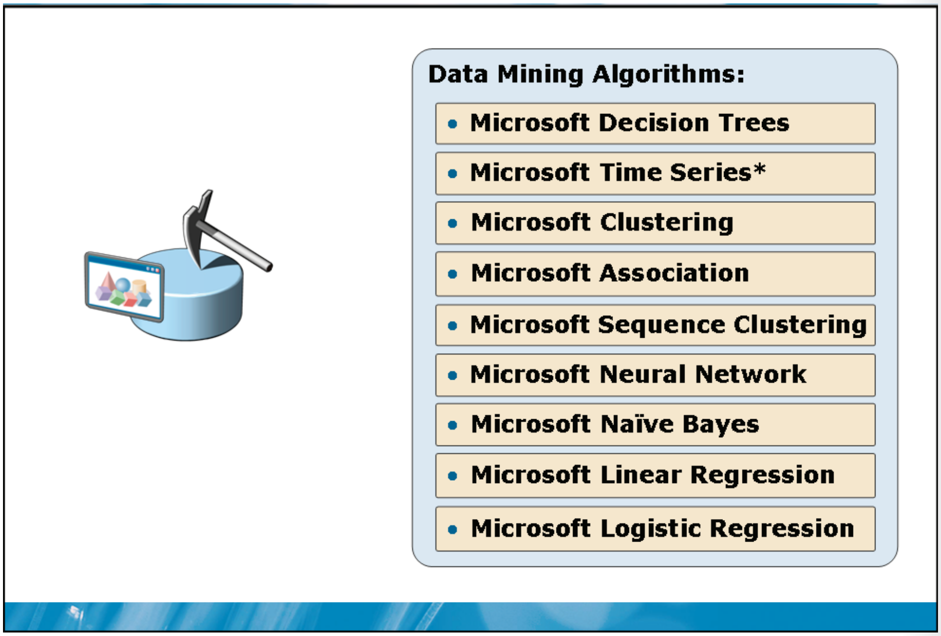
Key Points

A case table and one or more data mining models form a data mining structure. The data mining structure defines each data mining solution.

- The *data mining structure* is the central component of a data mining solution.
- The *case table* stores the source data for the data mining models.
- The *data mining model* defines the data mining algorithm. It also defines which columns the algorithm uses from the case table, and whether each column is an input column, a key column, or a predictable column.

Question: Where is the training data stored?

Data Mining Algorithms



Key Points

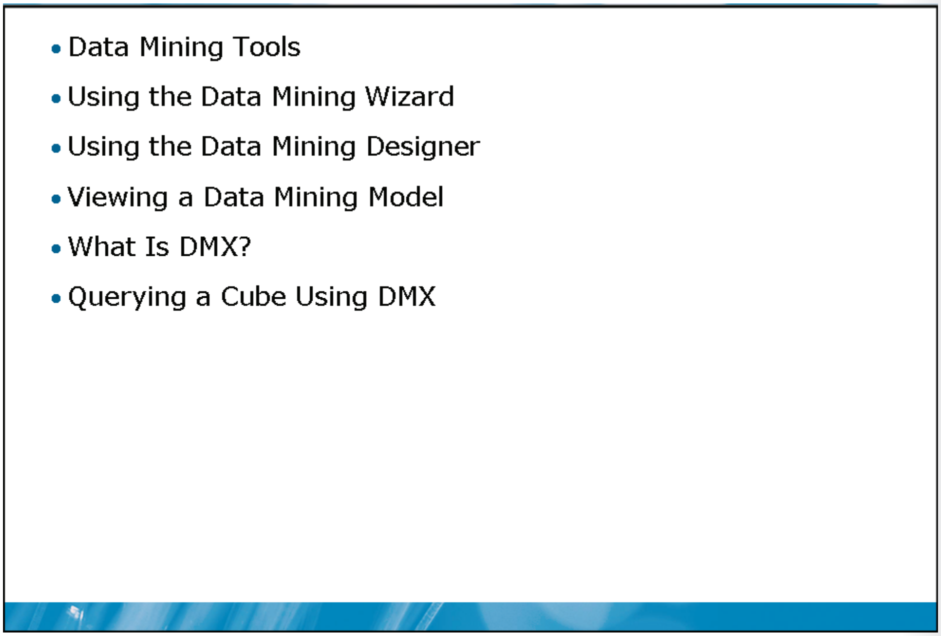
Data mining algorithms provide the rules for the analysis of data. Analysis Services provides nine algorithms, and you can also add further third-party algorithms if necessary.

Algorithms that are in wide use fall into the following broad categories:

- Classification algorithms predict one or more discreet variables based on other attributes. Microsoft Decision Trees is an example that might be used predict whether a customer will purchase a particular product. Microsoft Neural Network and Microsoft Naïve Bayes are also classification algorithms.
- Regression algorithms predict one or more continuous variables, such as profit or loss. The Microsoft Time Series algorithm is an example that might be used to determine seasonal sales for the upcoming year for a retail store. Microsoft Linear Regressions and Microsoft Logistic Regression are also regression algorithms that can predict continuous variables.

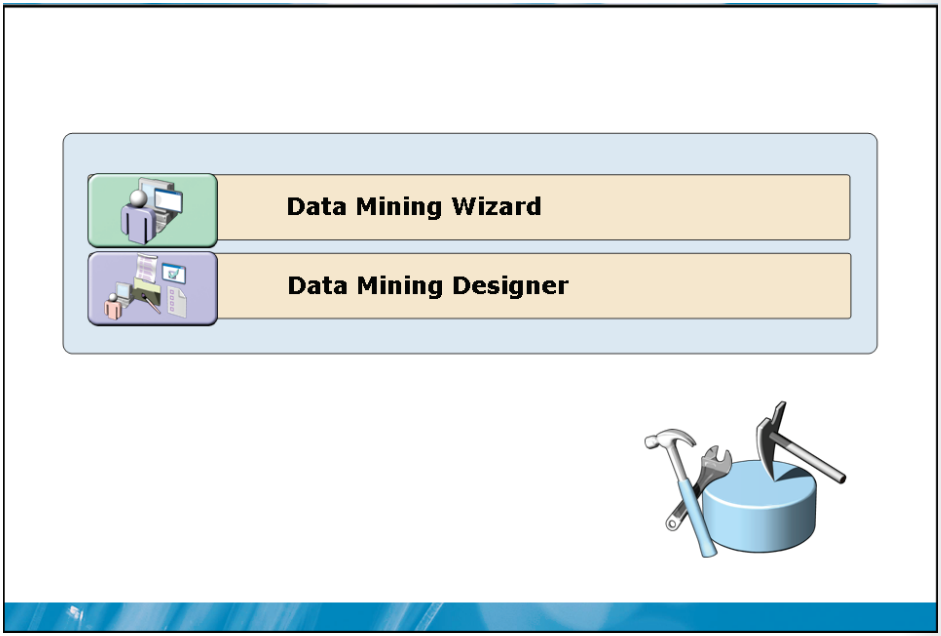
- Segmentation or Clustering algorithms divide data into groups or clusters of items that have similar properties. Microsoft Clustering is an example that might be used to divide customers into groups with similar purchasing habits or preferences.
- Association algorithms find correlation between different attributes in a data set. The Microsoft Association algorithm is an example that might be used to describe which items are likely to appear together in a transaction, such as products that might be purchased together.
- Sequence analysis finds common sequences in data. The Microsoft Sequence Clustering algorithm is an example that might be used to find common web click-through paths or order of placing items in a cart.

Lesson 2: Creating a Data Mining Solution



When you create a data mining solution in Analysis Services, you first create a model that describes your business problem, and then you run your data through an algorithm that generates a mathematical model of the data, a process that is known as training the model. You can then either visually explore the mining model or create prediction queries against it. Analysis Services can use datasets from both relational and OLAP databases, and includes a variety of algorithms that you can use to investigate that data. In this lesson, you will learn how to create a data mining structure with multiple data mining models.

Data Mining Tools



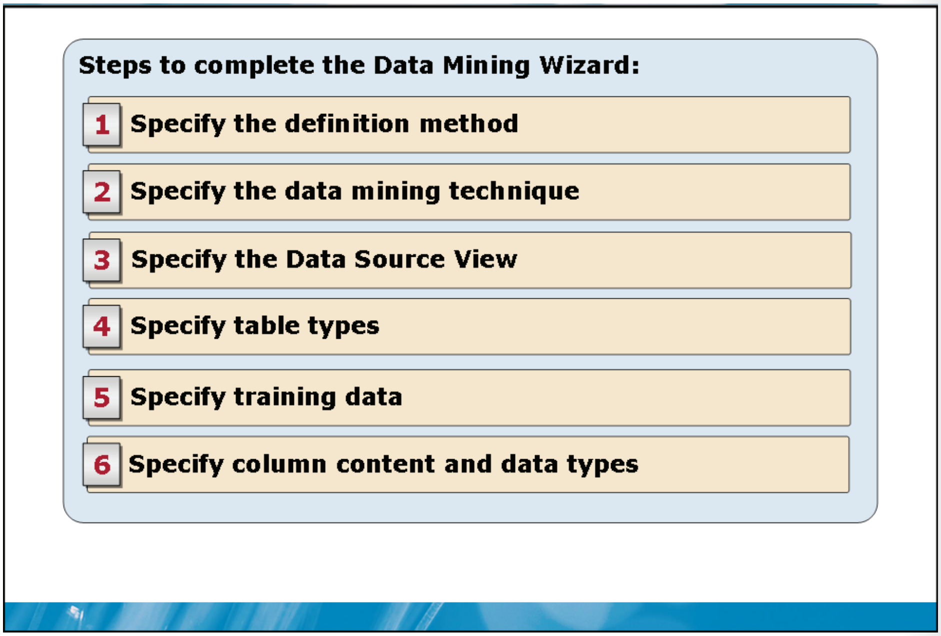
Key Points

In Business Intelligence Development Studio, use the **Data Mining Wizard** to create new data mining structures, and then use the **Data Mining Designer** to configure the structure.

- Use the *Data Mining Wizard* to define the data mining structure and create the first data mining model for the structure. The Data Mining Wizard starts automatically when you create a new data mining structure.

- Use the *Data Mining Designer* to configure the data mining structure and data mining model that you created with the Data Mining Wizard. You can also use it to add new data mining models, train, browse, and compare models, and create predictions based on mining models.

Using the Data Mining Wizard



Key Points

The Data Mining Wizard is the tool used to generate your initial data mining structure.

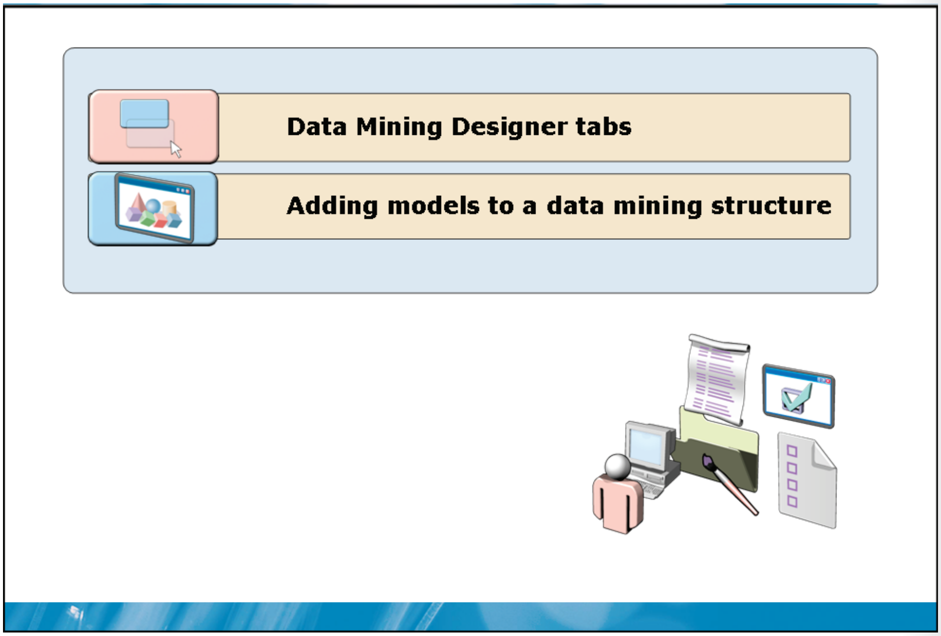
Use the following steps to complete the **Data Mining Wizard**.

1. First, you must select the **Definition Method**. For example, you might choose **From existing relational database or data warehouse**.
2. Next, select the **Data Mining Technique**. For example, you might choose **Microsoft Decision Trees**.
3. Select the **Data Source View** that will access the training data. For example, you might choose **Adventure Works DM**.
4. Next, specify the **Table Types**. For example, you might select a view named **vTargetMail** as the case table.

5. On the next page, specify the **Training Data**. For example, you might select **CustomerKey** as the Key column, select **BikeBuyer** as Input and Predictable, and select **CommuteDistance**, **NumberCarsOwned**, **Region**, and **YearlyIncome** as Input columns.
6. Next, specify **Columns Content** and **Data Type**. Analysis Services can attempt to detect the content and data type, but you should check the results. For example, click **Detect** to perform an automatic detection and then change all columns to a **Discrete** data type except **YearlyIncome**, which should be **Continuous**, and **CustomerKey**, which should be **Key**.
7. Finally, specify the name of the structure and the model, and whether to allow drill-through. For example, name the structure **Targeted Mailing** and name the model **TM_Decision_Tree**, and then select the **Allow drill-through** check box.

Question: What is the difference between a discreet value and a continuous value?

Using the Data Mining Designer

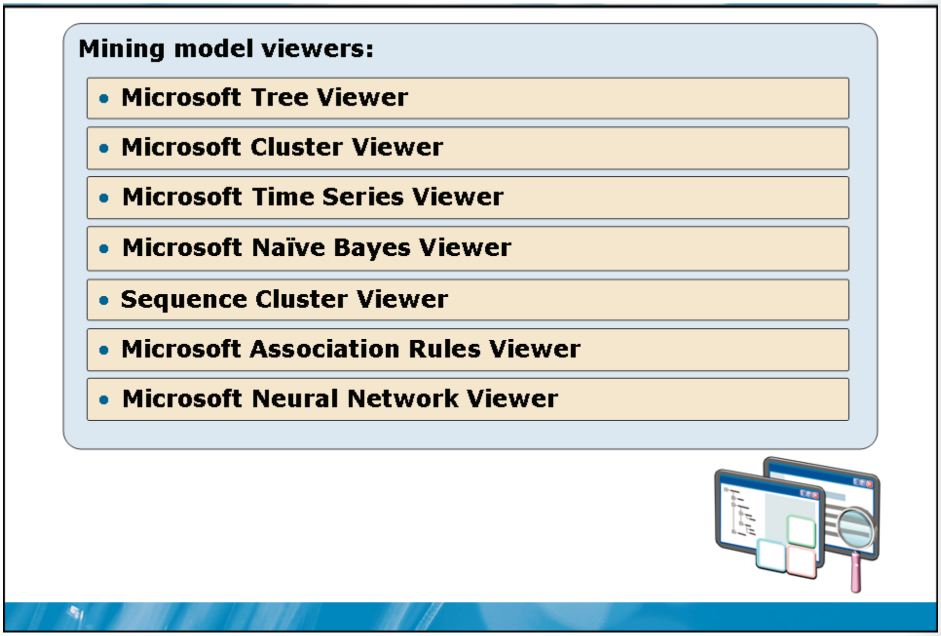


Key Points

Use the Data Mining Designer to edit the mining structure, add and edit mining models, test accuracy, and view model data.

- To add models to a data mining structure, use the following procedure:
  - On the Mining Models tab, click the **Create a related mining model** button or right-click an existing mining model, and then click **New Mining Model**.
  - Specify a **Name** and select an **Algorithm** for the new mining model.

Viewing a Data Mining Model

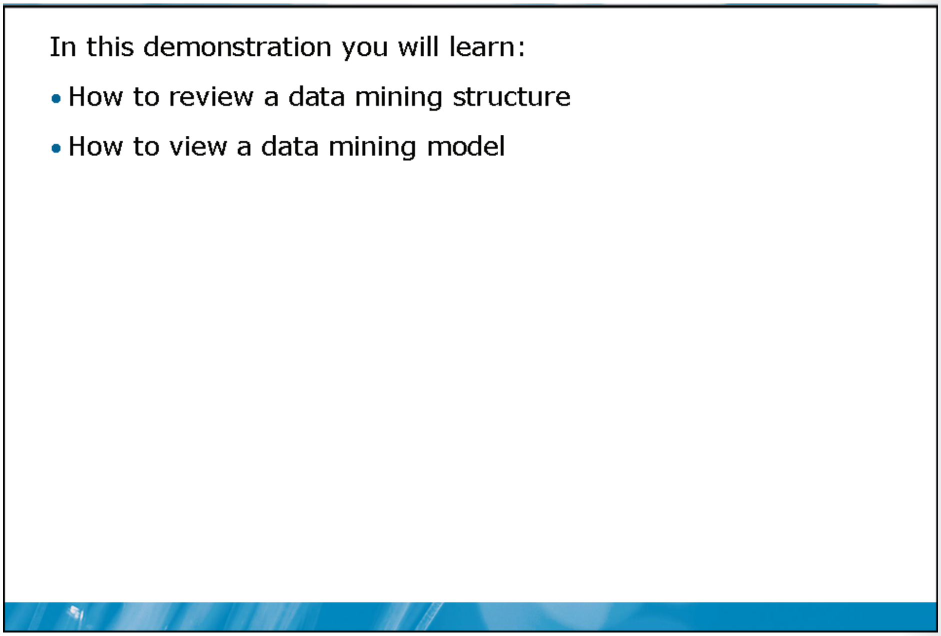


Key Points

Each data mining model has its own viewer and each viewer has multiple views of the results.

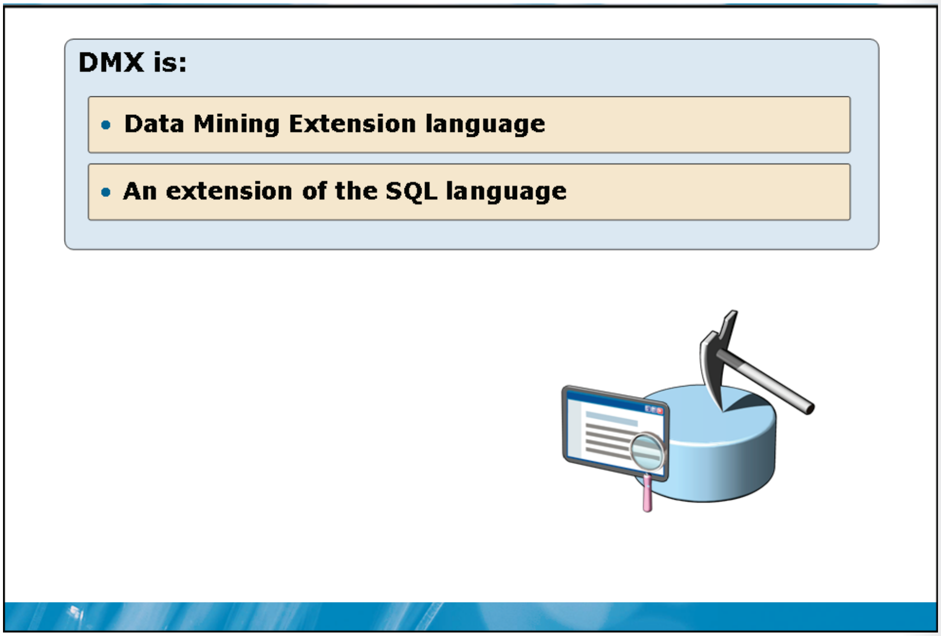
Question: Why does each algorithm have its own viewer?

Demonstration: Using Data Mining



Question: If you wanted to use an algorithm to predict the types of products that customers are likely to purchase, how might you use the settings in the viewer to help your design?

What Is DMX?

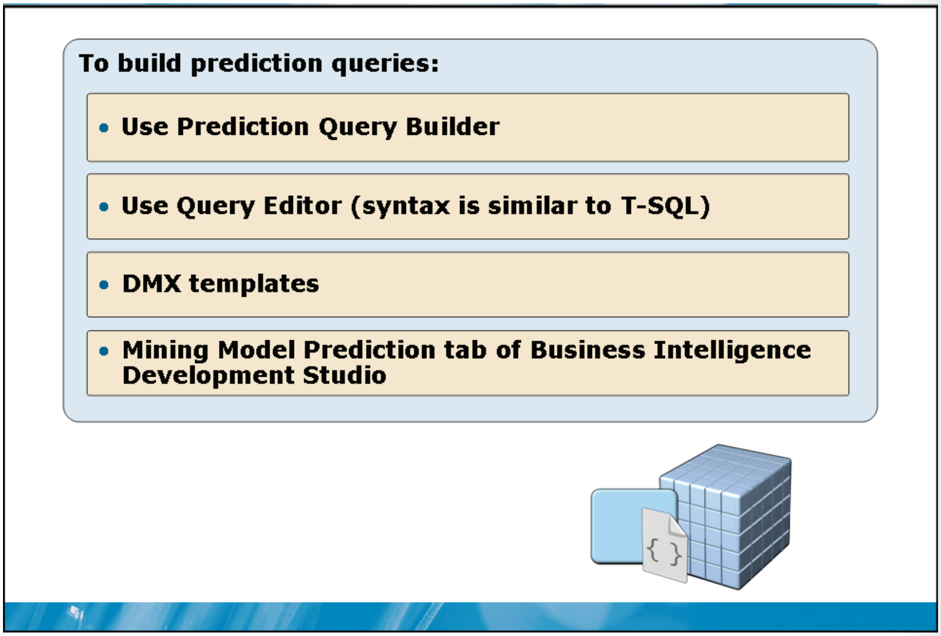


Key Points

Analysis Services provides a query language called Data Mining Extensions (DMX) that is the basis for creating predictions.

- You can use DMX to create the structure of new data mining models, to train these models, and to browse, manage, and predict against them. DMX is composed of data definition language (DDL) statements, data manipulation language (DML) statements, and functions and operators.
- You can use the result of any DMX query as the basis of a report, and can take advantage of the parameterization and formatting features that are available in Reporting Services.

Querying a Cube Using DMX



Key Points

You can use DMX statements to create, process, delete, copy, browse, and predict against data mining models. There are two types of statements in DMX: data definition statements and data manipulation statements. You can use each type of statement to perform different kinds of tasks.

- You can use the prediction query in Data Mining Extensions (DMX) to predict unknown column values based on the results of a mining model.
- To help you build DMX prediction queries, SQL Server provides a query builder, and DMX templates for the query editor in Management Studio.
- Within Business Intelligence Development Studio, you access the query builder from the Mining Model Prediction tab of Data Mining Designer.

Question: How might you use DMX prediction queries in a web application?

Lesson 3:
Validating Data Mining Models

- Overview of Data Mining Validation
- Accuracy Charts
- Viewing Accuracy Charts

Validation is the process of assessing how well your mining models perform against real data. It is important that you validate your mining models by understanding their quality and characteristics before you deploy them into a production environment.

This lesson introduces some basic concepts related to model quality, and introduces the strategies for model validation that are provided in SQL Server 2008 Analysis Services.

Overview of Data Mining Validation

- **Validate and Compare Mining Models**
- **Compare the results of the mining model to known data**
- **Display the accuracy of the models using accuracy charts**

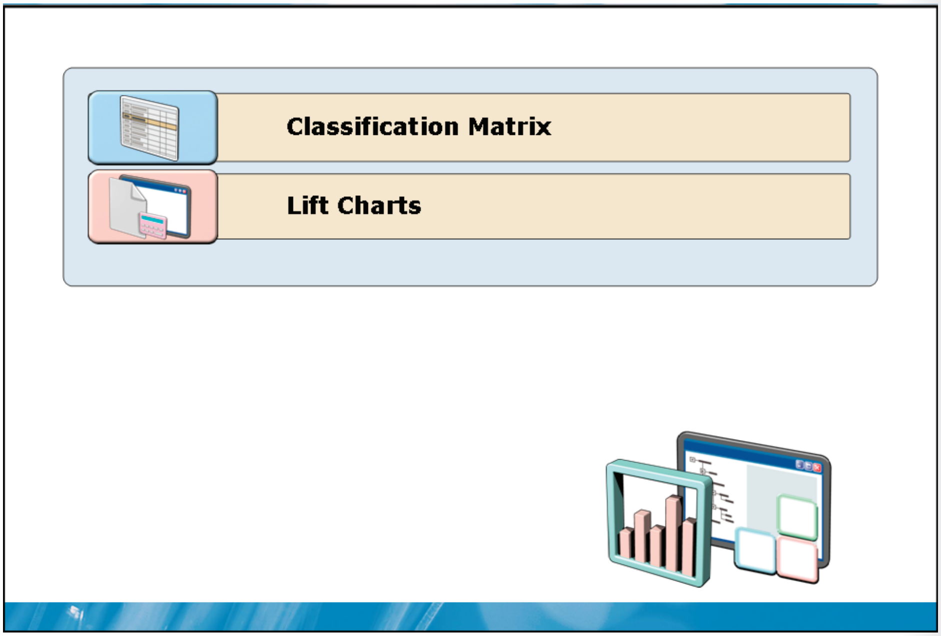
Key Points

Data mining algorithms use statistical models to predict the outcome of scenarios based on training data. Each algorithm uses different techniques and will generate different results.

- Because each algorithm generates different results, it is important to validate and compare algorithms to ensure that you are using the best model.
- To test the validity of a data mining model, you supply a known set of data, and then the validation tools compare the results of the mining model to this data.
- The accuracy of the models can be displayed by using accuracy charts. The accuracy charts compare predicted values with actual values.

Question: Why is data mining considered an iterative process?

Accuracy Charts



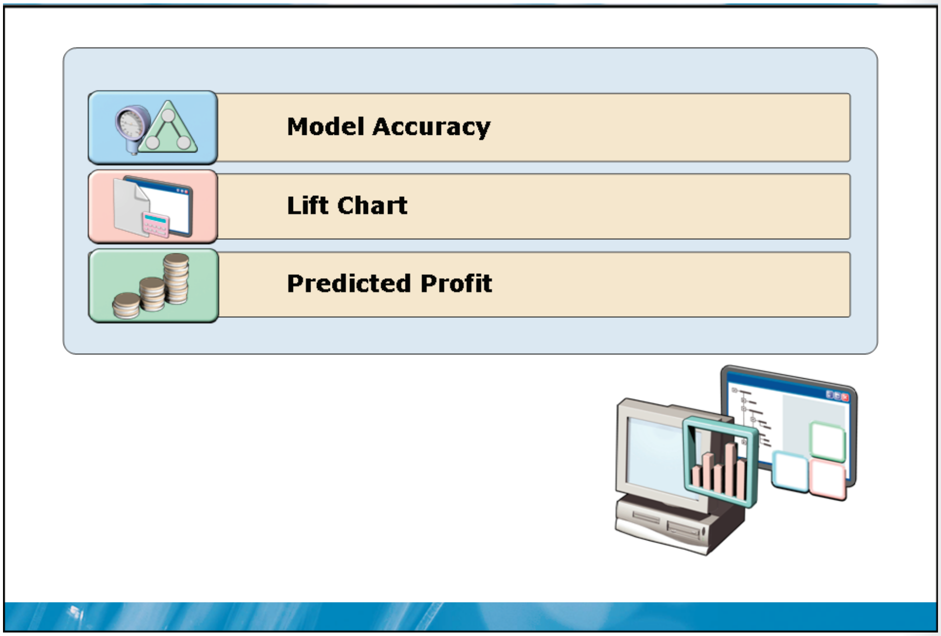
Key Points

The accuracy charts display the difference between predicted values and actual values. Some models are easier to test with a classification matrix, while others are easier to test with a lift chart. Typically, a classification matrix is better for discrete values and a lift chart is better for continuous values.

- A *classification matrix* compares actual values with predicted values. The matrix displays how often the model made an accurate prediction and also which inaccurate predictions the model made.
- *Lift charts* display the actual results, randomly generated results, and the results of one or more models as lines on a graph.

Question: Why use a chart to evaluate model accuracy?

Viewing Accuracy Charts



Key Points

You can view accuracy charts from the Mining Accuracy Chart tab of Data Mining Designer. Accuracy charts can test the accuracy, view the lift, and predict the potential profit of your models.

Lab: Implementing Data Mining

- Exercise 1: Creating a Data Mining Structure
- Exercise 2: Adding a Data Mining Model
- Exercise 3: Exploring Data Mining Models
- Exercise 4: Validating Data Mining Models

Logon information

Virtual machine	NY-SQL-01
User name	Administrator
Password	Pa$$w0rd

Estimated time: 60 minutes

Exercise 1: Creating a Data Mining Structure

Scenario

You have been asked to add additional features to an existing demonstration data mining solution. You will create a new data mining structure that will be used to forecast future sales of bike models in different regions. You will base the data mining structure on data from the AdventureWorksDW database.

This exercise's main tasks are:

1. Start the 6434A-NY-SQL-01 virtual machine and log on as Student.
2. Install the Adventure Works Data Warehouse database.
3. Browse the data mining project and data source view.
4. Create a data mining structure.
5. Modify a data mining structure.

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Student

- Start 6234A-NY-SQL-01, and log on as **Student** with the password of **Pa$$w0rd**.

▶ Task 2: Install the Adventure Works Data Warehouse database

- Open **E:\Mod09\Labfiles\Starter\AdventureWorks Data Warehouse**, click **instawdwdb.sql** in **SQL Server Management Studio** and execute it to install the Adventure Works Data Warehouse database.

▶ Task 3: Browse the data mining project and data source view

- In **Business Intelligence Development Studio**, open **E:\Mod09\Labfiles\Starter\Adventure Works Data Mining.sln**.
- Browse the **Adventure Works DM DSV** data source view to review the tables and views included in it.

▶ Task 4: Create a data mining structure

- Use the **Data Mining Wizard** to create a new data mining structure.
  - Use the Microsoft Time Series algorithm.
  - Build the structure from the existing **Adventure Works DM DSV**.
  - Use **vTimeSeries** as the case table.
  - The **ModelRegion** and **TimeIndex** columns should be key columns.

Note: Having a key that includes the TimeIndex and ModelRegion columns will only work if the TimeIndex column is unique across each individual value in the ModelRegion column.

  - The **Quantity** and **Amount** columns should be both input and predictable columns.
- Name the mining structure **Sales Forecasting**.
- Name the mining model **Time Series Forecast**

▶ Task 5: Modify a data mining structure

- Use the **Mining Models** tab of the **Sales Forecasting.dmm Designer** to set the algorithm parameters.
  - Set the **PERIODICITY_HINT** parameter to **{12}**. (Because your data represents months in a year, a period repeats once every 12 months.)

Results: After this exercise, you should have successfully browsed the data mining project and data source views, created a data mining structure, and modified a data mining structure.

Exercise 2: Adding a Data Mining Model

Scenario

You will now add two data mining models to the project.

This exercise's main tasks are:

1. Add a Naive Bayes mining model to the Targeted Mailing mining structure.
2. Add a Clustering mining model to the Targeted Mailing mining structure.

▶ Task 1: Add a Naive Bayes mining model to the Targeted Mailing mining structure

- Open the **Targeted Mailing.dmm Designer** in Business Intelligence Development Studio.
- In the **Targeted Mailing.dmm Designer**, add a related mining model that uses the **Microsoft Naive Bayes** algorithm.
    - The **Yearly Income** and **Age** columns will be ignored because the **Naive Bayes** algorithm does not support continuous columns.
- Name the model **TM_Naive_Bayes**.

▶ Task 2: Add a Clustering mining model to the Targeted Mailing mining structure

- In the **Targeted Mailing.dmm Designer**, add a related mining model that uses the **Microsoft Clustering** algorithm.
- Name the model **TM_Clustering**.

Results: After this exercise, you should have successfully added two data mining models to the project.

Exercise 3: Exploring Data Mining Models

Scenario

You must now deploy and explore the solution you have created.

This exercise's main tasks are:

1. Deploy the data mining solution.
2. View the data mining model in the Sales Forecasting mining structure.
3. View the data mining models in the Targeted Mailing mining structure.

▶ Task 1: Deploy the data mining solution

- Use Business Intelligence Development Studio to deploy your data mining solution.

▶ Task 2: View the data mining model in the Sales Forecasting mining structure

- In the **Sales Forecasting.dmm Designer** in Business Intelligence Development Studio, view the **Mining Model Viewer** tab.
- Use the **Microsoft Time Series** viewer to view the **Time Series Forecast** model.
- View the information on the **Charts** tab.
    - Remove the **M200 Europe:Quantity**, **M200 North America:Quantity**, and **M200 Pacific:Quantity** series and review the remaining lines in the chart.

▶ Task 3: View the data mining models in the Targeted Mailing mining structure

- In the **Targeted Mailing.dmm Designer** in Business Intelligence Development Studio, view the **Mining Model Viewer** tab.
- View the data in the **Microsoft Tree Viewer**.
- View the dependency data.
  - Use the color legend to determine the direction of the dependencies, and which nodes are dependent on which other nodes.
  - Move the slider to show only the strongest links.

Results: After this exercise, you should have successfully deployed the data mining solution and viewed the data mining models.

Exercise 4: Validating Data Mining Models

Scenario

You must now validate the accuracy of the data mining models.

This exercise's main tasks are:

- View the accuracy of the mining models in the Targeted Mailing mining structure.

▶ Task 1: View the accuracy of the mining models in the Targeted Mailing mining structure

- Use the **Mining Accuracy Chart** tab of the **Targeted Mailing.dmm Designer**.
- Use the **vTargetMail (dbo)** view from the **Adventure Works DM DSV** data source view as the case table.
- Use a **Predict Value** of **1** in the **Select predictable mining model columns to show** in the lift chart.
- Use the **Lift Chart** tab to view the lift chart.
    - Notice that the **TM_Decision_Tree** model is the most accurate.
- Use the **Classification Matrix** tab to view a numeric representation of the accuracy of the different models.
    - The top line of the lift chart represents the actual data, the bottom line represents a random guess model. You can compare the results of the three mining models created in this mining structure to these two lines to determine the relative accuracy of each.

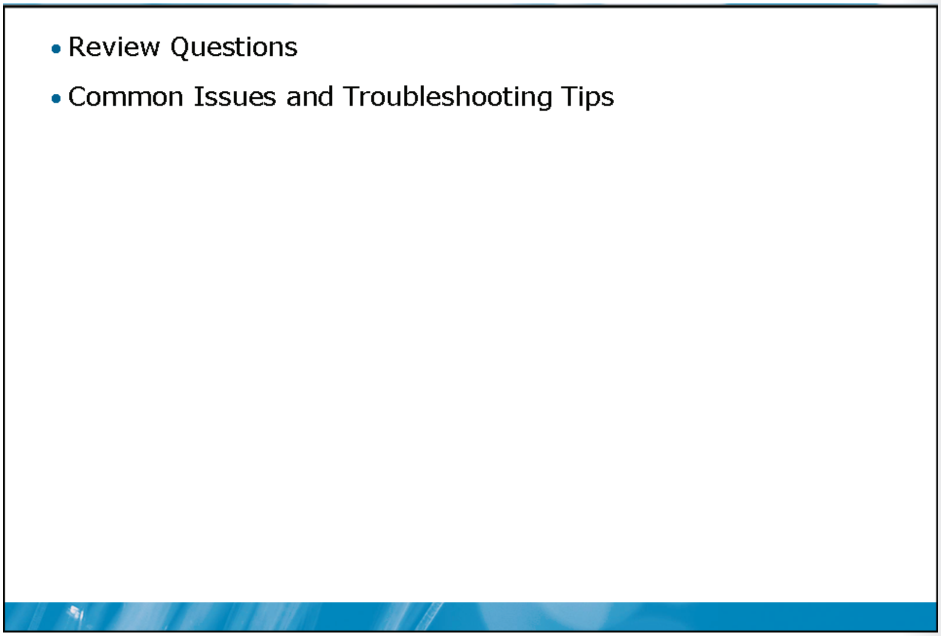
Results: After this exercise, you should have successfully validated the accuracy of the data mining models.

Note: The answers to the practices and labs are on the Student Materials CD.

Lab Shutdown

After you complete the lab, you must shut down the 6234A-NY-SQL-01 virtual machine and discard any changes.

Module Review and Takeaways



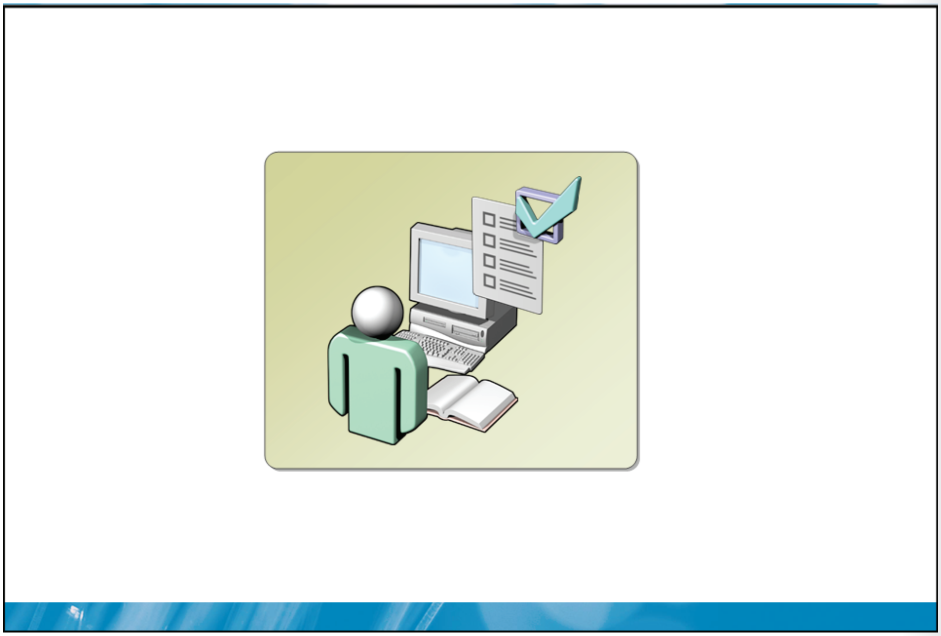
Review Questions

1. Why is data mining different from OLAP?
2. Describe various scenarios in which organizations may benefit from implementing data mining.
3. Which algorithms can be used to predict customer behavior?
4. What is the purpose of the accuracy chart?

Common Issues and Troubleshooting Tips

It is important to remember that data mining is an iterative process and that you should plan to refine and create many models throughout the life of your solution. There are a number of tools that can help you with this process.

Course Evaluation



Your evaluation of this course will help Microsoft understand the quality of your learning experience.

Please work with your training provider to access the course evaluation form.

Microsoft will keep your answers to this survey private and confidential and will use your responses to improve your future learning experience. Your open and honest feedback is valuable and appreciated.

Module 1: Introduction to Microsoft SQL Server Analysis Services

Lab: Installing and Verifying SQL Server Analysis Services

Exercise 1: Installing SQL Server Analysis Services

▶ Task 1: Start the 6234A-NY-SQL-02 virtual machine and log on as Administrator

1. In the Lab Launcher, next to **6234A-NY-SQL-02**, click **Launch**.
2. Log on to **NY-SQL-02** as **Student** using the password **Pa$$w0rd**.

▶ Task 2: Install an instance of SQL Server Analysis Services

1. Browse to **E:\Evaluation** and then double-click **Setup**. The **User Account Control** dialog box appears. Click **Continue**.
2. The **Microsoft SQL Server 2008 Setup** dialog box appears. Click **OK**.
3. The Windows Update Standalone Installer appears. Click **OK**.
4. The Download and Install Updates window appears. When installation is finished, click **Restart Now**.
5. Once the virtual machine has restarted, log on to **NY-SQL-02** as **Student** using the password **Pa$$w0rd**.
6. Browse to **E:\Evaluation** and then double-click **Setup**. The **User Account Control** dialog box appears. Click **Continue**.
7. The SQL Server Installation Center window opens. Click **Installation** on the left.
8. Click **New SQL Server stand-alone installation or add features to an existing installation**.
9. The SQL Server 2008 Setup window opens. Once the operation on the Setup Support Rules page has completed, click **OK**.
10. On the Product Key page, select **Specify a free edition: Enterprise Evaluation**, and then click **Next**.

11. On the License Terms page, check **I accept the license terms**, and then click **Next**.
12. On the Setup Support Files page, click **Install**. Once installation has finished, click **Next**.
13. On the Feature Selection page, select the **Analysis Services**, **Client Tools Connectivity**, **Management Tools – Basic**, **and Management Tools - Complete** checkboxes, and then click **Next**.
14. On the Instance Configuration page, click **Next**.
15. On the Disk Space Requirements page, click **Next**.
16. On the Server Configuration page, click **Use the same account for all SQL Server services**. The **Use the same account for all SQL Server 2008 services** dialog box appears.
17. Enter **sqlserver** as the account name and **Pa$$w0rd** as the password, and then click **OK**.
18. Click **Next**.
19. On the Analysis Services Configuration page, click **Add Current User**, and then click **Next**.
20. On the Error and Usage Reporting page, clear both checkboxes, and then click **Next**.
21. On the Installation Rules page, click **Next**.
22. On the Ready to Install page, click **Install**.
23. When installation is complete, click **Next**.

Results: After this exercise, SQL Server Analysis Services is installed.

Exercise 2: Verifying Installation

▶ Task 1: View the installation log file

1. Click the link to the log file.
2. In the Notepad window, press **CTRL+F**.
3. In the **Find what** field, type **Detailed results**, and then click **Find Next**. Take note of the log information regarding the installation of Analysis Services.
4. Close **Notepad**.
5. On the Complete page of **the SQL Server 2008 Setup** window, click **Close**.
6. Close SQL Server Installation Center.

▶ Task 2: Connect to Analysis Services

1. Click **Start**, click **All Programs | Microsoft SQL Server 2008**, and then click **SQL Server Management Studio**.
2. On the **Connect to Server** dialog box, in the **Server type** drop-down, click **Analysis Services**.
3. Verify that the **Server name** is **NY-SQL-02**, and then click **Connect**.
4. In the console tree, expand **Databases**. Take note that there are no databases installed by default.
5. Close SQL Server Management Studio.
6. Turn off **the 6234A-NY-SQL-02** virtual machine and discard any changes.

Results: After this exercise, you have verified that the Analysis Services installation was successful.

Module 2: Creating Multidimensional Analysis Solutions

Lab: Creating a Multidimensional Analysis Solution

Exercise 1: Creating a Data Source

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

1. In the Lab Launcher, next to **6234A-NY-SQL-01**, click **Launch**.
2. Log on to **NY-SQL-01** as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, verify that **Server type** is set to **Database Engine** and **Server name** is set to **NY-SQL-01**, and then click **Connect**.
3. Maximize Microsoft SQL Server Management Studio.
4. Expand **NY-SQL-01 | Databases | AdventureWorksDW2008 | Security**, right-click on the **Users** folder, and then click **New User**.
5. The Database User – New window opens.
6. In the **User** name field, enter **sqlserver**.
7. In the **Login** name field, enter **NY-SQL-01\sqlserver**.
8. In the **Default schema** field, enter **dbo**.
9. Under the **Schemas owned by this user** section, ensure that **db_owner** and **db_securityadmin** are checked.
10. Under the **Database role membership** section, ensure that **db_owner** and **db_securityadmin** are checked.
11. Click **OK**.
12. Close Microsoft SQL Server Management Studio.

▶ Task 3: Create a new Analysis Services project

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Business Intelligence Development Studio**.
2. On the **File** menu, point to **New**, and click **Project**.
3. In the **Templates** pane, click **Analysis Services Project**.
4. Click **Browse**, browse to the **E:\MOD02-03\Labfiles\Starter** folder, and then click **Select Folder**.
5. Change the value in the **Name** box to **Adventure Works OLAP**, and then click **OK**.

▶ Task 4: Create a data source for the AdventureWorksDW2008 relational database

1. In Solution Explorer, right-click the **Data Sources** folder, and then click **New Data Source**.
2. On the Welcome to the Data Source Wizard page, click **Next**.
3. On the Select how to define the connection page, click **New**.
4. In the **Connection Manager** dialog box, in the **Server name** box, type **NY-SQL-01**.
5. In the **Log on to the server** area, ensure that **Use Windows Authentication** is selected.
6. In the **Connect to a database** area, click **AdventureWorksDW2008** in the **Select or enter a database name** box, and then click **OK**.
7. On the Select how to define the connection page, click **Next**.
8. On the Impersonation Information page, select **Use the service account**, and then click **Next**.
9. On the Completing the Wizard page, change the data source name to **Adventure Works Data Warehouse**, and then click **Finish**.

Results: After this exercise, you should see the Adventure Works Data Warehouse.ds data source in the Data Sources folder.

Exercise 2: Creating and Modifying a Data Source View

▶ Task 1: Create a data source view

1. In the Solution Explorer, right-click the **Data Source Views** folder, and then click **New Data Source View**.
2. On the Welcome to the Data Source View Wizard page, click **Next**.
3. On the Select a Data Source page, verify that the **AdventureWorksDW2008** data source is selected, and then click **Next**.
4. In the **Available objects** list, click **DimCustomer (dbo)**, and then hold down the CTRL key and click **DimDate (dbo)**, **DimGeography (dbo)**, **DimProduct (dbo)**, and **FactInternetSales (dbo)**.
5. On the Select Tables and Views page, click the **>** button to add the selected tables to the **Included objects** list, and then click **Next**.
6. On the Completing the Wizard page, change the name of the data source view to **Adventure Works DSV**, and then click **Finish**. Note that the Data Source View Designer opens automatically after you click Finish.

▶ Task 2: Modify the data source view

1. In the **Data Source View Designer**, use the **Zoom** icon to change the zoom level to **50%**.
2. In the **Diagram** pane, click the **FactInternetSales** table.
3. In the **Properties** pane, in the **FriendlyName** box, type **InternetSales**.
4. In the **Diagram** pane, click the **DimCustomer** table.
5. In the **Properties** pane, in the **FriendlyName** box, type **Customer**.
6. Change the **FriendlyName** of the remaining tables to remove the *dim* prefix.
7. In the **Diagram** pane, right-click the **Customer** table, and then click **New Named Calculation**.
8. In the **Create Named Calculation** dialog box, type **Full Name** in the **Column name** box.

9. In the **Expression** box, type the following:

```
CASE
   WHEN MiddleName IS NULL THEN
   FirstName + ' ' + LastName
   ELSE
   FirstName + ' ' + MiddleName + ' ' + LastName
END
```

10. In the **Create Named Calculation** dialog box, click **OK**.
11. In the **Diagram** pane, right-click the **Date** table, and then click **New Named Calculation**.
12. In the **Create Named Calculation** dialog box, type **Simple Date** in the **Column name** box.
13. In the **Expression** box, type the following:

```
DATENAME(mm, FullDateAlternateKey) + ' ' +
DATENAME(dd, FullDateAlternateKey) + ' ' +
DATENAME(yy, FullDateAlternateKey)
```

14. In the **Create Named Calculation** dialog box, click **OK**.
15. On the **File** menu, click **Save All**.
16. In the **Diagram** pane, right-click the **Date** table, and then click **New Named Calculation**.
17. In the **Create Named Calculation** dialog box, type **Calendar Semester Description** in the **Column name** box.
18. In the **Expression** box, type the following:

```
CASE
   WHEN CalendarSemester = 1 THEN
   'H1' + 'CY' + CONVERT(CHAR(4), CalendarYear)
   ELSE
   'H2' + 'CY' + CONVERT(CHAR(4), CalendarYear)
END
```

19. In the **Create Named Calculation** dialog box, click **OK**.
20. In the **Diagram** pane, right-click the **Date** table, and then click **New Named Calculation**.

21. In the **Create Named Calculation** dialog box, type **Calendar Quarter Description** in the **Column name** box.
22. In the **Expression** box, type the following:

```
'Q' + CONVERT(CHAR(1), CalendarQuarter) + ' CY ' + CONVERT(CHAR(4),
CalendarYear)
```

23. In the **Create Named Calculation** dialog box, click **OK**.
24. On the **File** menu, click **Save All**.

Results: After this exercise, you should see the Adventure Works DSV.dsv data source view in the Data Source Views folder.

Exercise 3: Creating and Modifying a Cube

▶ Task 1: Create a cube

1. In the Solution Explorer, right-click the **Cubes** folder, and then click **New Cube**.
2. On the Welcome to the Cube Wizard page, click **Next**.
3. On the Select Creation Method page, verify that **Use existing tables** is selected, and then click **Next**.
4. On the Select Measure Group Tables page, click **Suggest**. Take note that the wizard selects the **InternetSales** table as the measure group table, and then click **Next**.
5. On the Select Measures page, click **Next**.
6. On the Select New Dimensions page, clear the **Internet Sales** check box and then click **Next**. Take note that the **InternetSales** box clears automatically.
7. On the Completing the Wizard page, change the **Cube name** to **Adventure Works Cube**, and then click **Finish**. Notice that the Cube Designer opens automatically after you click Finish.

▶ Task 2: Modify the cube

1. In the **Measures** pane of the Cube Designer, expand the **Internet Sales** measure group, right-click the **Unit Price Discount Pct** measure, and then click **Rename**.
2. Rename **Unit Price Discount Pct** to **Unit Price Discount Percent**.
3. In the **Measures** pane, right-click **Tax Amt**, and then click **Rename**.
4. Rename **Tax Amt** to **Tax Amount**.
5. In the **Measures** pane, right-click anywhere, point to **Show Measures in**, and then click **Grid**.
6. In the **Measures** pane, click **Unit Price**, and then hold the CTRL key and click the **Extended Amount**, **Discount Amount**, **Product Standard Cost**, **Total Product Cost**, **Sales Amount**, and **Tax Amount** measures.
7. In the **Properties** pane, click **Currency** in the **FormatString** box.
8. In the **Measures** grid, click **Unit Price Discount Percent**.
9. In the **Properties** pane, click **Percent** in the **FormatString** box.
10. On the **File** menu, click **Save All**.

▶ Task 3: Modify dimensions

1. In Solution Explorer, right-click **Customer.dim**, and then click **View Designer**.
2. In the **Data Source View** pane, in the **Geography** table, click **City**.
3. Hold SHIFT and click **SalesTerritoryKey**, and then drag the selected columns to the **Attributes** pane.
4. In the **Data Source View** pane, in the **Customer** table, click **CustomerAlternateKey**.
5. Scroll to the bottom of the **Customer** table, hold SHIFT and click **Full Name**.
6. Drag the selected columns to the **Attributes** pane.
7. On the **File** menu, click **Save All**.
8. Close the **Customer.dim** dimension designer.
9. In Solution Explorer, right-click **Product.dim**, and then click **View Designer**.

10. In the **Data Source View** pane, in the **Product** table, click **ProductAlternateKey**.
11. Scroll to the bottom of the **Product** table, hold SHIFT and click **Status**.
12. Drag the selected columns to the **Attributes** pane.
13. In the **Attributes** pane, click **Large Photo** to select only that item, right-click **Large Photo**, click **Delete**, and then click **OK**.
14. On the **File** menu, click **Save All**.
15. Close the **Product.dim** dimension designer.
16. In Solution Explorer, right-click **Date.dim**, and then click **View Designer**.
17. In the **Data Source View** pane, in the **Date** table, click **FullDateAlternateKey**.
18. Scroll to the bottom of the **Date** table, hold SHIFT and click **Simple Date**.
19. Drag the selected columns to the **Attributes** pane.

▶ Task 4: Modify attribute columns

1. In the **Attributes** pane, click **Calendar Quarter**.
2. In the **Properties** pane, scroll down to the **Source** section.
3. Click the **KeyColumns** field, and then click the **ellipsis (...)** button.
4. In the **Key Columns** dialog box, in the **Available Columns** table, click **Calendar Year**, and then click the **right arrow (>)** icon.
5. Click the up arrow to move **CalendarYear** above **CalendarQuarter**, and then click **OK**.
6. In the **Properties** pane, click the **NameColumn** field, and then click the **ellipsis (...)** button.
7. In the **Name Column** dialog box, in the **Source column** field, click **Calendar Quarter Description**, and then click **OK**.
8. In the **Attributes** pane, click **English Month Name**.
9. In the **Properties** pane, change the **Name** field to **Month**.
10. Scroll down to the **Source** section, click the **KeyColumns** field, and then click the **ellipsis (...)** button.
11. In the **Key Columns** dialog box, in the **Key Columns** table, click **EnglishMonthName**, and then click the **left arrow (<)** icon.

12. In the **Available Columns** table, click **Calendar Year**, and then click the **right arrow** (>) icon.
13. In the **Available Columns** table, click **MonthNumberOfYear**, click the **right arrow** (>) icon, and then click **OK**.
14. In the **Properties** pane, click the **NameColumn** field, and then click the **ellipsis (…)** button.
15. In the **Name Column** dialog box, in the **Source column** field, click **EnglishMonthName**, and then click **OK**.
16. In the **Attributes** pane, click **Calendar Semester**.
17. In the **Properties** pane, scroll down to the **Source** section.
18. Click the **KeyColumns** field, and then click the **ellipsis (…)** button.
19. In the **Key Columns** dialog box, in the **Available Columns** table, click **Calendar Year**, and then click the **right arrow** (>) icon.
20. Click the up arrow to move **CalendarYear** above **CalendarSemester**, and then click **OK**.
21. In the **Properties** pane, click the **NameColumn** field, and then click the **ellipsis (…)** button.
22. In the **Name Column** page, in the **Source column** field, click **Calendar Semester Description**, and then click **OK**.
23. On the **File** menu, click **Save All**.
24. Close the **Date.dim** dimension designer.

▶ Task 5: Browse the cube

1. In Solution Explorer, right-click the **Adventure Works OLAP** project, and then click **Deploy**.
2. When deployment has completed successfully, click the **Browser** tab in the **Cube Designer**.

> **Tip**: Click the **Auto Hide** icon on the various panes in Business Intelligence Development Studio to make it easier to see the entire Cube Browser window.

3. In the **Measure Group** pane, expand **Measures**, expand **Internet Sales**, and then drag the **Sales Amount** measure to the **Drop Totals or Detail Fields Here** area of the **Data** pane.
4. In the **Measure Group** pane, drag **Internet Sales Count** and drop it next to the **Sales Amount** column.
5. In the **Measure Group** pane, drag **Unit Price Discount Percent** and drop it next to the **Internet Sales Count** column.
6. Verify that the format strings for the measures are correct.
7. In the **Measure Group** pane, expand the **Order Date** dimension. Drag the **Order Date.Date Key** hierarchy to the **Drop Row Fields Here** area of the **Data** pane.
8. In the **Measure Group** pane, expand the **Customer** dimension, expand the **Full Name** attribute, expand the **Members** node, expand the **All** node, and verify that the named calculation created in Exercise 2 is functioning.
9. Minimize SQL Server Business Intelligence Development Studio.
10. Click **Start**, point to **All Programs**, click **Microsoft Office**, and then click **Microsoft Office Excel 2007**.
11. Maximize the Microsoft® Office Excel® 2007 window.
12. On the toolbar click **Data**, then click **From Other Sources**, and then click **From Analysis Services**.
13. In the **Server name** field of the **Data Connection Wizard** enter **NY-SQL-01**, verify that **Use Windows Authentication** is selected, and then click **Next**.

14. On the Select Database and Table page of the **Data Connection Wizard**. click **Next**.
15. On the Save Data Connection File and Finish page, click **Finish**.
16. On the **Import Data** dialog box, select **PivotTable Report**, and then click **OK**.
17. In the **PivotTable Field List**, expand the **Internet Sales** measure, and then click **Internet Sales Count**.
18. Scroll down the **PivotTable Field List**, expand the **Customer** dimension, and then click **Full Name**.
19. Choose other fields from the **PivotTable Field List** to continue browsing the cube.
20. When you are finished browsing the **Adventure Works Cube** cube in Microsoft Office Excel 2007, close the window without saving the changes.
21. Maximize SQL Server Business Intelligence Development Studio.
22. On the **File** menu, click **Save All** to save your project.

Note: This solution is used in the lab for module 3, so do not delete changes.

Results: After this exercise, you should have successfully deployed the cube and reviewed the modifications made to ensure they are reflected when browsing the cube.

Module 3: Working with Cubes and Dimensions

Lab: Defining Dimensions

Exercise 1: Configuring Dimensions

▶ Task 1: Open the Adventure Works OLAP solution

1. In the Lab Launcher, next to **6434A-NY-SQL-01**, click **Launch**.
2. Log on to **NY-SQL-01** as **Administrator** using the password **Pa$$w0rd**.
3. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Business Intelligence Development Studio**.

▶ Task 2: Remove attributes from the Customer dimension

1. In Solution Explorer, right-click the **Customer.dim** dimension, and then click **View Designer**.
2. In the **Attributes** pane, press the CTRL key and click the **French Country Region Name**, **French Education**, **French Occupation**, **Spanish Country Region Name**, **Spanish Education**, and **Spanish Occupation** attributes, right-click any of the highlighted attributes, and then click **Delete**.
3. In the **Delete Objects** dialog box, click **OK**.
4. On the **File** menu, click **Save All**.
5. In the **Customer Dimension Designer**, click the **Close** icon.

▶ Task 3: Remove an attribute from the Product dimension

1. In Solution Explorer, right-click the **Product.dim** dimension, and then click **View Designer**.
2. In the **Attributes** pane, right-click the **Thai Description** attribute, and then click **Delete**.
3. In the **Delete Objects** dialog box, click **OK**.
4. On the **File** menu, click **Save All**.
5. In the **Product Dimension Designer**, click the **Close** icon.

▶ Task 4: Add dimension intelligence

1. In Solution Explorer, right-click **Date.dim**, and then click **Add Business Intelligence**.
2. On the **Welcome to the Business Intelligence Wizard** page, click **Next**.
3. On the Choose Enhancement page, click **Define dimension intelligence**, and then click **Next**.
4. On the Define Dimension Intelligence page, in the **Dimension type** field, click **Time**.
5. In the **Dimension attributes** table, select the **Include** check box for the following **Attribute Types**, and select the corresponding item in the **Dimension Attribute** column:

Attribute Type	Dimension Attribute
Year	Calendar Year
Half Year	Calendar Semester
Quarter	Calendar Quarter
Month	Month
Date	Simple Date

6. Click **Next**, and then click **Finish**.

Results: After this exercise, you should see that the Customer, Product, and Date dimensions have been modified, and time intelligence has been added to the date dimension.

Exercise 2: Defining Relationships and Hierarchies

▶ Task 1: Create attribute relationships in the Date dimension

1. In Solution Explorer, right-click the **Date.dim** dimension, and then click **View Designer**.
2. In the **Attributes** pane, click the **Simple Date** attribute.
3. In the **Properties** pane, change **Simple Date** to **Date** in the **Name** box.
4. Click the **Attributes Relationships** tab.
5. In the **Attribute Relationships** pane, right-click an empty space, and then click **New Attribute Relationship**.
6. In the **Create Attribute Relationship** dialog box, in the **Source Attribute** section, in the **Name** field, click **Date**.
7. In the **Related Attribute** section, in the **Name** field, click **Month**, and then click **OK**.
8. Repeat step 5 to create the following relationships:

Source Attribute	Related Attribute
Month	Calendar Quarter
Calendar Quarter	Calendar Semester
Calendar Semester	Calendar Year

▶ Task 2: Create a natural hierarchy in the Date dimension

1. Click the **Dimension Structure** tab.
2. In the **Attributes** pane, drag **Calendar Year** into the **Hierarchies** pane.
3. In the **Attributes** pane, drag the following attributes one-by-one to the **Hierarchies** pane and drop them on to the **<new level>** area: **Calendar Semester**, **Calendar Quarter**, **Month**, and **Date**.
4. In the **Hierarchies** pane, click the title bar of the **Hierarchy** hierarchy.
5. In the **Properties** pane, change the **Name** field to **Calendar Date**.
6. On the **File** menu, click **Save All**.
7. Close the **Date Dimension Designer**.

▶ Task 3: Create hierarchies in the Customer dimension

1. In Solution Explorer, right-click the **Customer.dim** dimension, and then click **View Designer**.
2. In the **Attributes** pane, right-click **English Country Region Name**, and then click **Rename**. Rename the attribute to **Country-Region**.
3. Rename the **State Province Name** attribute to **State-Province**.
4. In the **Attributes** pane, drag **Country-Region** into the **Hierarchies** pane.
5. In the **Attributes** pane, drag the following attributes one-by-one to the **Hierarchies** pane and drop them on to the **<new level>** area: **State-Province**, **City**, **Postal Code**, and **Full Name**.
6. In the **Hierarchies** pane, click the title bar of the **Hierarchy** hierarchy.
7. In the **Properties** pane, change the **Name** field to **Customer Geography**.

> **Note**: The hierarchy may display a warning icon. This can be ignored.

8. In the **Attributes** pane, drag **Gender** into an empty area of the **Hierarchies** pane.
9. In the **Attributes** pane, drag **Marital Status** to the **<new level>** area beneath **Gender** in the **Hierarchies** pane.
10. In the **Hierarchies** pane, click the title bar of the **Hierarchy** hierarchy.
11. In the **Properties** pane, change the **Name** field to **Gender - Marital Status**.

> **Note**: The hierarchy may display a warning icon. This can be ignored.

12. On the **File** menu, click **Save All**.
13. Close the **Customer Dimension Designer**.

> **Results**: After this exercise, you should have created a Calendar Date hierarchy, a Customer Geography hierarchy, and a Gender - Marital Status hierarchy.

Exercise 3: Sorting and Grouping Dimension Attributes

▶ Task 1: Modify the sort order of the Month attribute

1. In Solution Explorer, right-click the **Date.dim** dimension, and then click **View Designer**.
2. In Solution Explorer, right-click the **Adventure Works OLAP** solution, and then click **Deploy**. Wait for the **Deploy Succeeded** message in the status bar.
3. Click the **Browser** tab of the **Date.dim Dimension Designer**, and then click the **Reconnect** icon.
4. In the designer pane, right-click **All** and click **Expand All**. Notice that the months are listed under the correct quarter, semester, and year, but that they are sorted in alphabetical order instead of chronological order.
5. Click the **Dimension Structure** tab.
6. In the **Attributes** column, click **Month Number of Year**.
7. If necessary, click the **Properties** tab.
8. In the **Properties** pane, set the following values:

Property	Value
AttributeHierarchyOptimizedState	NotOptimized
AttributeHierarchyOrdered	False

9. Click the **Attribute Relationships** tab. In the **Attributes** pane, right-click **Month** , and then click **New Attribute Relationship**.
10. In the **Create Attribute Relationship** window, in the **Related Attribute** section, in the **Name** field, click **Month Number of Year**, and then click **OK**.
11. Click the **Dimension Structure** tab.
12. In the **Attributes** pane, click **Month**.
13. In the **Properties** pane, click **AttributeKey** in the **OrderBy** box, and then in the **OrderByAttribute** box, click **Month Number Of Year**.
14. On the **File** menu, click **Save All**.

15. In Solution Explorer, right-click the **Adventure Works OLAP** solution, and then click **Deploy**. Wait for the **Deploy Succeeded** message in the status bar.
16. On the **Browser** tab of the **Date.dim Dimension Designer**, click the **Reconnect** icon.
17. If necessary, in the **Hierarchy** box, click **Calendar Date**. Expand the hierarchy to the month level and notice that the months are now in date order rather than alphabetical order.

▶ Task 2: Group the Yearly Income attribute hierarchy members in the Customer dimension

1. In Solution Explorer, right-click the **Customer.dim** dimension, and then click **View Designer**.
2. In the **Data Source View** pane, right-click the **Customer** table, and then click **Explore Data**. Notice the range of values for the **YearlyIncome** column.
3. On the **Explore Customer Table** tab, click the **Close** icon.
4. In the **Attributes** pane, click **Yearly Income**.
5. If necessary, click the **Properties** tab.
6. In the **Properties** pane, click **Automatic** in the **DiscretizationMethod** box, and then type **5** in the **DiscretizationBucketCount** box.
7. In Solution Explorer, right-click the **Adventure Works OLAP** solution, and then click **Deploy**. Wait for the **Deploy Succeeded** message in the status bar.
8. On the **Browser** tab of the **Customer.dim Dimension Designer**, click the **Reconnect** icon.
9. In the **Hierarchy** box, click **Yearly Income**.
10. Expand the **All** level and view the salary ranges created.
11. Click **File**, and then click **Save All**.
12. Turn off the 6234A-NY-SQL-01 virtual machine and discard any changes.

Results: After this exercise, you should be able to view the correctly sorted and grouped dimensions in the cube.

Module 4: Working with Measures and Measure Groups

Lab: Configuring Measures and Measure Groups

Exercise 1: Configuring Measures

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

1. In the Lab Launcher, next to **6234A-NY-SQL-01**, click **Launch**.
2. Log on to **NY-SQL-01** as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, verify that **Server type** is set to **Database Engine** and **Server name** is set to **NY-SQL-01**, and then click **Connect**.
3. Maximize Microsoft SQL Server Management Studio.
4. Expand **NY-SQL-01 | Databases | AdventureWorksDW2008 | Security**, right-click on the **Users** folder, and then click **New User**.
5. The Database User – New window opens.
6. In the **User name** field, enter **sqlserver**.
7. In the **Login name** field, enter **NY-SQL-01\sqlserver**.
8. In the **Default schema** field, enter **dbo**
9. Under the **Schemas owned by this user** section, ensure that **db_owner** and **db_securityadmin** are checked.
10. Under the **Database role membership section**, ensure that **db_owner** and **db_securityadmin** are checked.
11. Click **OK**.
12. Close Microsoft SQL Server Management Studio.

▶ Task 3: Configure measure display format using the FormatString property

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Business Intelligence Development Studio**.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** window, browse to the **E:\Mod04\Labfiles** folder, click the **Mod04Lab.dwproj** project file, and then click **Open**.
4. In Solution Explorer, right-click **Adventure Works.cube**, and then click **Open**.
5. Switch to the **Cube Structure** tab of Cube Designer for the Adventure Works cube, expand the **Internet Sales** measure group in the **Measures** pane, right-click **Internet Order Quantity**, and then click **Properties**.
6. In the Properties window, in the **FormatString** list, enter **#,#**.
7. On the toolbar of the **Cube Structure** tab, click **Show Measures Grid**.
8. Select the following measures, by holding down the CTRL key to select multiple measures: **Unit Price**, **Extended Amount**, **Discount Amount**, **Product Standard Cost**, **Total Product Cost**, **Sales Amount**, **Tax Amt**, and **Freight**.
9. In the Properties window, in the **FormatString** list, select **Currency**.
10. In the drop-down list box at the top of the Properties window, select the measure **Unit Price Discount Pct**, and then select **Percent** in the **FormatString** list.

▶ Task 4: Configure measure display name using the Name property

1. In the Properties window, change the **Name** property for the **Unit Price Discount Pct** measure to **Unit Price Discount Percent**.
2. In the **Measures** pane, click **Tax Amt** and change the name of this measure to **Tax Amount**.
3. On the **File** menu, click **Save All**.

▶ Task 5: Deploy the Analysis Services project and review formatting changes

1. Click **Show Measures Tree** on the toolbar of the **Cube Structure** tab.
2. On the **Build** menu of **SQL Server Business Intelligence Development Studio**, click **Deploy Module04Lab**.

> If you get a deployment error, right-click **Adventure Works.ds**, click **Open**, click the **Impersonation Information** tab, select **Use a specific Windows user name and password**, enter **Administrator** as the User name and **Pa$$w0rd** as the Password, click **OK**, and then repeat step 2.

3. When deployment has successfully completed, click the **Browser** tab of Cube Designer.
4. In the **Measure Group** pane, expand **Measures**, expand **Reseller Sales**, and add **Unit Price**, **Unit Price Discount Percent**, **Extended Amount**, **Discount Amount**, **Product Standard Cost**, **Total Product Cost**, **Sales Amount**, **Tax Amount**, and **Freight** to the data area.
5. Keep SQL Server Business Intelligence Development Studio open for the next exercise.

> **Results**: After this exercise, you should have configured format strings and display names for several measures. You should have also deployed and reviewed the Analysis Services project.

Exercise 2: Defining Dimension Usage and Relationships

▶ Task 1: Define a Referenced Relationship

1. In Solution Explorer, right-click **Reseller.dim**, and then click **View Designer**.
2. If **GeographyKey** is not visible in the **Reseller** attribute, click **Show Table** on Dimension Structure toolbar, select **Geography** from the Show Tables window, click **OK,** and then drag **GeographyKey** from **Geography** in the **Data Source View** pane to the **Reseller** dimension in the **Attributes** pane.
3. In the **Attributes** pane, right-click **GeographyKey** and click **Properties**.
4. Change the **AttributeHierarchyOptimizedState** property to **NotOptimized**.
5. Change the **AttributeHierarchyVisible** property to **False**.
6. Switch to Cube Designer for the Adventure Works cube, click the **Dimension Usage** tab, and then click the ellipsis button (**...**) at the intersection of the **Reseller Sales** measure group and the **Geography** cube dimension.
7. In the **Select relationship** type list, select **Referenced**.
8. In the **Intermediate dimension** list, select **Reseller**.
9. In the **Reference dimension attribute** list, select **Geography Key**, and then select **Geography Key** in the **Intermediate dimension attribute** list.
10. Click **OK**.
11. In the **Dimensions** list on the **Dimension Usage** tab, right-click **Geography**, and then click **Rename**.
12. Change the name of this cube dimension to **Reseller Geography**, and then press ENTER to make this name change take effect.
13. On the **Build** menu, click **Deploy Module04Lab**.
14. When deployment has successfully completed, click the **Browser** tab in Cube Designer for the **Adventure Works** cube, and then click **Reconnect**.
15. In the **Measure Group** pane, expand **Reseller Geography**, right-click **Reseller Geography.Geography Key**, and then click **Add to Row Area**.

▶ Task 2: Define a Regular Relationship

1. On the **Project** menu, click **New Dimension**. The **Dimension Wizard** opens.
2. On the Welcome to the Dimension Wizard page, click **Next**.
3. On the Select Creation Method page, select **Use an existing table** and click **Next**.
4. On the Specify Source Information page, select **InternetSales** in the **Main table** list, select **CustomerPONumber** in the **Name column** list, and then click **Next**.
5. On the Select Related Tables page, deselect all related tables, and then click **Next**.
6. On the Select Dimension Attributes page, deselect all attributes except for **Sales Order Number**, and then click **Next**.
7. On the Completing the Wizard page, change the dimension name to **Internet Sales Order Details**, and then click **Finish**.
8. In the **Attributes** pane, click **Sales Order Number**.
9. In the **Properties** pane, change the **Name** property of the **Sales Order Number** attribute to **Order Number**.
10. Change the **OrderBy** property to **Key**.
11. Drag the **Order Number** attribute to the **Hierarchies** pane to create a new hierarchy.
12. Switch to **Cube Designer** for the **Adventure Works** cube, and click on the **Cube Structure** tab.
13. In the **Dimensions** pane, right-click on the **Adventure Works** cube, and then click **Add Cube Dimension**.
14. On the Add Cube Dimension window, click **Internet Sales Order Details**, and then click **OK**.
15. Click the **Dimension Usage** tab.
16. At the intersection of the **Internet Sales** measure group and the **Internet Sales Order Details** dimension, click the ellipsis button (**...**) in the cell to review the relationship properties.
17. The **Define Relationship** dialog box opens. Select **Regular** from the **Select relationship type** drop-down.

18. Select **Order Number** from the **Granularity attribute** drop-down menu.
19. Select **SalesOrderNumber** from the drop-down menu in the first row of the **Measure Group Columns** column.
20. Select **SalesOrderLineNumber** from the drop-down menu in the second row of the **Measure Group Columns** column.
21. Click **OK**.
22. Keep SQL Server Business Intelligence Development Studio open for the next exercise.

Results: After this exercise, you should have defined dimension usage and relationships for measure groups by defining a referenced relationship and defining a regular relationship.

Exercise 3: Configuring Measure Group Storage

▶ Task 1: Configure the storage mode of the Internet Sales measure group

1. In **Cube Designer** for the **Adventure Works** cube, on the **Cube Structure** tab, in the **Measures** pane, right-click **Internet Sales**, and then click **Properties**.
2. In the **Properties** window, click the browse (**...**) button for the **ProactiveCaching** property setting.
3. Under **Standard setting**, review each of the storage setting options. When finished, drag the slider to **Automatic MOLAP** and then click **OK**.

▶ Task 2: Design aggregations for the Internet Sales measure group

1. Click the **Aggregations** tab of **Cube Designer** for the **Adventure Works** cube.
2. Right-click **Internet Sales (0 Aggregation Designs)**, and then click **Design Aggregations**.
3. The **Aggregation Design Wizard** appears. Click **Next**.
4. On the Review Aggregation Usage page, click **Set All to Default**, and then click **Next**.
5. On the **Specify Object Counts** page, click **Count**.

> **Note**: This process may take several minutes.

6. Once the count process has completed, click **Next**.
7. On the Set Aggregations Options page, select **I click Stop**, and then click **Start**.
8. Click **Stop** when the graph on the right reaches roughly 60%, and then click **Next**.
9. On the Completing the Wizard page, change the name of the aggregation to **InternetSalesAgg**, select **Save the aggregations but do not process them**, and then click **Finish**.
10. In the **Aggregations** tab of **Cube Designer** for the **Adventure Works** cube, expand **Internet Sales** to view the new aggregation named **InternetSalesAgg**.
11. Turn off the 6234A-NY-SQL-01 virtual machine and discard any changes.

> **Results**: After this exercise, you should have defined aggregations and storage modes for the Internet Sales Measure group.

Module 5: Querying Multidimensional Analysis Solutions

Lab: Querying a Cube

Exercise 1: Querying a Cube by Using MDX

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

1. In the Lab Launcher, next to **6234A-NY-SQL-01**, click **Launch**.
2. Log on to **NY-SQL-01** as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, verify that **Server type** is set to **Database Engine** and **Server name** is set to **NY-SQL-01**, and then click **Connect**.
3. Maximize Microsoft SQL Server Management Studio.
4. Expand **NY-SQL-01 | Databases | AdventureWorksDW2008 | Security**, right-click on the **Users** folder, and then click **New User**.
5. The Database User – New window opens.
6. In the **User name** field, enter **sqlserver**.
7. In the **Login name** field, type **NY-SQL-01\sqlserver**.
8. In the **Default schema** field, enter **dbo**
9. Under the **Schemas owned by this user** section, ensure that **db_owner** and **db_securityadmin** are checked.
10. Under the **Database role membership section**, ensure that **db_owner** and **db_securityadmin** are checked.
11. Click **OK**.
12. Close Microsoft SQL Server Management Studio.

▶ Task 3: Write MDX queries to show Internet Sales amounts and Reseller Sales amounts

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Business Intelligence Development Studio**.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** window, browse to the **E:\Mod05\Labfiles** folder, click the **Mod05Lab.dwproj** project file, and then click **Open**.
4. In the **Solution Explorer** pane, right-click **Module05Lab**, and then click **Deploy**.
5. Close SQL Server Business Intelligence Development Studio.
6. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Management Studio**.
7. In the **Connect to Server** dialog box, in the **Server type** list, click **Analysis Services**.
8. In the **Server name** list, type **NY-SQL-01** and then click **Connect**.
9. Click the **New Query** window.
10. Type the following MDX query:

```
SELECT
  [Measures].[Sales Amount - Internet Sales] On 0,
  NonEmpty([Dim Product].[Product Key].Members,
  [Measures].[Sales Amount - Internet Sales]) On 1
FROM [Adventure Works]
```

11. Click the **Execute** button.
12. Review the query results.
13. Close the query window, in the **Microsoft SQL Server Management Studio** dialog box, click **No**.
14. Click the **New Query** window.

15. Type the following MDX query:

```
SELECT
  [Measures].[Sales Amount] On 0,
  NonEmpty([Dim Product Category].[Product Category Key].Members,
[Measures].[Sales Amount]) On 1
FROM [Adventure Works]
```

16. Click the **Execute** button.
17. Review the query results.
18. Close the query window, in the **Microsoft SQL Server Management Studio** dialog box, click **No**.

▶ Task 4: Write an MDX query to show Internet Sales amounts across Customer geography

1. Click the New Query window.
2. Type the following MDX query:

```
SELECT
  [Dim Date].Members ON COLUMNS,
  [Dim Customer - Geography].Members ON ROWS
FROM [Adventure Works]
WHERE [Measures].[Sales Amount]
```

3. Click the **Execute** button.
4. Review the query results.
5. Close the query window, in the **Microsoft SQL Server Management Studio** dialog box, click **No**.
6. Close SQL Server Management Studio.

Results: After this exercise, you should have written an MDX query to show Internet Sales amounts, written an MDX query to show Reseller Sales amounts, and written an MDX query to show Internet Sales amounts across Customer geography.

Exercise 2: Creating a Calculated Member

▶ Task 1: Define calculations to aggregate physical measures

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Business Intelligence Development Studio**.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** window, browse to the **E:\Mod05\Labfiles** folder, click the **Module05Lab** project file, and then click **Open**.
4. In Solution Explorer, right-click **Adventure Works.cube**, click **Open**, and then click the **Calculations** tab.
5. On the toolbar of the **Calculations** tab, click **New Calculated Member**.
6. In the **Name** box, change the name of the calculated measure to **[Total Sales Amount]**.
7. On the **Metadata** tab in the **Calculation Tools** pane of the **Calculations** tab, expand **Measures** and then expand **Internet Sales** to view the metadata for the **Internet Sales** measure group.

> **Note**: If you cannot view any metadata in the **Calculation Tools** pane, click **Reconnect** on the toolbar. If this does not work, you may have to process the cube or start the instance of Analysis Services.

8. Drag **Sales Amount** from the **Metadata** tab in the **Calculation Tools** pane into the **Expression** box in the **Calculation Expressions** pane.
9. In the **Expression** box, type a plus sign (**+**) after **[Measures].[Sales Amount]**.
10. On the **Metadata** tab in the **Calculation Tools** pane, expand **Reseller Sales**, and then drag **Sales Amount – Fact Reseller Sales** into the **Expression** box in the **Calculation Expressions** pane after the plus sign (+).
11. In the **Format string** list, select **"Currency"**.
12. In the **Non-empty behavior** list, select the check boxes for **Sales Amount** and **Sales Amount – Fact Reseller Sales**, and then click **OK**.

▶ Task 2: Define gross profit margin calculations

1. Verify that **[Total Sales Amount]** is selected in the **Script Organizer** pane, and then click **New Calculated Member** on the toolbar of the **Calculations** tab.
2. In the **Name** box, change the name of this new calculated measure to **[Internet GPM]**.
3. In the **Expression** box, create the following MDX expression:

```
([Measures].[Sales Amount] -
[Measures].[Total Product Cost]) /
[Measures].[Sales Amount]
```

4. In the **Format string** list, select "**Percent**".
5. In the **Non-empty behavior** list, select the check box for **Internet Sales Amount**, and then click **OK**.
6. On the toolbar of the **Calculations** tab, click **New Calculated Member**.
7. In the **Name** box, change the name of this new calculated measure to **[Reseller GPM]**.
8. In the **Expression** box, create the following MDX expression:

```
([Measures].[Sales Amount - Fact Reseller Sales] -
[Measures].[Total Product Cost - Fact Reseller Sales]) /
[Measures].[Sales Amount - Fact Reseller Sales]
```

9. In the **Format** string list, select "**Percent**".
10. In the **Non-empty behavior** list, select the check box for **Reseller Sales Amount**, and then click **OK**.
11. On the **File** menu, click **Save All**.

▶ Task 3: Browse the new calculated members

1. On the **Build** menu of Business Intelligence Development Studio, click **Deploy Mod05Lab**.
2. When deployment has successfully completed, click the **Browser** tab. If there is data in the Data area, right-click anywhere in the Data area and then click **Clear Results**.

> **Note**: If you see a warning that the structure of the cube has changed, click the link to process the cube. When the Process Cube window appears, click **Run**. When cube processing is complete, click **Close** twice.

3. In the **Measure Group** pane, expand **Measures** to view the new calculated members in the **Measures** dimension.
4. Drag the **Total Sales Amount**, **Internet GPM**, and **Reseller GPM** measures to the **Drop Totals of Detail Fields Here** area, and then review the results.
5. Keep SQL Server Business Intelligence Development Studio open for the next exercise.

> **Results**: After this exercise, you should have created several calculated members and then browsed the calculated members to review your work.

Exercise 3: Defining a Named Set

▶ Task 1: Define a Core Products named set

1. Click the **Calculations** tab of **Cube Designer** for the **Adventure Works** cube.
2. Click **New Named Set** on the toolbar of the **Calculations** tab.
3. In the **Name** box, change the name of the new named set to **[Core Products]**.
4. On the **Metadata** tab in the **Calculation Tools** pane, expand **Dim Product Category**, expand **English Product Category Name**, expand **Members**, and then expand **All**.

> **Note**: If you cannot view any metadata in the **Calculation Tools** pane, click **Reconnect** on the toolbar. If this does not work, you may have to process the cube or start the instance of Analysis Services.

5. Drag **Bikes** into the **Expression** box.
6. On the **File** menu, click **Save All**.

▶ Task 2: Browse the cube using the new named set

1. On the **Build** menu of Business Intelligence Development Studio, click **Deploy Mod05Lab**.
2. When deployment has successfully completed, click the **Browser** tab, and then click **Reconnect**.

> **Note**: If you see a warning that the structure of the cube has changed, click the link to process the cube. When the Process Cube window appears, click **Run**. When cube processing is complete, click **Close** twice.

3. Right-click inside the **Data** pane, and then click **Clear Results**.
4. In the **Measure Group** pane, expand **Measures | Fact Reseller Sales**, and then add the **Sales Amount - Fact Reseller Sales** measure to the **Drop Totals or Detail Fields Here** area.

5. Expand the **Dim Product Category** dimension, and then add the **English Product Category Name** hierarchy to the **Drop Row Fields Here** area.
6. Select the drop-down arrow next to **English Product Category Name** in the **Row** area, select the checkbox next to the **(All)** level to select all members of this level (if it is not already selected), and then click **OK**.
7. In the **Measure Group** pane, in the **Dim Product Category** dimension, right-click **Core Products**, and then select **Add to Subcube Area**.
8. Review the changes that have occurred in the **Sales Amount** column.
9. Turn off the 6234A-NY-SQL-01 virtual machine and discard any changes.

Results: After this exercise, you should have defined a named set and browsed the cube using the new named set.

Module 6: Customizing Cube Functionality

Lab: Customizing a Cube

Exercise 1: Implementing a KPI

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Administrator

1. In the Lab Launcher, next to **6234A-NY-SQL-01**, click **Launch**.
2. Log on to **NY-SQL-01** as **Administrator** using the password **Pa$$w0rd**.

▶ Task 2: Create a new SQL Server user

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, verify that **Server type** is set to **Database Engine** and **Server name** is set to **NY-SQL-01**, and then click **Connect**.
3. Maximize Microsoft SQL Server Management Studio.
4. Expand **NY-SQL-01 | Databases | AdventureWorksDW2008 | Security**, right-click on the **Users** folder, and then click **New User**.
5. The Database User – New window opens.
6. In the **User name** field, enter **sqlserver**.
7. In the **Login name** field, type **NY-SQL-01\sqlserver**.
8. In the **Default schema** field, enter **dbo**
9. Under the **Schemas owned by this user** section, ensure that **db_owner** and **db_securityadmin** are selected.
10. Under the **Database role membership section**, ensure that **db_owner** and **db_securityadmin** are selected.
11. Click **OK**.
12. Close Microsoft SQL Server Management Studio.

▶ Task 3: Define the Reseller Revenue KPI

1. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Business Intelligence Development Studio**.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** window, browse to the **E:\Mod06\Labfiles** folder, click the **Mod06Lab.dwproj** project file, and then click **Open**.
4. In the **Solution Explorer** pane, right-click **Mod06Lab**, and then click **Deploy**.
5. After deployment is complete, in Solution Explorer right-click **Adventure Works.cube**, click **Open**, and then click the **KPIs** tab.
6. On the toolbar of the **KPIs** tab, click **New KPI**.
7. In the **Name** box, type **Reseller Revenue**, and then click **Reseller Sales** in the **Associated measure group** list.
8. On the **Metadata** tab in the **Calculation Tools** pane, expand **Measures**, expand **Reseller Sales**, and then drag the **Sales Amount – Fact Reseller Sales** measure to the **Value Expression** box.
9. On the **Metadata** tab in the **Calculation Tools** pane, expand **Measures**, expand **Sales Quota**, and then drag the **Sales Amount Quota** measure to the **Goal Expression** box.
10. Verify that **Gauge** is selected in the **Status indicator** list, and then type the following MDX expression in the **Status expression** box:

```
Case
 When
  KpiValue("Reseller Revenue")/KpiGoal("Reseller Revenue")>=.95
   Then 1
 When
  KpiValue("Reseller Revenue")/KpiGoal("Reseller Revenue")<.95
   And
  KpiValue("Reseller Revenue")/KpiGoal("Reseller Revenue")>=.85
   Then 0
  Else -1
End
```

11. Verify that **Standard arrow** is selected in the **Trend indicator** list, and then type the following expression in the **Trend expression** box:

```
Case
 When IsEmpty
  (ParallelPeriod
   ([Date].[Calendar Time].[Calendar Year],1,
       [Date].[Calendar Time].CurrentMember))
  Then 0
 When  (
  KpiValue("Reseller Revenue") -
   (KpiValue("Reseller Revenue"),
    ParallelPeriod
     ([Date].[Calendar Time].[Calendar Year],1,
       [Date].[Calendar Time].CurrentMember))
      /
      (KpiValue ("Reseller Revenue"),
       ParallelPeriod
        ([Date].[Calendar Time].[Calendar Year],1,
         [Date].[Calendar Time].CurrentMember)))
       >=.02
  Then 1
   When(
    KpiValue("Reseller Revenue") -
     (KpiValue ( "Reseller Revenue" ),
      ParallelPeriod
       ([Date].[Calendar Time].[Calendar Year],1,
        [Date].[Calendar Time].CurrentMember))
       /
        (KpiValue("Reseller Revenue"),
         ParallelPeriod
          ([Date].[Calendar Time].[Calendar Year],1,
            [Date].[Calendar Time].CurrentMember)))
        <=.02
  Then -1
   Else 0
End
```

▶ Task 4: Browse the cube using the Reseller Revenue KPI

1. On the **File** menu, click **Save All**.
2. On the **Build** menu of Business Intelligence Development Studio, click **Deploy Mod06Lab**.
3. When deployment has successfully completed, click **Browser View** on the toolbar of the **KPIs** tab, and then click **Reconnect**.
4. In the top pane of the KPI browser, click **Sales Territory** in the **Dimension** list, click **Sales Territory** in the **Hierarchy** list, and then click **Equal** in the **Operator** list.
5. In the **Filter Expression** list, expand **All Sales Territories**, select **2**, **3**, and **4**, and then click **OK**.
6. In the top pane of the KPI browser, click **Date** in the **Dimension** list, click **Calendar** in the **Hierarchy** list, and then **Equal** in the **Operator** list.
7. In the **Filter Expression** list, expand **All**, select **1**, and then click **OK**.
8. Click anywhere in the **KPI Browser** pane to update the values for the **Reseller Revenue KPI**.
9. Keep SQL Server Business Intelligence Development Studio open for the next exercise.

Results: After this exercise you should have defined the reseller revenue KPI and browsed the cube using the reseller revenue KPI.

Exercise 2: Implementing an Action

▶ Task 1: Define the Drillthrough Action properties

1. In Cube Designer for the Adventure Works cube, click the **Actions** tab.
2. On the toolbar of the **Actions** tab, click **New Drillthrough Action**.
3. In the **Name** box, change the name of this action to **Internet Sales Details Drillthrough Action**.
4. In the **Measure group members** list, click **Internet Sales**.
5. In the **Drillthrough Columns** box, click **Internet Sales Order Details** in the **Dimensions** list.

6. In the **Return Columns** list, select the **Sales Order Number** checkbox, and then click **OK**.
7. Expand the **Additional Properties** box.
8. In the **Maximum Rows** box, type **10**.
9. In the **Caption** box, type **Drillthrough to Order Details…**.
10. On the **File** menu, click **Save All**.

▶ Task 2: Use the Drillthrough Action

1. On the **Build** menu of **Business Intelligence Development Studio**, click **Deploy Module06Lab**.
2. When deployment has successfully completed, click the **Browser** tab in Cube Designer for the Adventure Works cube, and then click **Reconnect**.
3. If there is any data in the **Data** pane, right click in the data area and then click **Clear Results**.
4. In the **Measure Group** pane, expand **Measures**, and then expand **Internet Sales**.
5. Right-click the **Sales Amount** measure and click **Add to Data Area**.
6. Expand the **Customer** dimension, drag the **Customer Name** user-defined hierarchy from the **Measure Groups** pane to the **Filter** pane.

> **Note**: The Filter pane is the section that has the Dimension, Hierarchy, Operator, and Filter Expression headers.

7. In the **Filter Expression** list, expand **All**, clear any selected check boxes, expand **Adam**, select **Powell**, and then click **OK**.

8. Click the data cell in the **Data** pane, then right-click that data cell and click **Drillthrough to Order Details**.
9. Click **Close** to close the Data Sample Viewer window.
10. Keep SQL Server Business Intelligence Development Studio open for the next exercise.

Results: After this exercise, you should have defined a drillthrough action and used the drillthrough action to return sales information.

Exercise 3: Implementing a Perspective

▶ Task 1: Define a Sales Summary perspective

1. In the Cube Designer for the Adventure Works cube, click the **Perspectives** tab.
2. On the toolbar of the **Perspectives** tab, click **New Perspective**.
3. Scroll all the way to the right and then change the name of the new perspective to **Sales Summary**.
4. Clear all check boxes for all objects.
5. Select the check boxes for the following objects:
   - **Fact Internet Sales Count** measure.
   - **Fact Reseller Sales Count** measure.

▶ Task 2: Browse the cube using the Sales Summary perspective

1. On the **File** menu, click **Save All**.
2. On the **Build** menu of Business Intelligence Development Studio, click **Deploy Module06Lab**.
3. When deployment has successfully completed, switch to the **Browser** tab, and then click **Reconnect**.
4. If there is any data in the **Data** pane, right-click in the data area and then click **Clear Results**.
5. If there are any dimensions in the **Filter** pane, right-click each dimension and click **Delete**.

6. In the **Perspective** list on the toolbar of the **Browser** tab, click **Sales Summary**.
7. In the **Measure Group** pane, expand **Measures**, expand **Internet Sales** and expand **Reseller Sales**.
8. Keep SQL Server Business Intelligence Development Studio open for the next exercise.

Results: After this exercise, you should have defined a perspective and browsed the cube using the new perspective.

Exercise 4: Implementing a Translation

▶ Task 1: Specify translations for the Time Dimension metadata

1. In Solution Explorer right-click **Date.dim**, click **View Designer**, and then click the **Translations** tab.
2. On the toolbar of the **Translations** tab, click **New Translation**, click **French (France)** in the **Select Language** dialog box, and then click **OK**.
3. In the row for the **Caption** object for the **Date** dimension, type **Date** in the **French (France)** translation column.
4. In the row for the **Caption** object for the **Month Number of Year** attribute, type **Mois d'Année** in the **French (France)** translation column.
5. Click the ellipsis (**...**) for the **French (France)** translation for the **Month Number of Year** attribute. The **Attribute Data Translation** dialog box appears.
6. In the **Translation columns** list, click **FrenchMonthName**, and then click **OK**.
7. On the **File** menu, click **Save All**.
8. Close the **Date.dim** dimension designer.

▶ Task 2: Specify translations for the Adventure Works cube metadata

1. Switch to Cube Designer for the Adventure Works cube, and then click the **Translations** tab.
2. On the toolbar of the **Translations** tab, click **New Translation** click **French (France)** in the **Select Language** dialog box, and then click **OK**.
3. In the row for the **Caption** object for the **Internet Sales** measure group, type **Ventes d'Internet** in the **French (France)** translation column.
4. In the row for the **Caption** object for the **Sales Amount** measure of the **Internet Sales** measure group, type **Quantité de Ventes d'Internet** in the **French (France)** translation column.
5. On the **File** menu, **Save All**.

▶ Task 3: Browse the cube using the new translations

1. On the **Build** menu of Business Intelligence Development Studio, click **Deploy Module06Lab**.
2. When deployment has successfully completed, click the **Browser** tab, and then click **Reconnect**.
3. Remove any hierarchies and measures from the **Data** pane, click **Adventure Works** in the **Perspectives** list, and then verify that **Default** appears in the **Language** list.
4. In the **Metadata** pane, expand the **Date** dimension, right-click **Date.Mois d'Ainee**, and then click **Add to Row Area**.
5. On the toolbar, click **French (France)** in the **Language** list.
6. In the **Measure Group** pane, expand **Measures**, expand **Ventes d'Internet**, right-click **Quantité de Ventes d'Internet** and then select **Add to Data Area**.
7. Turn off the 6234A-NY-SQL-01 virtual machine and discard any changes.

Results: After this exercise, you should have specified translations for the time dimension metadata, specified translations for the Adventure Works cube metadata, and browsed the cube using the new translations.

Module 7: Deploying and Securing an Analysis Services Database

Lab: Deploying and Securing an Analysis Services Database

Exercise 1: Deploying an Analysis Services Database

▶ Task 1: Start the 6434A-NY-SQL-01 virtual machine and log on as Student

1. In the Lab Launcher, next to **6434A-NY-SQL-01,** click **Launch**.
2. Log on as **Student** with the password of **Pa$$w0rd**.

▶ Task 2: Install the Adventure Works Data Warehouse database

1. Click **Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio**.
2. On the **Connect to Server** dialog box, click **Connect**.
3. In Microsoft SQL Server Management Studio, click **File | Open | File**. In the **Open File** dialog box, browse to **E:\MOD07\Labfiles\Starter\AdventureWorks Data Warehouse**, click **instawdwdb.sql**, and then click **Open**.
4. In Microsoft SQL Server Management Studio, on the toolbar, click **Execute**.
5. Wait for the query to finish executing, and then close Microsoft SQL Server Management Studio.

▶ Task 3: Use the Deployment Wizard to create a deployment script

1. Click **Start | All Programs | Microsoft SQL Server 2008 | Analysis Services | Deployment Wizard**.
2. On the Welcome to the Analysis Services Deployment Wizard page, click **Next**.
3. On the Specify Source Analysis Services Database page, click the ellipsis **(...)** button , and then navigate to **E:\MOD07\Labfiles\Starter\Adventure Works OLAP\bin**.

4. Click **Adventure Works OLAP.asdat**abase, click **Open**, and then click **Next**.
5. On the Installation Target page, in the **Database** field, type **Adventure Works OLAP** copy, and then click **Next**.
6. On the Specify Options for Partitions and Roles page, click **Next**.
7. On the Specify Configuration Properties page, click **Next**.
8. On the Specify Processing Options page, click **Next**.
9. On the Confirm Deployment page, select **Create deployment script**, and then click **Next**.
10. On the **Deploying Database** page, when the deployment script is finished, click **Next**.
11. On the Deployment Complete page, click **Finish**.

▶ Task 4: Review and run the deployment script

1. Click **Start | All Programs | Microsoft SQL Server 2008 | SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, in the **Server type** list, click **Analysis Services**, type **NY-SQL-01** in the **Server name** field, and then click **Connect**.
3. In Microsoft SQL Server Management Studio, click **File | Open | File**.
4. In the **Open File** dialog box, navigate to **E:\MOD07\Labfiles\Starter\Adventure Works OLAP\bin**, click **Adventure Works OLAP copy Script.xmla**, and then click **Open**.
5. In the **Details** pane, review the script. Notice the new database name and ID of **Adventure Works OLAP Copy**.
6. On the toolbar, click **Execute**. Wait for the query to finish executing.
7. In Object Explorer, expand **Databases** and verify that the **Adventure Works OLAP Copy** database has been created. If the database is not listed, click the **Refresh** button on the Object Explorer toolbar.
8. Keep Microsoft SQL Server Management Studio open for the next exercise.

Results: After this exercise, you should have successfully created, reviewed and executed the Adventure Works OLAP copy deployment script.

Exercise 2: Securing an Analysis Services Database

▶ Task 1: Set up the lab environment

1. Click **Start**, right-click **Command Prompt**, and then click **Run as Administrator**.
2. In the **User Account Control** dialog box, in the **Password** field, type **Pa$$w0rd**, and then press ENTER.
3. Type **E:**, and then press ENTER.
4. Type **CD MOD07\Labfiles\Starter**, and then press ENTER.
5. Type **Setup,** and then press ENTER.
6. Wait for the command to finish, type **exit**, and then press ENTER.

▶ Task 2: Add a user to the Analysis Services Server role

1. In SQL Server Management Studio, in the **Object Explorer** pane, refresh the **Databases** folder and verify that the **Adventure Works OLAP** database has been created.
2. Right-click **NY-SQL-01**, and then click **Properties**.
3. In the **Analysis Server Properties** dialog box, in **Select a page**, click **Security**.
4. Click **Add**, and then in the **Select Users or Groups** dialog box, in the **Enter the object names to select** field, type **ASAdmin**, click **Check Names**, and then click **OK**.
5. Click **OK** to close the **Analysis Server Properties** dialog box.
6. Close Microsoft SQL Server Management Studio.

▶ Task 3: Add a new role with the ability to process the database

1. Click **Start | All Programs | Microsoft SQL Server 2008 | SQL Server Business Intelligence Development Studio**.
2. Click **File | Open | Project/Solution**. In the **Open Project** dialog box, browse to **E:\MOD07\Labfiles\Starter**, click **Adventure Works OLAP.sln**, and then click **Open**.
3. In Solution Explorer, right-click **Roles**, and then click **New Role**.

4. In Solution Explorer, click **Role.role**.
5. In the **Properties** pane, change the **File Name** to **DB Process Role.role**, and then press ENTER. When prompted, click **Yes** to change the object name.
6. On the **General** tab of the Role Designer, select **Process Database** and **Read definition**.
7. Click the **Membership** tab, and then click **Add**.
8. In the **Select Users or Groups** dialog box, in the **Enter the object names to select** field, type **ProcessAdmin**, click **Check Names**, and then click **OK**.
9. Click **File | Save All**, and then close **Role Designer**.

▶ Task 4: Add a new role with specific cube and dimension permissions

1. In Solution Explorer, right-click **Roles**, and then click **New Role**.
2. In Solution Explorer, click **Role 1.role**.
3. In the **Properties** pane, change the **File Name** to **Marketing Manager Role.role**, and then press ENTER. When prompted, click **Yes** to change the object name.
4. In **Role Designer,** click the **Membership** tab, and then click **Add**.
5. In the **Select Users or Groups** dialog box, in the **Enter the object names to select** field, type **DBradley**, click **Check Names**, and then click **OK**.
6. Click the **Cubes** tab, and then in the **Access** column, choose **Read**.
7. Click the **Dimensions** tab. Notice that the role has been automatically granted Read access to all dimensions in the cube.
8. Click the **Cell Data** tab, and then select the **Enable read permissions** check box.
9. In the **Allow reading of cube content** field, type the following MDX code:

```
NOT Measures.CurrentMember IS [Measures].[Sales Amount Quota]
```

> **Note:** You can copy this code from E:\MOD07\Labfiles\Starter\CellDataMDX.txt.

10. Click **File | Save All**.

▶ Task 5: Deploy the solution

1. In Solution Explorer, right-click the **Adventure Works OLAP** solution, and then click **Properties**.
2. In the **Adventure Works OLAP Property Pages** dialog box, in the left pane, click **Deployment**.
3. Under **Options**, for **Processing Option**, select **Full**, for **Deployment Mode**, select **Deploy All**. Under **Target**, change the **Database** field to **Adventure Works OLAP Secure**, and then click **OK**.
4. In Solution Explorer, under **Data Sources**, double-click **Adventure Works Data Warehouse**.
5. In **Data Source Designer**, on the **Impersonation Information** tab, click **Use a specific Windows user name and password**.
6. In the **User name** field, type **administrator**, in the **Password** field, type **Pa$$w0rd**, and then click **OK**.
7. In Solution Explorer, right-click **Adventure Works OLAP**, and then click **Deploy**.
8. Wait for the **Deployment Completed Successfully** message.

▶ Task 6: Test user security by using Business Intelligence Development Studio

1. Click the **Role Designer**, and then click the **Cell Data** tab.
2. Click **Test cube security**.
3. Notice the information line just below the toolbar stating that you are using the **Marketing Manager Role** credentials.
4. In the **Measure Group** pane, expand **Measures**, expand **Sales Quotas**, right-click **Sales Amount Quota**, and then click **Add to Data Area**.
5. In the **Measure Group** pane, under **Measures**, expand **Internet Sales**, expand **Sales**, right-click **Internet Sales-Sales Amount**, and then click **Add to Data Area**.
6. Notice that the value of **Sales Amount Quota** is not visible.
7. Click the **Close** button to close the Adventure Works Cube [Browse] window.

8. In Solution Explorer, right-click **Adventure Works Cube.cube**, and then click **View Designer**.
9. In the **Cube Designer**, click the **Browser** tab.
10. On the Browser page, click the **Change User** button.
11. In the **Security Context – Adventure Works Cube** dialog box, click **Roles**, select the **Marketing Manager Role**, and then click **OK**. Click **OK** to close the **Security Context – Adventure Works Cube** dialog box.
12. In the **Measure Group** pane, expand **Measures**, expand **Sales Quotas**, right-click **Sales Amount Quota**, and then click **Add to Data Area**.
13. In the **Measure Group** pane, under **Measures**, expand **Internet Sales**, expand **Sales**, right-click **Internet Sales-Sales Amount**, and then click **Add to Data Area**.
14. Notice that the value of **Sales Amount Quota** is not visible.
15. On the Browser page, click the **Change User** button.
16. In the **Security Context – Adventure Works Cube** dialog box, click the **Roles** dropdown, clear **Marketing Manager Role**, select **DB Process Role**, and then click **OK**. Click **OK** to close the **Security Context – Adventure Works Cube** dialog box.
17. Notice that you cannot browse the cube.
18. Close Business Intelligence Development Studio.
19. Turn off 6234A-NY-SQL-01 and delete changes.

Results: After this exercise, you should have successfully created and tested analysis database security roles.

Module 8: Maintaining a Multidimensional Solution

Lab: Maintaining a Multidimensional Solution

Exercise 1: Configuring Processing

▶ Task 1: Start the 6234A-NY-SQL-01 virtual machine and log on as Student

1. In the Lab Launcher, next to **6234A-NY-SQL-01,** click **Launch**.
2. Log on as **Student** with the password of **Pa$$w0rd**.

▶ Task 2: Install the Adventure Works Data Warehouse database and setup the lab environment

1. Click **Start | All Programs | Microsoft SQL Server 2008**, right-click **SQL Server Management Studio**, and then click **Run as Administrator**.
2. In the **User Account Control** dialog box, in the **Password** field, type **Pa$$w0rd**, and then press ENTER.
3. On the **Connect to Server** dialog box, click **Connect**.
4. In Microsoft SQL Server Management Studio, click **File | Open | File**. In the **Open File** dialog box, browse to **E:\MOD08\Labfiles\Starter\AdventureWorks Data Warehouse**, click **instawdwdb.sql**, and then click **Open**.
5. In Microsoft SQL Server Management Studio, on the toolbar, click **Execute**.
6. Wait for the query to finish executing, and then close Microsoft SQL Server Management Studio.
7. Click **Start**, right-click **Command Prompt**, and then click **Run as Administrator**.
8. In the **User Account Control** dialog box, in the **Password** field, type **Pa$$w0rd**, and then press ENTER.
9. Type **E:**, and then press ENTER.

10. Type **CD MOD08\Labfiles\Starter**, and then press ENTER.
11. Type **Setup,** and then press ENTER**.**
12. Wait for the command to finish, type **exit**, and then press ENTER.

▶ Task 3: Write a script to perform a full process of the Reseller Sales partition

1. Click **Start | All Programs | Microsoft SQL Server 2008**, right-click **SQL Server Management Studio**, and then click **Run as Administrator**.
2. In the **User Account Control** dialog box, in the **Password** field, type **Pa$$w0rd**, and then press ENTER.
3. In the Connect to Server window, click **Analysis Services** in the **Server type** box, type **NY-SQL-01** in the **Server name** box, and then click **Connect**.
4. In Object Explorer, expand **Databases | Adventure Works OLAP | Data Sources**.
5. Right-click **Adventure Works Data Warehouse**, and then click **Properties**.
6. Under **Security Settings**, click the browse button next to **ImpresonateServiceAccount**.
7. In the **Impersonation Information** dialog box, click **Use a specific Windows user name and password**, in the **User name** field, type **Administrator**, in the **Password** field, type **Pa$$w0rd**, and then click **OK** twice.
8. In Object Explorer, right click **Adventure Works OLAP**, and then click **Process**. In the **Process Database – Adventure Works OLAP** dialog box, click **OK**. Wait for the processing to complete, and then click **Close**.
9. Click the **Auto Hide** button to hide the Object Explorer window.
10. In SQL Server Management Studio, click the **Analysis Services XMLA Query** button, and then in the Connect to Analysis Services window, click **Connect**.

11. In the query window, type the following code:

```
<Batch
xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <Parallel>
    <Process      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Object>
        <DatabaseID>Adventure Works OLAP</DatabaseID>
        <CubeID>Adventure Works DW</CubeID>
        <MeasureGroupID>Reseller Sales</MeasureGroupID>
        <PartitionID>Reseller Sales</PartitionID>
      </Object>
      <Type>ProcessFull</Type>
      <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
    </Process>
  </Parallel>
</Batch>
```

Note: The completed query is located in E:\MOD08\Labfiles\Solution\ProcessPartition.xmla.

12. Click **File**, and then click **Save XMLAQuery1.xmla As**.
13. In the **Save As** dialog box, browse to the **E:\MOD08\Labfiles\Starter folder**, then in the **File name** box, type **ProcessPartition.xmla**, and then click **Save**.
14. Keep Microsoft SQL Server Management Studio open for the next task.

▶ Task 4: Update the data in the relational Reseller fact table and then perform a full process

1. In SQL Server Management Studio, on the **File** menu, click **Open**, and then click **File**.
2. In the **Open File** dialog box, browse to the **E:\MOD08\Labfiles\Starter** folder, click **MDXQuery.mdx** and then click **Open**. In the **Connect to Analysis Services** dialog box, click **Connect**.
3. Review the query. Then, on the toolbar, click **Execute**. Note that the value of the **Reseller Sales-Sales Amount** is **$80,450,596.98**.
4. On the **File** menu, point to **Open**, and then click **File**.
5. In the **Open File** dialog box, browse to the **E:\MOD08\Labfiles\Starter** folder, click **InsertNewFact.sql**, and then click **Open**.
6. In the **Connect to Database Engine** dialog box, click **Connect**.
7. Review the SQL script and then click **Execute**. This script adds a new row into the **FactResellerSales** table.
8. Click the query tab for the **MDXQuery.mdx** file, and then click **Execute**. Note that the **Reseller Sales-Sales Amount** value has not changed.

> **Note**: If you can not see the query tab with the MDXQuery.mdx file, click the Active Files icon in the Query pane.

9. Click the **ProcessPartition.xmla** tab, and then click **Execute**. This will perform a full process of the **Reseller Sales** partition.
10. Click the **MDXQuery.mdx** tab, and then click **Execute**. Notice that the value of the **Reseller Sales** amount is now **$80,450,935.98**.
11. Keep SQL Server Management Studio open for the next exercise.

> **Results:** After this exercise, you should have successfully created, reviewed and executed the ProcessPartition.xmla script.

Exercise 2: Implementing Logging and Monitoring

▶ Task 1: Configure Logging for an Analysis Services instance

1. In Microsoft SQL Server Management Studio, restore Object Explorer if it is still hidden. In Object Explorer, right-click the **NY-SQL-01** instance of **Analysis Services**, and then click **Properties**.
2. In the **Analysis Server Properties** dialog box, on the General page, in the **Value** column, in the **Log\QueryLog\CreateQueryLogTable** row, click **true**.
3. On the General page, in the **Value** column, in the **Log\QueryLog\QueryLogConnectionString** row, click the cell, then click the box that appears on the right side of the cell.
4. In the **Connection Manager** dialog box, in the **Server name** box, type **NY-SQL-01**, and then verify that **Use Windows Authentication** is selected.
5. In the **Connect to a database** area, verify that **Select or enter a database name** is selected, click **AdventureWorksDW**, and then click **OK**.
6. On the **General** page, in the **Value** column, in the **Log\QueryLog\QueryLogSampling** row, type **2**.

> **Note**: Setting a sampling rate of one in every two queries is too frequent for most production environments; it is used here to overcome the low query volume on a test server.

7. Click **OK**.
8. If a message appears stating that changes will take place when Analysis Services is restarted, click **OK**.
9. In Object Explorer, right-click the **NY-SQL-01** instance of **Analysis Services**, and then click **Restart**. When prompted to confirm the restart, click **Yes**.
10. Leave SQL Server Management Studio open so that you can use it later in this exercise.

▶ Task 2: Simulate activity in the database

1. Click **Start | All Programs | Microsoft SQL Server 2008 | Performance Tools, and then click SQL Server Profiler**.
2. In SQL Server Profiler, on the **File** menu, click **Open**, and then click **Trace File**.
3. Browse to the **E:\MOD08\Labfiles\Starter** folder, click **playbacktrace.trc**, and then click **Open**.
4. Press F5 to replay the trace. When prompted, click **Connect**.
5. In the **Replay Configuration** dialog box, click **Replay events using multiple threads**. This option optimizes performance and disables debugging, and then click **OK**.
6. Wait for the replay to complete, and then close SQL Server Profiler.

▶ Task 3: Run usage-based optimization for the Internet Sales partition

1. In SQL Server Management Studio, in Object Explorer, expand **Databases**, expand **Adventure Works OLAP**, expand **Cubes**, expand **Adventure Works Cube**, expand **Measure Groups**, expand **Internet Sales**, expand **Partitions**, right-click **Internet Sales**, and then click **Usage Based Optimization**.
2. On the Welcome to the Usage-Based Optimization Wizard page, click **Next**.
3. On the Specify Query Criteria page, click **Next**.
4. On the Review the Queries that will be Optimized page, review the queries, and then click **Next**.
5. On the Review Aggregation Usage page, click **Next**.
6. On the Specify Object Counts page, click **Count**, wait for the counts to finish, and then click **Next**.

7. On the Set Aggregation Options page, click **Performance gain reaches**, verify that the default value of **30%** is selected, and then click **Start**.
8. Wait for the aggregation design to finish, and then click **Next**.
9. On the Completing the Wizard page, click **Finish**.
10. Leave SQL Server Management Studio open for the next exercise.

Results: After this exercise, you should have successfully configured logging, created activity in the database, and run usage-based optimization for a partition.

Exercise 3: Backing Up and Restoring an Analysis Services Database

▶ Task 1: Back up the Adventure Works OLAP database

1. In SQL Server Management Studio, in Object Explorer, right-click the **Adventure Works OLAP** database, and then click **Back Up**.
2. On the General page, change the **Backup file** to **E:\MOD08\Labfiles\Starter\Adventure Works OLAP.abf**.
3. In the **Options** section, select the **Allow file overwrite** check box, and then clear the **Encrypt backup file** check box.
4. Click **OK**. When the backup completes, the Backup Database window will automatically close.

▶ Task 2: Create a restore script for the Adventure Works OLAP database

1. In Object Explorer, right-click the **Adventure Works OLAP** database, and then click **Restore**.
2. In the **Restore Target** section, in the **Restore database** field, click **Adventure Works OLAP**.
3. In the **Restore source** section, in the **Backup file** field, type **E:\MOD08\Labfiles\Starter\Adventure Works OLAP.abf**.
4. In the **Options** section, select the **Allow database overwrite** check box.
5. On the toolbar, click **Script**.
6. Close the Restore Database window by clicking **Cancel**.
7. Review the XMLA script created.
8. Click **File**, and then click **Save XMLAQuery2.xmla As**.
9. In the **Save As** dialog box, browse to the **E:\MOD08\Labfiles\Starter** folder, then in the **File name** box, type **AWOLAPRestore.xmla**, and then click **Save**.

▶ Task 3: Test a restore of the Adventure Works OLAP database

1. In the Object Explorer, right-click the **Adventure Works OLAP** database, and then click **Delete**.
2. In the Delete Objects window, click **OK**. Verify that the **Adventure Works OLAP** database has been deleted.
3. Click the **AWOLAPRestore.xmla** script tab, and then click **Execute**.
4. Wait for the query to finish running.
5. In the Object Explorer, right-click the **Databases** folder, and then click **Refresh**. Expand **Databases** and verify that the **Adventure Works OLAP** database has been successfully restored.
6. Turn off 6234A-NY-SQL-01 and delete changes.

Results: After this exercise, you should have successfully backed up an Analysis Services database, created a restore script for the database, and tested restoring the database.

Module 9: Introduction to Data Mining

Lab: Implementing Data Mining

Exercise 1: Creating a Data Mining Structure

▶ Task 1: Start the 6434A-NY-SQL-01 virtual machine and log on as Student

1. In the Lab Launcher, next to **6434A-NY-SQL-01,** click **Launch**.
2. Log on as **Student** with the password of **Pa$$w0rd**.

▶ Task 2: Install the Adventure Works Data Warehouse database

1. Click **Start | All Programs | Microsoft SQL Server 2008**, right-click **SQL Server Management Studio**, and then click **Run as Administrator**.
2. In the **User Account Control** dialog box, in the **Password** field, type **Pa$$w0rd**, and then press ENTER.
3. On the **Connect to Server** dialog box, click **Connect**.
4. In Microsoft SQL Server Management Studio, click **File | Open | File**. In the **Open File** dialog box, browse to **E:\MOD09\Labfiles\Starter\AdventureWorks Data Warehouse**, click **instawdwdb.sql**, and then click **Open**.
5. In Microsoft SQL Server Management Studio, on the toolbar, click **Execute**.
6. Wait for the query to finish executing, and then close Microsoft SQL Server Management Studio.

▶ Task 3: Browse the data mining project and data source view

1. Click **Start |All Programs | Microsoft SQL Server 2008**, right-click **SQL Server Business Intelligence Development Studio**, and then click **Run as Administrator**.
2. In the **User Account Control** dialog box, in the **Password** field, type **Pa$$w0rd**, and then press ENTER.
3. On the **File** menu, click **Open**, and then click **Project/Solution**. Browse to **E:\MOD09\Labfiles\Starter**, click **Adventure Works Data Mining.sln**, and then click **Open**.
4. In Solution Explorer, under **Data Sources,** double-click **Adventure Works DM.ds.**
5. In the Data Source Designer, on the **Impersonation Information** tab, click **Use a specific Windows user name and password**, in the **User name** field, type **Administrator**, in the **Password** field, type **Pa$$w0rd**, and then click **OK.**
6. In Solution Explorer, under **Data Source Views**, double-click **Adventure Works DM DSV.dsv**.
7. Review the tables and views included in the data source view.

▶ Task 4: Create a data mining structure

1. In Solution Explorer, right-click **Mining Structures**, and then click **New Mining Structure**.
2. On the Welcome to the Data Mining Wizard page, click **Next**.
3. On the Select the Definition Method page, verify that **From existing relational database or data warehouse** is selected, and then click **Next**.
4. On the Create the Data Mining Structure page, in the **Which data mining technique do you want to use** box, click **Microsoft Time Series**, and then click **Next**.
5. On the Select Data Source View page, verify that the **Adventure Works DM DSV** is selected, and then click **Next**.
6. On the Specify Table Types page, select the **Case** check box for the **vTimeSeries** table, and then click **Next**.

7. On the Specify the Training Data page, click the **Key** check boxes for the **ModelRegion** and **TimeIndex** columns, click the **Input** and **Predictable** column check boxes for the **Quantity** and **Amount** columns, and then click **Next**.

> **Note:** This configuration will only work if the **TimeIndex** column is unique across each individual value in the **ModelRegion** column.

8. On the Specify Columns' Content and Data Type page, click **Next**.
9. On the Completing the Wizard page, change the **Mining structure name** to **Sales Forecasting**, change the **model name** to **Time Series Forecast**, and then click **Finish**

▶ Task 5: Modify a data mining structure

1. In the **Sales Forecasting.dmm** designer, click the **Mining Models** tab.
2. Right-click **Time Series Forecast** column heading, and then click **Set Algorithm Parameters**.
3. In the **Algorithm Parameters** dialog box, type **{12}** in the **Value** column of the **PERIODICITY_HINT** row, and then click **OK**.

> **Note:** The hint tells the algorithm that the data repeats itself every twelve periods (the data is organized in months per year).

> **Results:** After this exercise, you should have successfully browsed the data mining project and data source views, created a data mining structure, and modified a data mining structure.

Exercise 2: Adding a Data Mining Model

▶ Task 1: Add a Naive Bayes mining model to the Targeted Mailing mining structure

1. In **Solution Explorer**, right-click the **Targeted Mailing.dmm** mining structure, and then click **View Designer**.
2. In the **Targeted Mailing.dmm** designer, click the **Mining Models** tab. If prompted to build and deploy first, click **Yes**. If prompted to process **TM_Decision_Tree**, click **No**.
3. On the **Mining Model** toolbar, click the **Create a related mining model** button.
4. In the **New Mining Model** dialog box, in the **Model name** box, type **TM_Naive_Bayes**, in the **Algorithm name** box, click **Microsoft Naive Bayes**, and then click **OK**.
5. Read the message stating that some columns will not be supported, and then click **Yes**.
6. Notice that the **Yearly Income** and **Age** columns have a value of **Ignore** in the **TM_Naive_Bayes** column.

▶ Task 2: Add a Clustering mining model to the Targeted Mailing mining structure

1. On the **Mining Model** toolbar, click **Create a related mining model**.
2. In the **New Mining Model** dialog box, in the **Model name** box, type **TM_Clustering**, in the **Algorithm name** box, click **Microsoft Clustering**, and then click **OK**.

Results: After this exercise, you should have successfully added two data mining models to the project.

Exercise 3: Exploring Data Mining Models

▶ Task 1: Deploy the data mining solution

1. In Solution Explorer, right-click the **Adventure Works Data Mining** solution, and then click **Deploy**.
2. Wait for the **Deployment Completed Successfully** message.

▶ Task 2: View the data mining model in the Sales Forecasting mining structure

1. Click the Sales **Forecasting.dmm** designer tab, and then click the **Mining Model Viewer** tab. If you are warned about a reference, click **OK**.
2. In the **Mining Model Viewer**, verify that **Time Series Forecast** is selected in the **Mining Model** box and that **Microsoft Time Series Viewer** is selected in the **Viewer** box.
3. In the **Mining Model Viewer** tab, click the **Charts** tab.
4. To the right of the **Chart** view, clear the check boxes for the **M200 Europe:Quantity**, **M200 North America:Quantity**, and **M200 Pacific:Quantity** series.

> **Tip:** You can click the Auto Hide buttons to hide the various windows in the Microsoft® Visual Studio® environment and make it easier to view the designer.

5. Review the chart displayed. Notice that the actual data is to the left of the vertical line and noted by solid data lines, while predicted data is to the right of the vertical line and is noted with dotted lines.

▶ Task 3: View the data mining models in the Targeted Mailing mining structure

1. Click the **Targeted Mailing.dmm** designer tab, and then click the **Mining Model Viewer** tab.
2. In the **Mining Model Viewer**, verify that **TM_Decision_Tree** is selected in the **Mining Model name** box and that **Microsoft Tree Viewer** is selected in the **Viewer** box. You may have to click **TM_Decision_Tree** to refresh the view and load the data.
3. On the **Decision Tree** tab, in the **Background** box, click **1**. This will change the background color of the nodes. The darker the color, the stronger the link to bicycle purchase.
4. In the **Data** area, use the scroll bars to view all of the nodes. Notice that people under the age of 34 who own no cars are most likely to buy a bicycle.
5. On the **Mining Model Viewer** page, click the **Dependency Network** tab.
6. All links are currently shown. Move the **All Links** slider down three stops. Notice that several of the dependency lines disappear.
7. In the **Data** area, click the **Bike Buyer** node. Review the color legend at the bottom of the page.
8. Move the **All Links** slider down to **Strongest Links** to find the most significant factor in determining whether a customer will purchase a bicycle.

Results: After this exercise, you should have successfully deployed the data mining solution and viewed the data mining models.

Exercise 4: Validating Data Mining Models

▶ Task 1: View the accuracy of the mining models in the Targeted Mailing mining structure

1. On the **Targeted Mailing.dmm** designer, click the **Mining Accuracy Chart** tab.
2. In the **Select data set to be used for Accuracy Chart** area, click **Specify a different data set** if not already selected, and then click the ellipsis **(…)** button.
3. In the **Specify Column Mapping** dialog box, under **Select Input Table(s)**, click **Select Case Table**.
4. In the **Select Table** window, in the **Table/View Name** box, click **vTargetMail (dbo)**, and then click **OK**. Click **Close**.
5. In the **Select predictable mining model columns to show in the lift chart** area, in the **Predict Value** box of any mining model, click **1**.
6. Click the **Lift Chart** tab.
7. After the **Lift Chart** has been built successfully, notice that the **TM_Decision_Tree** model is the most accurate.
8. Click the **Classification Matrix** tab.
9. Review the matrix data and notice that it also confirms that the **TM_Decision_Tree** model is the most accurate.
10. Turn off 6234A-NY-SQL-01 and delete changes.

Results: After this exercise, you should have successfully validated the accuracy of the data mining models.