Microsoft

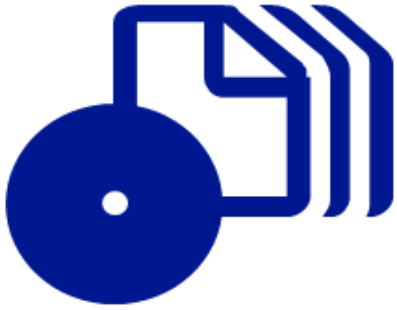
Microsoft® SQL Server® 2008 Analysis Services

Scott Cameron
Hitachi Consulting

eBook + exercises

Step by Step

How to access your CD files



The print edition of this book includes a CD. To access the CD files, go to http://aka.ms/626201/files, and look for the Downloads tab.

Note: Use a desktop web browser, as files may not be accessible from all ereader devices.

Questions? Please contact: mspinput@microsoft.com

Microsoft Press

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2009920805

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 4 3 2 1 0 9

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to mspinput@microsoft.com.

Acquisitions Editor: Ken Jones
Developmental Editor: Sally Stickney
Project Editor: Lynn Finnel
Editorial Production: Custom Editorial Productions, Inc.
Technical Reviewer: John Welch. Technical Review services provided by Content Master, a member of CM Group, Ltd.
Cover: Tom Draper Design

Body Part No. X14-72193

Contents at a Glance

Table of Contents

Acknowledgments

I am very grateful for the assistance and support I received from Hitachi Consulting while writing SQL Server 2008 Analysis Services Step by Step. Kevin Davis managed the project, provided daily motivation, and did a masterful job of juggling the schedule, coordinating the efforts of the co-authors and contributing authors, and communicating with Microsoft Press. Co-authors Dave DuVarney, Joe Kasprzak, and Bryan Smith; and contributing authors Harlan Smith, Phillip Duong, Shay Jones, Robert Brawn, and Renee De Voursney generously contri-buted their personal time and eased my workload. Reed Jacobsen, my long-time mentor, and Stacia Misner, Paul Turley, Richard Osbourn, and Ryan Clay contributed their technical expertise. Hilary Feier gave encouragement and, along with Carr Krueger, Drew Naukum, and Lance Baldwin, arranged the support from Hitachi Consulting that granted to me the time to write this book. David Han and Mary Gianopoulos performed additional project management.

I appreciate the opportunity to write this book given to me by Microsoft Press and the guidance I received from the team of editors. Thanks to Ken Jones, program manager, and Sally Stickney, developmental editor, who sponsored and initiated the project; and to Lynn Finnel, project editor, who showed great patience and assembled a great team of editors. Thanks also to technical editors John Welch, Matthew Roche, and Phillippe Freddi, who helped ensure that the explanations and procedures are correct and understandable, and editors Megan Smith-Creed, Becka McKay, Julie Hotchkiss, and Tom Lewis.

Finally, thanks to my wife Tarya, who patiently endured the many days when the alarm clock went off very early in the morning.

My apologies to anyone I may have overlooked. To all who have contributed to the creation of this book, my most sincere thanks.

Scott Cameron
March 2009

Introduction

Microsoft SQL Server 2008 Analysis Services is the multidimensional online analytical processing (OLAP) component of Microsoft SQL Server 2008 that integrates relational and OLAP data for business intelligence (BI) analytical solutions. The goal of this book is to show you how to use the tools and features of Analysis Services so you can easily create, manage, and share OLAP cubes within your organization. Step-by-step exercises are included to prepare you for developing your own BI solutions.

To help you learn the many features of Analysis Services, this book is organized into four parts. Part I, "Understanding Business Intelligence and Analysis Services," introduces BI, multidimensional analysis, and OLAP and explains how Analysis Services implements the benefits of OLAP. Part II, "Design Fundamentals," teaches you how to design data sources, data source views, dimensions, and cubes. Part III, "Advanced Design," shows you how to use Multidimensional Expressions (MDX) and aggregate functions to perform complex calculations and summarizations, and to create key performance indicators (KPIs). In addition, this part covers special Analysis Services features for advanced dimension design, financial analysis, globalization, and a variety of interactions that extend the analytical capabilities of cubes. You will also learn how to create analytical reports using Microsoft Office Excel and SQL Server 2008 Reporting Services. Part IV, "Production Management," explains how to use security to control access to cubes and to restrict the data that a particular user can see, how to design partitions to manage database scalability, and how to manage and monitor Analysis Services databases.

Finding Your Best Starting Point in This Book

This book covers the full life cycle of an Analysis Services solution from development to deployment. If you're responsible only for certain activities, you can choose to read the chapters that apply to your situation and skip the remaining chapters. Use the following table to find your best starting point.

If you are	Follow these steps
An information consumer who uses OLAP to make decisions	**1.** Install the practice files as described in the section, "Installing and Using the Practice Files." **2.** Work through Parts I and II to become familiar with the basic capabilities of Analysis Services. **3.** Skim chapters of interest to you in Part III to understand how additional features might meet your analytical requirements.
A BI analyst who develops OLAP models and prototypes for business analysis	**1.** Install the practice files as described in the section, "Installing and Using the Practice Files." **2.** Skim Part I to review BI concepts and learn how Analysis Services implements OLAP. **3.** Work through Part II to develop the necessary skills to create a prototype cube. **4.** Review the chapters that interest you in Parts III and IV to learn about advanced features of Analysis Services, to understand how cubes are accessed by users, and learn how cubes are managed after they are put into production.
An administrator who maintains server resources or production migration processes	**1.** Install the practice files as described in the section, "Installing and Using the Practice Files." **2.** Skim Parts I, II, and III to understand the functionality that is included in Analysis Services. **3.** Work through Part IV to learn how to manage and secure cube access and content on the server as well as how to configure, monitor, and manage server components and performance.
A BI architect who designs and develops analytical solutions	**1.** Install the practice files as described in the section, "Installing and Using the Practice Files." **2.** Complete Part I to become familiar with the benefits of Analysis Services. **3.** Work through Parts II and III to learn how to create dimensions and cubes and how to implement advanced design techniques. **4.** Complete Part IV to understand how to design cubes that implement the security, performance, and processing features of Analysis Services.

Conventions and Features in This Book

This book presents information using conventions designed to make the information readable and easy to follow. Before you start, read the following list, which explains conventions you'll see throughout the book and points out helpful features that you might want to use.

Conventions

- Each exercise is a series of tasks. Each task is presented as a series of numbered steps (1, 2, and so on).
- Notes labeled "tip" provide additional information or alternative methods for completing a step successfully.
- Notes labeled "important" alert you to information you need to check before continuing.
- Text that you type appears in **bold**.
- A plus sign (+) between two key names means that you must press those keys at the same time. For example, "Press Alt+Tab" means that you hold down the Alt key while you press the Tab key.

Hardware and Software Requirements

You'll need the following hardware and software to complete the practice exercises in this book:

- A 32-bit or 64-bit version of SQL Server 2008 Enterprise edition or SQL Server 2008 Developer edition. You need to install the Database Engine, Analysis Services, and Reporting Services components of SQL Server 2008. During installation add your Windows login ID to the list of SQL Server and Analysis Services system administrators.

Tip You can download a fully functional 180-day trial version of SQL Server 2008 Enterprise Edition from *http://www.microsoft.com/sqlserver/2008/en/us/trial-software.aspx*.

- SQL Server 2008 Enterprise Edition requires Microsoft Windows Server 2008 or Windows Server 2003 SP2. SQL Server 2008 Developer Edition requires Microsoft Windows Server 2008, Windows Server 2003 SP2, Windows Vista, or Windows XP SP2.
- 1.0 GHz Pentium III+ processor (2.0 Ghz or faster recommended) for 32-bit SQL Server 2008.
- 1.4 Ghz (2.0 Ghz or faster recommended) AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support processor, or 1.0 Ghz or faster Itanium processor for 64-bit SQL Server 2008.
- 512 MB (2 GB or more recommended) of available physical RAM.
- Video with VGA or higher resolution with at least 1,024 x 768 pixel resolution.
- CD-ROM or DVD-ROM drive.
- Microsoft mouse or compatible pointing device.

Detailed requirements for installing SQL Server 2008 can be found in the SQL Server Books Online article "Hardware and Software Requirements for Installing SQL Server 2008" (*http://msdn.microsoft.com/en-us/library/ms143506.aspx*).

The step-by-step exercises in this book and the accompanying practice files were tested using Windows Vista Enterprise Service Pack 1 and Microsoft SQL Server 2008 Developer Edition. If you're using another version of the operating system or SQL Server 2008, you might notice some slight differences.

Practice Files and SQL Server 2008 Configuration

The companion CD inside this book contains the practice files that you'll use as you perform the exercises. The practice files and the step-by-step instructions in the lessons let you learn by doing, which is an easy and effective way to acquire and remember new skills. The companion CD also contains a SQL Server 2008 database that is the data source for the analytical solutions that you will create and use throughout this book.

Installing the Practice Files

Follow these steps to install the practice files on your computer so that you can use them with the exercises.

1. Remove the companion CD from the package inside this book and insert it into your CD-ROM drive.

Note An end user license agreement should open automatically. If this agreement does not appear, open Computer on the desktop or Start menu, double-click the icon for your CD-ROM drive, and then double-click StartCD.exe.

2. Review the end user license agreement. If you accept the terms, select the accept option and then click Next.

   A menu will appear with options related to the book.

3. Click Install Practice files.

4. Follow the instructions that appear.

   The practice files will be copied from the CD to your hard drive. The default installation folder is C:\Microsoft Press\Analysis Services 2008 SBS. If you install the practice files in a different location, you'll need to reference the new location when working through the exercises. Each chapter in this book explains when and how to use any practice files for that chapter. When it's time to use a practice file, the book will list the instructions for how to open the file.

Tip In the C:\Microsoft Press\Analysis Services 2008 SBS\Answers folder, you'll find a separate folder for each chapter in which you make changes to the practice files. The files in these folders are copies of these sample files when you complete a chapter. You can refer to these files if you want to preview the results of completing all exercises in a chapter.

5. Remove the CD from your CD-ROM drive.

Configuring SQL Server 2008

The exercises in this book will have you create an Analysis Services database and populate that database using data contained in a SQL Server database. The exercises assume that SQL Server 2008 is installed on your computer. The following steps will guide you through the process of configuring SQL Server and Analysis Services so that you can complete the exercises. To complete these steps you must first install the practice files; you will also need to be a member of the Administrators group on your computer.

Analysis Services will need to connect to SQL Server, so you must first discover the Windows login user name for Analysis Services and then grant SQL Server permissions to that user name.

1. On the Microsoft Windows task bar, click Start. In the Search box, type **services** and then in the Programs group select Services. If a User Account Control dialog box appears, click Continue. (If you are using Windows XP or Windows Server 2003, click Start, select Control Panel, select Administrative Tools, and then select Services.)
2. In the Services window, find SQL Server Analysis Services in the Name column and make a note of the value in the Log On As column.

Important The Analysis Services login user name will be a two-part name with the format *DomainName**Username* or .*Username* or it may be *Network Service* or *Local Service*. You will use this user name in step 5.

3. Close the Services window.
4. On the Microsoft Windows task bar, click the Start button and then click Computer. In Windows Explorer, browse to C:\Microsoft Press\Analysis Services 2008 SBS\Setup. Right-click SSAS_Login.sql and select Open. In the Connect To Database Engine dialog box, click Cancel.

   Microsoft SQL Server Management Studio opens and displays the SSAS_Login.sql file in edit mode. This SQL script file creates a SQL Server login for Analysis Services (unless it already exists) and then grants permissions to that login. You will need to edit the script and enter the Analysis Services login user name that you discovered in step 2.

5. In the second line of the file, find DOMAIN\ACCOUNT and replace it with the two-part name you discovered in step 2.

Important If in step 2 you discovered that the Analysis Services login user name is *Network Service* or *Local Service*, then you will need to replace DOMAIN\ACCOUNT with NT Authority\Network Service or NT Authority\Local Service.

If the user name you discovered in step 2 has the format *DomainName**Username*, use that two-part name. If the user name you discovered has the format .*UserName*, you will need to replace the dot (.) with the name of the local computer. For example, on my computer the Analysis Services login is .\AS_Service. The name of my computer is Hitachi, so I will need to find DOMAIN\ACCOUNT and replace it with Hitachi\AS_Service. For me, the second line of the script should looks like this:

```
SET @user ='Hitachi\AS_Service';
```

6. On the File menu, select Exit. In the dialog box, click Yes to save the changes to SSAS_Login.sql.
7. In Windows Explorer, right-click Setup.cmd and select Open.

   This command file attaches the Microsoft SQL Server 2008 database that is the data source for the analytical solution you will create and use throughout this book. It then executes the script that you modified in step 5.

Now you're set to begin working through the exercises.

Uninstalling the Practice files

Follow these steps to remove the practice files from your computer.

1. In Control Panel, open Add Or Remove Programs.
2. From the list of Currently Installed Programs, select Microsoft SQL Server 2008 Analysis Services Step by Step.
3. Click Remove.
4. Follow the instructions that appear to remove the practice files.

Find Additional Content Online

As new or updated material becomes available that complements your book, it will be posted online on the Microsoft Press Online Developer Tools Web site. The type of material you might find includes updates to book content, articles, links to companion content, errata, sample chapters, and more. This Web site is available at *www.microsoft.com/learning/books/online/developer*, and is updated periodically.

Support for This Book

Every effort has been made to ensure the accuracy of this book and the contents of the companion CD. As corrections or changes are collected, they will be added to a Microsoft Knowledge Base article.

Microsoft Press provides support for books and companion CDs at the following Web site:

http://www.microsoft.com/learning/support/books/

Questions and Comments

If you have comments, questions, or ideas regarding the book or the companion CD, or questions that are not answered by visiting the sites above, please send them to Microsoft Press via e-mail to:

mspinput@microsoft.com

Or via postal mail to:

Microsoft Press
Attn: *Microsoft SQL Server 2008 Analysis Services Step by Step* Editor
One Microsoft Way
Redmond, WA 98052-6399

Please note that Microsoft software product support is not offered through the above addresses.

Chapter 3

Accessing Source Data

In this chapter, you will learn how to:

- Use Business Intelligence Development Studio to create a business intelligence solution and an Analysis Services project.
- Create a data source using the Data Source Wizard.
- Develop a data source view using the Data Source View Wizard.
- Create data source view diagrams.
- Add logical primary keys and entity relationships to a data source view.
- Create named calculations and named queries.

In this chapter, you will take the first steps toward creating an Analysis Services database, beginning by creating an Analysis Services project within a business intelligence solution. An Analysis Services project contains all of the code needed for one Analysis Services database. You will then add a data source to your project. A data source contains the information Analysis Services needs to connect to a source database. Finally, you will add a data source view that contains information about the tables and views in the source database. The data source view also allows you to provide supplemental metadata.

Creating a Business Intelligence Solution

Throughout this book, you will be using SQL Server Business Intelligence Development Studio (BIDS) to design and deploy a sample Analysis Services business intelligence solution. BIDS serves as the development environment for all of the SQL Server business intelligence tools, including Integration Services and Reporting Services, in addition to Analysis Services. If you are familiar with the Microsoft Visual Studio application development environment, you will notice that BIDS is Visual Studio with additional project templates specifically designed for SQL Server business intelligence. BIDS contains four main windows that you will use to develop BI applications: Designer Window, Solution Explorer, Properties Window, and Toolbox. With the exception of the Toolbox window, which is not used for developing Analysis Services projects, you will work with each of these BIDS components extensively as you work through the procedures in this book.

To get started, you will create a business intelligence solution that contains a single Analysis Services project. Then in the following sections of this chapter, you will add a data source and a data source view to the project and add additional functionality in each of the remain-

ing chapters of this book. The Analysis Services database that you will create is the analytical layer of a full-featured business intelligence solution. Most business intelligence solutions will also have a data transformation layer that extracts data from business systems, transforms the data, and then loads it into a data mart or data warehouse. Business intelligence solutions also have a presentation layer that provides reporting and data visualization. The code for the data layer and presentation layer can be contained in additional projects within a business intelligence solution. These other layers of a business intelligence solution are beyond the scope of this book, but they can be developed using the functionality of SQL Server 2008 Database Engine, Integration Services, Reporting Services, and Office 2007.

Create a new Analysis Services project

1. On the Microsoft Windows task bar, click Start, point to All Programs, expand the Microsoft SQL Server 2008 folder, and then select SQL Server Business Intelligence Development Studio.
2. On the File menu, point to New and then select Project.

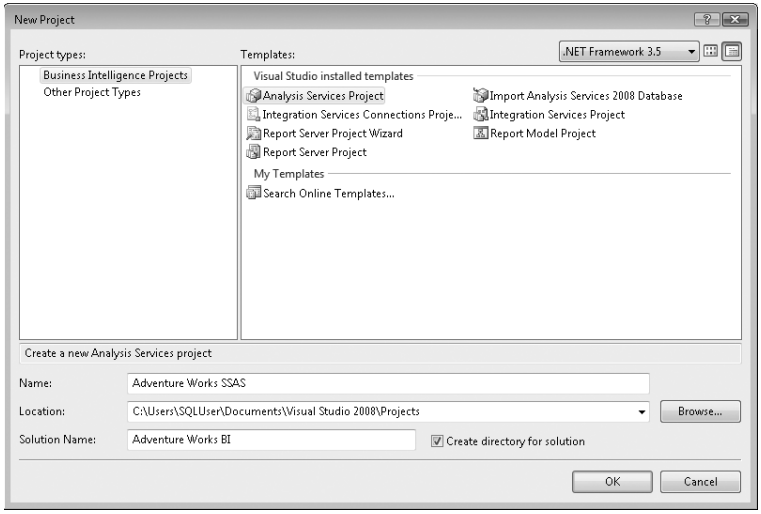
Note In BIDS, you create a solution that may contain multiple projects. However, the File menu doesn't have an item that allows you to create a solution. Instead, you choose to create a new project and BIDS will automatically create a solution for you.

3. Select the Business Intelligence Projects project type and select the Analysis Services Project template. The New Project dialog box also contains templates for Integration Services and Reporting Services projects.
4. Name your project **Adventure Works SSAS**. The text in the Solution Name box changes automatically to match the project name. When you create a solution that might have multiple projects, you can rename the solution so that the name is more representative of the entire solution.
5. Select Create Directory for Solution and change the solution name to **Adventure Works BI**.
6. Confirm that the solution location is C:\Users\<*YourUserName*>\Documents\Visual Studio 2008\Projects.

   This folder is the default location for any solution you create using Visual Studio 2008. If you have previously created a Visual Studio project, the solution location will display the location of the last project you created.

Note If you are using Windows XP or Windows Server 2003, the default location for Visual Studio 2008 solutions is C:\Documents and Settings\<*YourUserName*>\My Documents\Visual Studio 2008\Projects.

The New Project dialog box should look similar to the following:



7. Click OK to create the project. BIDS will create the Adventure Works SSAS project and display it in Solution Explorer.

   You may not see the Adventure Works BI solution. With the default BIDS settings, if a solution contains only one project, BIDS will not display the solution. The solution will be displayed when you add another project. If you would like to display the solution even when there is only one project, select Options from the Tools menu and then select Always Show Solution on the Projects And Solutions tab of the Options dialog box.

Creating a Data Source

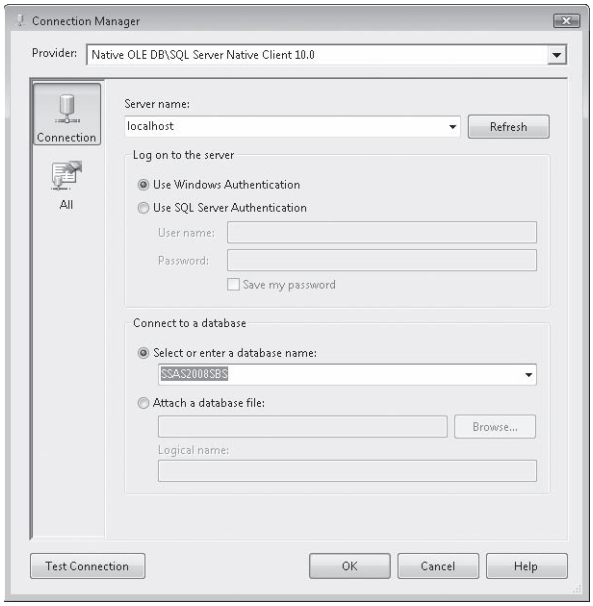
The first task in creating an Analysis Services project is to create a data source. The data source contains the information that Analysis Services uses to connect to the source database. It contains the data provider name, the server and database name, and the authentication credentials that Analysis Services will use.

The source data must be contained in a relational database. Analysis Services can read data contained in SQL Server 2008, Microsoft Access, Teradata, Oracle, IBM DB2, and other relational databases for which you have an OLE DB or ODBC driver, but it cannot read data from a Microsoft Office Excel file, a text file, or other nonrelational data sources. Analysis Services connects to a relational database using a .NET or OLE DB data provider. When you create a data source, BIDS will default to the SQL Server Native Client data provider. BIDS also includes data providers for other databases such as Microsoft Access and Oracle. If BIDS doesn't include a data provider for the database that contains your source data, you will need to install that database's client connectivity components.

Note The approved list of data providers can be found in the SQL Server 2008 Books Online article "Defining Data Sources," located at *http://msdn.microsoft.com/en-us/library/ms175608.aspx*.

Create a data source

1. In BIDS in Solution Explorer, right-click the Data Sources folder and select New Data Source. The Data Source Wizard appears.
2. On the Welcome page, click Next.
3. On the Select How To Define The Connection page, click the New button. The Connection Manager dialog box appears.
4. Type a server name: **localhost**.
5. In the Select Or Enter A Database Name list box, select SSAS2008SBS. The Connection Manager dialog box should now look like this:



6. Click Test Connection. A dialog box opens with the message "Test connection succeeded." Click OK.

   Now is a good time to ensure that BIDS can connect to the source database—you will not be able to successfully create a data source view unless BIDS has database connectivity. If you receive an error message, verify that you have permission to read from the source database and that you have properly specified the data source provider, server name, and database name.

7. Click OK to close the Connection Manager dialog box.
8. In the Data Source Wizard, on the Select How To Define The Connection page, click Next.

   The Impersonation Information page appears. On this page, you configure the Windows security credentials that Analysis Services will use when it connects to the data source you defined in steps 4 through 7.

   - The Use A Specific Windows User Name And Password option lets you enter the user name and password of a Windows user account.
   - The Use The Service Account option will have Analysis Services use its service logon user ID to connect to the data source.
   - The Use The Credentials Of The Current User option is only used for some very specialized circumstances. It is important to note that when you use this option, Analysis Services will not use the Windows user name and password of the current user for most processing and query tasks.
   - The Inherit option causes this data source to use the impersonation information contained in the Analysis Services DataSourceImpersonationInfo database property.

   The most commonly selected options are Use The Service Account or Use A Specific Windows User Name and Password.
9. On the Impersonation Information page, select Use The Service Account and click Next.
10. Leave SSAS2008SBS as the data source name and click Finish to complete the wizard.

Tip The name of a data source defaults to the name of the database that it references. In general, this name works well and helps you remember which database each data source points to. However, if the name of the source database in your development environment is different than the name of the source database in your test or production environments, you want to give the data source a more generic name that is meaningful in all three environments.

Notice that the Data Sources folder now contains the file SSAS2008SBS.ds. This XML file contains the data source information that you just created using the Data Source Wizard.

Now that you have created a data source, you are ready to create a data source view.

Creating a Data Source View

A data source view is a logical data model that exists between your physical source database and Analysis Services dimensions and cubes. When you create a data source view, you designate the tables and views in the source database that contain data that you want to use to build dimensions and cubes. BIDS connects to the source database and reads the table and view definitions and stores that metadata in the data source view. Metadata is "data about the data": the names of the tables and views, the column names, data types, primary key columns, and foreign key relationships.

The data source view allows you to provide additional metadata. If you need to transform data, you can add a named calculation that contains a SQL expression to a table. If you need to filter, group, or join data from multiple tables, you can create a logical table called a *named query* that contains a SQL select statement. Although it is common for source data to be stored in a single data warehouse or data mart, sometimes the business data you need to access is stored in multiple databases. You can create a data source for each database and then add tables from the databases into a single data source view. You can then add logical foreign key relationships to the data source view to show how the data from different databases should be joined together.

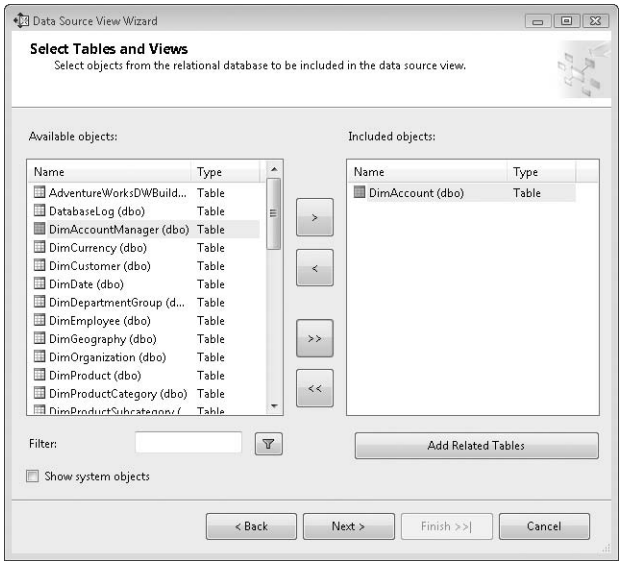
When you create dimensions and cubes, you will start with the basic metadata stored in the data source view and enhance it with information about hierarchical relationships, sort order, proper formatting, and so forth, so that your Analysis Services database becomes a metadata-rich data analysis environment.

Create a data source view

1. In Solution Explorer, right-click the Data Source Views folder and select New Data Source View. The Data Source View Wizard appears.
2. On the Welcome page, click Next.
3. On the Select a Data Source page, select the SSAS2008SBS relational data source and click Next.

   The Select Tables And Views page appears. The Available Objects list shows all of the tables and views contained in the source database. The Included Objects list displays the tables and views that will appear in your data source view.
4. Double-click DimAccount to add it to the Included Objects list. Alternatively, you can select a table or multi-select several tables and then click the Add button (>) to add the table or tables to the Included Objects list.

   The Select Tables and Views page should look like the following:

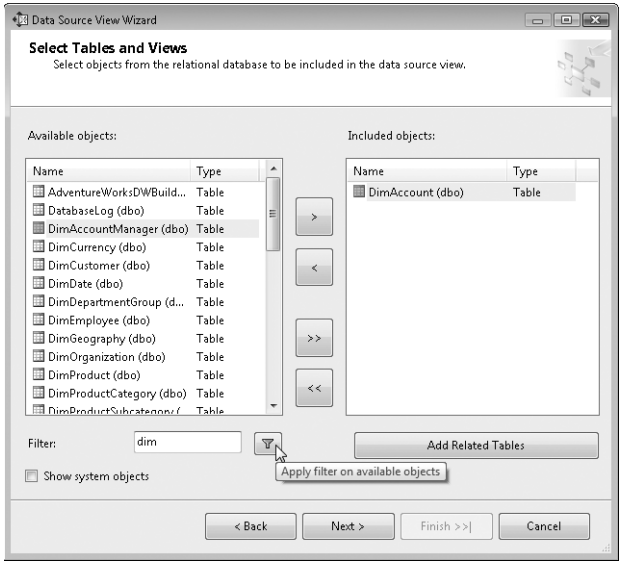


5. Select the DimAccount table and then click Add Related Tables.

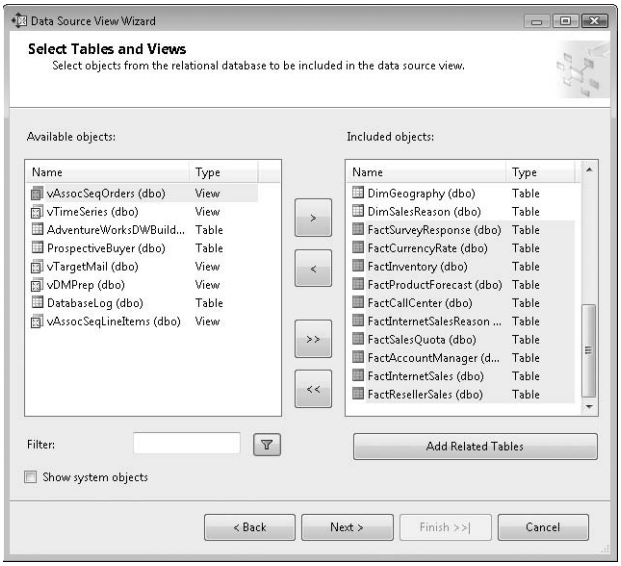
   The Data Source Wizard analyzes the foreign key relationships in the source database and adds the FactFinance table to the Included Objects list. The FactFinance table is the only table related to DimAccount.

6. In the Filter box, type **dim** and click the Filter button.

   You can filter the Available Objects list in order to make it easier to find the tables and views that you want to include in your data source view. In the SSAS2008SBS database, all dimension table names begin with "Dim", so the Available Objects list now shows all of the dimension tables in SSAS2008SBS.



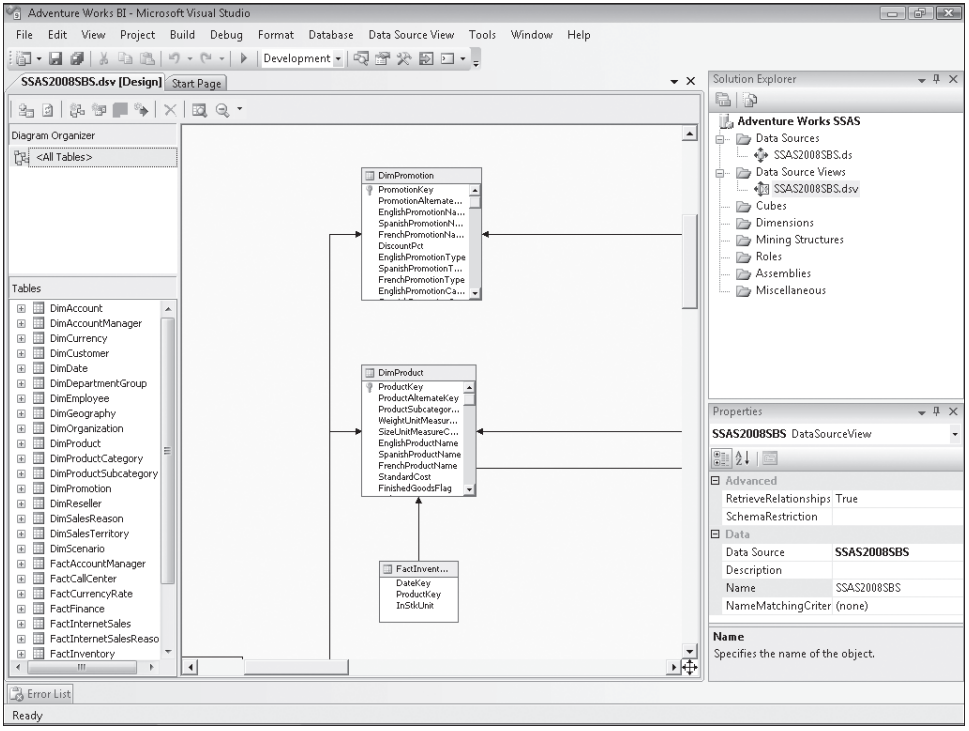
7. Click the Add All button (>>) to add the dimension tables to the Included Objects list.
8. In the Filter box, delete dim, type **fact**, and click the Filter button. Click the Add All button (>>) to add the fact tables to the Included Objects list.
9. Clear the Filter box and click the Filter button. The tables and views that remain in the Available Objects list are in the SSAS2008SBS database, but they will not be included in your data source view. The Select Tables and Views page should look similar to this:



10. Click Next. On the Completing The Wizard page, accept the default name SSAS2008SBS and click Finish. The Data Source View designer appears, displaying the tables you selected.

    Notice that the Data Source Views folder in Solution Explorer contains the file SSAS2008SBS.dsv. This XML file contains the data source view information that you just created using the Data Source View Wizard.

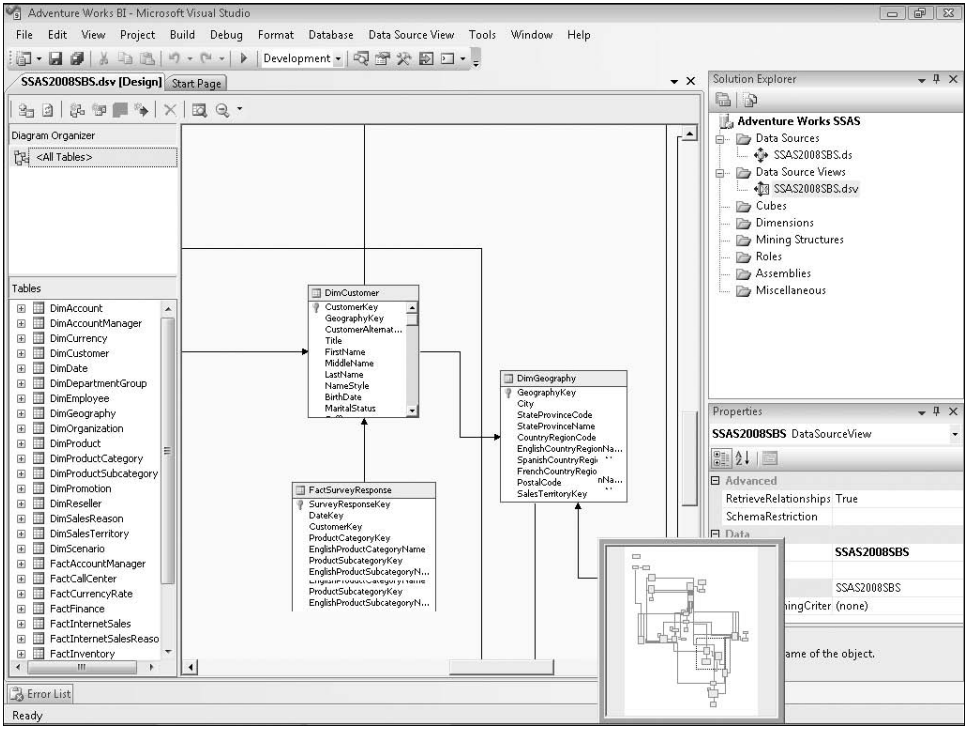
Tip If the source database changes, you want your data source view to change as well. Select Refresh from the Data Source View menu, and BIDS compares the source database tables and the data source view metadata, shows you the differences, and allows you to accept or ignore changes.



Browse a data source view

If your data source view contains many tables, you won't be able to see all of the tables in the diagram pane. The following procedure shows you how to browse the contents of a data source view.

1. Click and hold the four-headed arrow button at the lower-right corner of the data source view diagram pane. The table locator looks like the following image.



2. Drag your mouse around the table locator. The diagram pane scrolls to match your movements.

3. Click a table name in the Tables pane. The diagram pane scrolls so that it is centered on the table you selected.

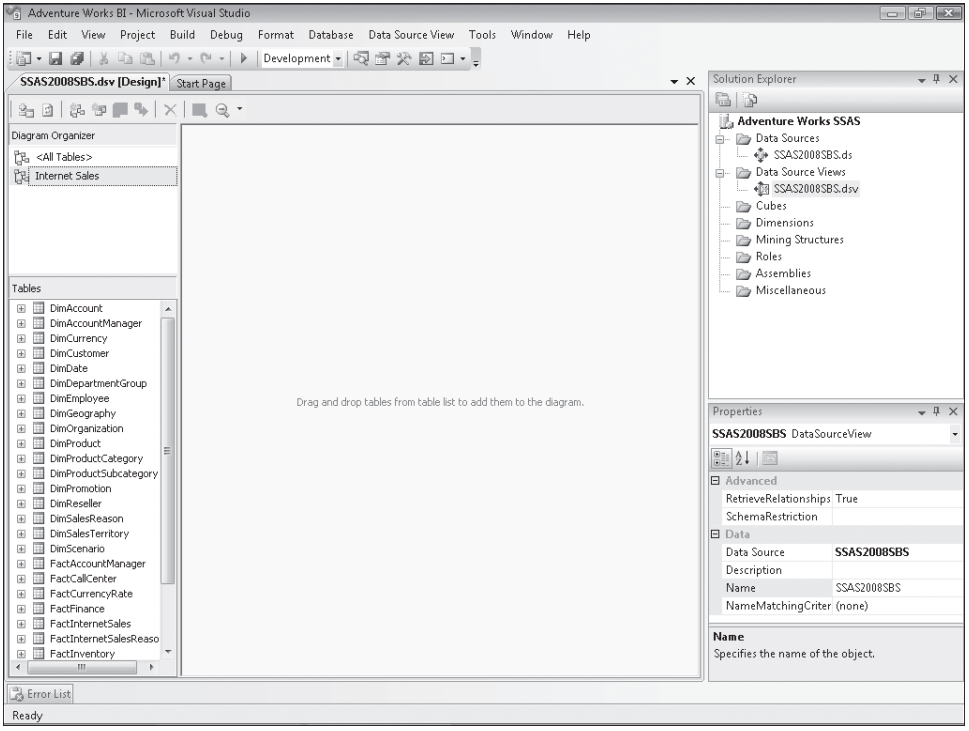
Modifying a Data Source View

The data source view you have created contains metadata that was read from the source database. You can now enhance the data source view by adding additional metadata. You can add logical primary keys and relationships. You can also add a SQL expression to a table or add a logical table based on a SQL query. A data source view diagram may contain a large number of tables and relationships. This makes it hard to understand the structure of the source data. To get a more understandable view of the data, you can create additional data source view diagrams that display a subset of related tables.

Create a subject area diagram

You can create a diagram that contains a subset of the tables in your data source view. I like to create a diagram for each fact table and all of its related dimension tables.

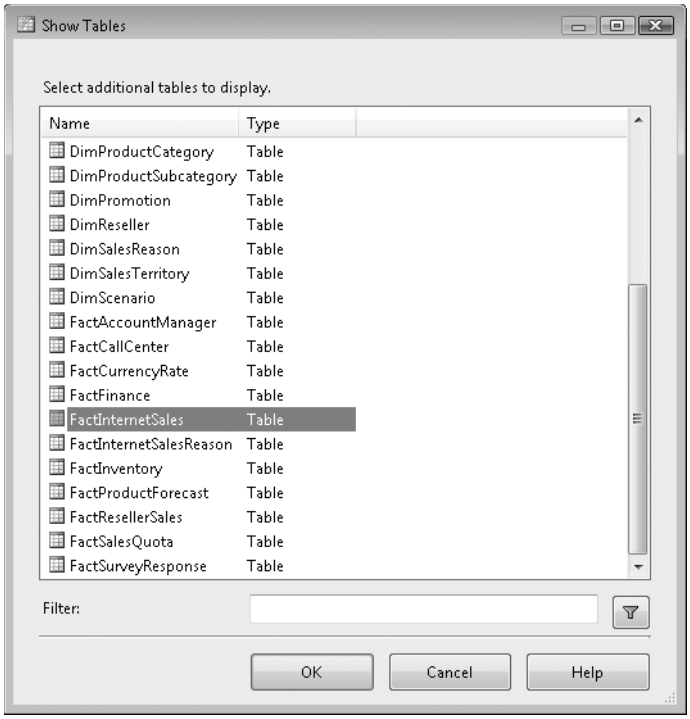
1. On the Data Source View menu, select New Diagram. A new diagram appears in the Diagram Organizer pane. The diagram pane is empty.
2. In the Diagram Organizer, change the name of the diagram to **Internet Sales**. The Diagram Organizer and diagram pane should look like this:



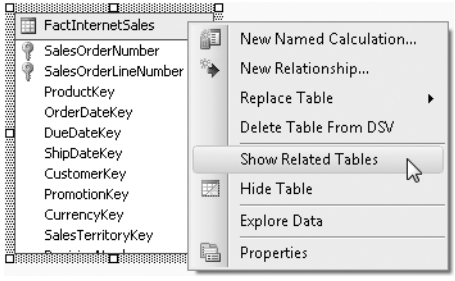
3. Right-click in the diagram pane and select Show Tables.

Important In a diagram, you can choose to show or hide tables without affecting other diagrams in the data source view. Any other changes you make to one diagram are applied to all diagrams in the data source view.

4. In the Show Tables dialog box, select FactInternetSales. The Show Tables dialog box should look like the following image.



5. Click OK. Right-click the header of the FactInternetSales table and select Show Related Tables.



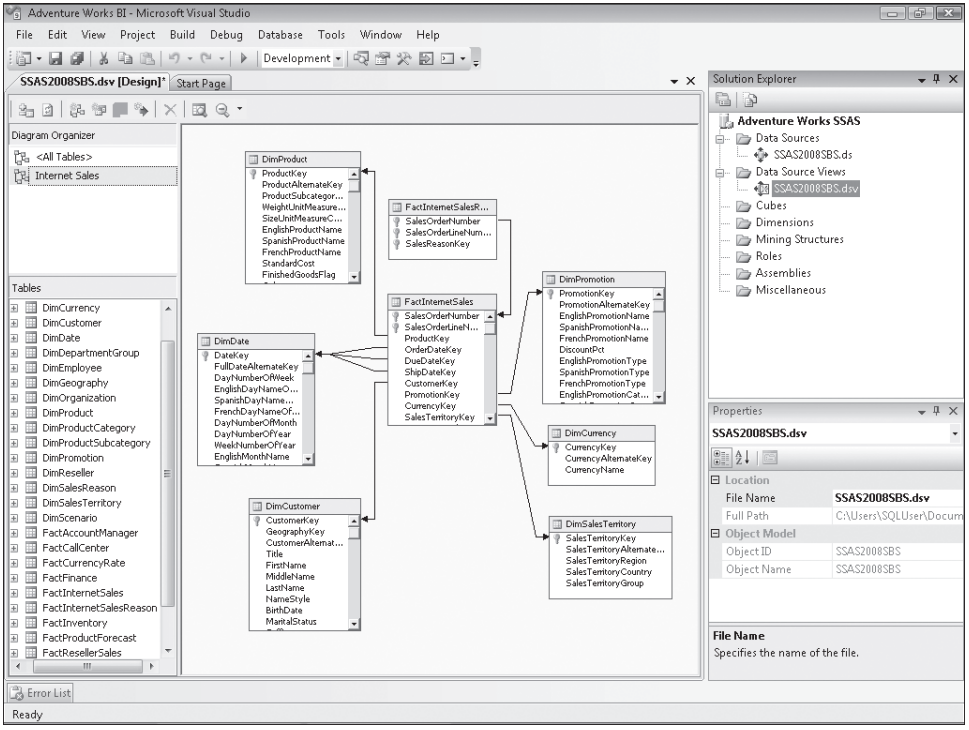
All of the tables related to FactInternetSales appear in the diagram.

6. Right-click a blank area of the diagram pane and select Switch To Diagonal Layout.

You can arrange tables in the data source view using rectangular layout or diagonal layout. With rectangular layout, the relationship lines are drawn between the tables. I prefer diagonal layout where the relationship lines are drawn between the columns that are related.

Tip When you switch between rectangular layout and diagonal layout, BIDS will rearrange the tables in your diagram. You should select rectangular or diagonal layout before you manually arrange the tables in your diagram.

7. Right-click a blank area of the diagram pane, point to Zoom, and select 75%. You may wish to further organize your diagram by selecting each table and moving or resizing it. With a little practice, you can get your data source view diagram pane to look like this:



Add primary keys and relationships

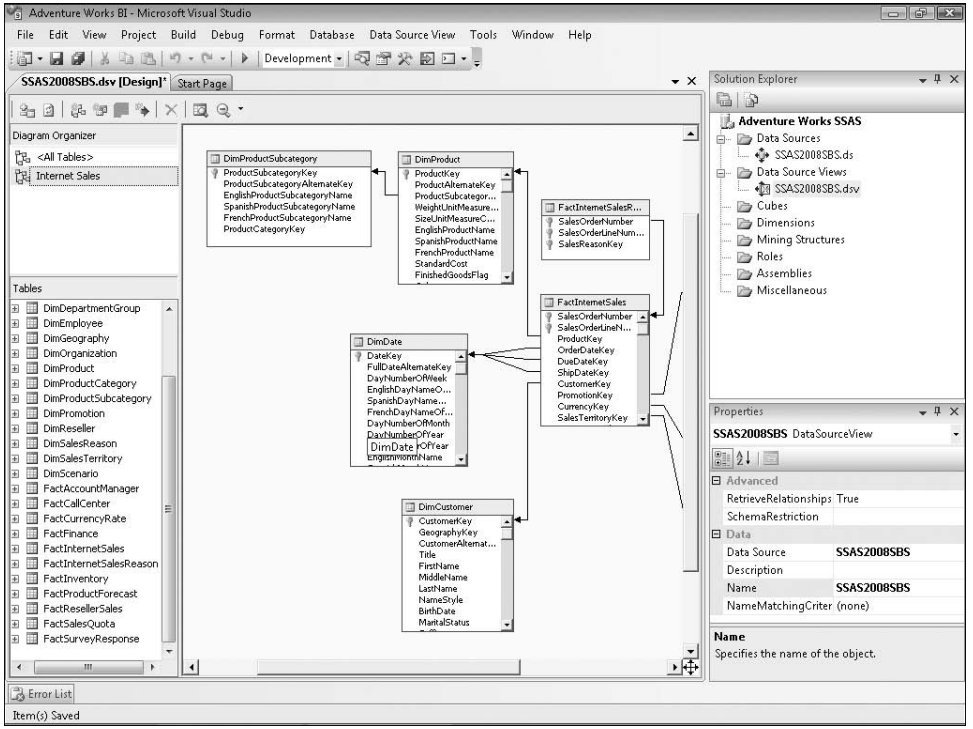
The SSAS2008SBS database contains tables with product category and subcategory information that you will want to include in your data source view. The metadata for these two tables is incomplete. No primary keys or relationships have been defined. In this procedure, you will show these tables in the Internet Sales diagram and add logical primary keys and relationships. A database primary key designates the column(s) that uniquely identify the rows in a table and does not allow any rows to have duplicate values in the primary key columns. A logical primary key in the data source view also designates the columns that uniquely identify the rows in a table, but it doesn't enforce uniqueness. You must have some other mechanism to guarantee that there is a unique value for each row in the table or Analysis Services will generate errors when you try to process a dimension.

1. In the Tables list, select DimProductSubcategory, drag it onto the diagram pane, and drop it near DimProduct.
2. In the DimProductSubcategory table, right-click the ProductSubcategoryKey column and select Set Logical Primary Key. BIDS creates a logical primary key on the DimProductSubcategory table and displays a key icon next to ProductSubcategoryKey.

Note A table can have a primary key with more than one column. Just select the primary key columns and then right-click and select Set Logical Primary Key.

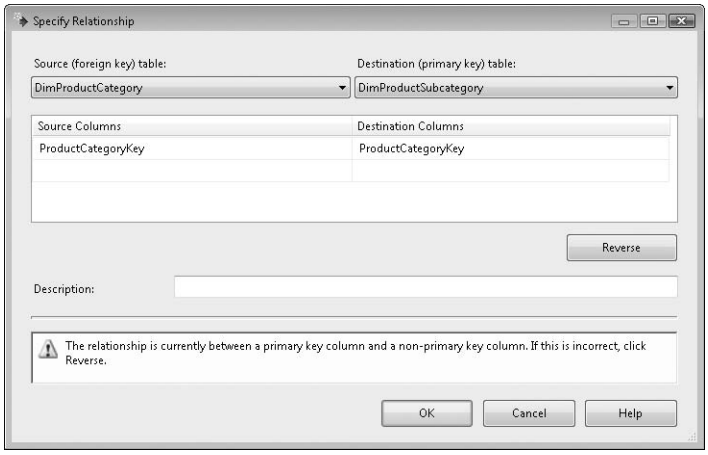
3. Click ProductSubcategoryKey in the DimProduct table and drop it on ProductSubcategoryKey in the DimProductSubcategory table.

   BIDS creates a relationship between the DimProduct and DimProductSubcategory tables and displays a relationship line showing that the tables are joined on ProductSubcategoryKey. Your diagram pane should look similar to this:

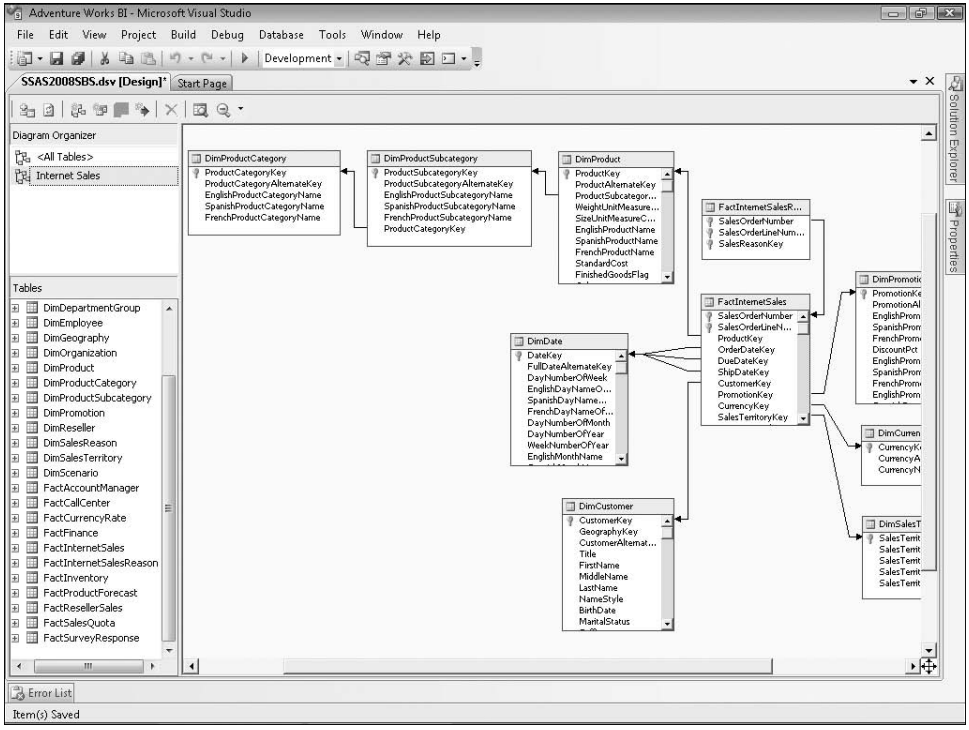


4. In the Tables list, select DimProductCategory, drag it onto the diagram pane, and drop it near DimProductSubcategory.
5. In the DimProductCategory table, right-click the ProductCategoryKey column and select Set Logical Primary Key.

6. Click ProductCategoryKey in the DimProductCategory table and drop it on ProductCategoryKey in the DimProductSubcategory table. The Specify Relationship dialog box appears, warning you that you have reversed the typical foreign key–primary key relationship.



7. Click Reverse to correct the logical foreign key relationship and then click OK. You may want to rearrange the tables in your diagram so that the data source view now looks like the following image.



Add a named calculation

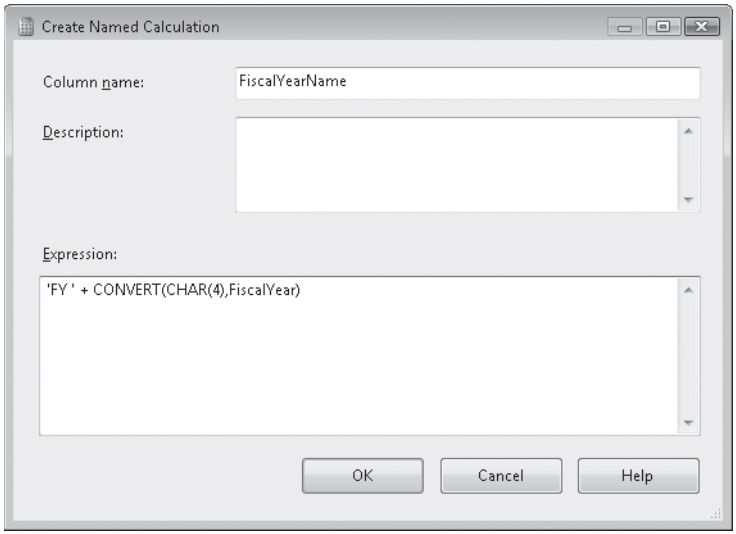
You may need to apply transformations to your source data. It is common to append columns, parse data from a column, or perform calculations. For example, you may want to append customers' first and last names or you may want to add sales amount, tax amount, and shipping to get the total amount. In the next procedure, you add the characters FY to the fiscal year so that users can easily distinguish fiscal and calendar dates.

1. Right-click the header of the DimDate table and select New Named Calculation. The Create Named Calculation dialog box appears.
2. In the Column Name box, type **FiscalYearName**.
3. In the Expression box, type this SQL expression:

```
'FY ' + CONVERT(CHAR(4),FiscalYear)
```

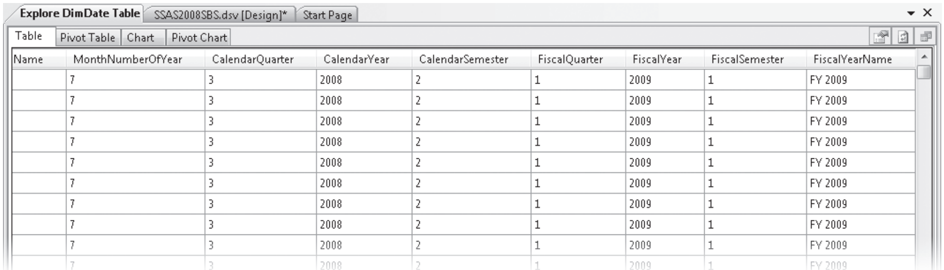
Important The SQL expression that you enter is passed through to the source database. It must use valid SQL syntax for the relational database management system (RDBMS) that is hosting your data. I often use SQL Server Management Studio to write a SQL query that contains the SQL expression, execute the query to make sure the expression is correct, and then copy the expression and paste it into the named calculation.

The Create Named Calculation text box now looks like this:



4. Click OK. The new named calculation FiscalYearName now appears as the last column in the DimDate table.
5. Right-click the header of the DimDate table and select Explore Data. Scroll to the far right to see the values in the FiscalYearName column.

The Explore Data screen should look similar to this:



6. Close the Explore DimDate Table window.

Create a named query

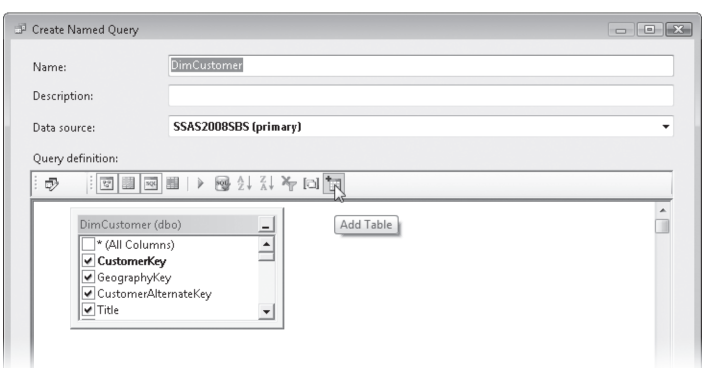
Sometimes you may need to apply transformations to your data that are more complicated than just applying a SQL expression. You may want to filter, group, or join data from multiple tables. You can do this by creating a SQL SELECT statement and then putting the statement into a named query.

In the next procedure, you will replace the DimCustomer table with a named query. The SQL SELECT statement will join DimCustomer and DimGeography so that you can identify a customer's city, state, and country.

1. Right-click the header of the DimCustomer table, point to Replace Table, and select With New Named Query.

   The Create Named Query dialog box appears. The diagram pane shows the DimCustomer table with each column selected, the grid pane shows each column from DimCustomer, and the SQL pane shows the corresponding SQL select statement. These three panes are alternative displays of the same information.

2. Click the Add Table button.

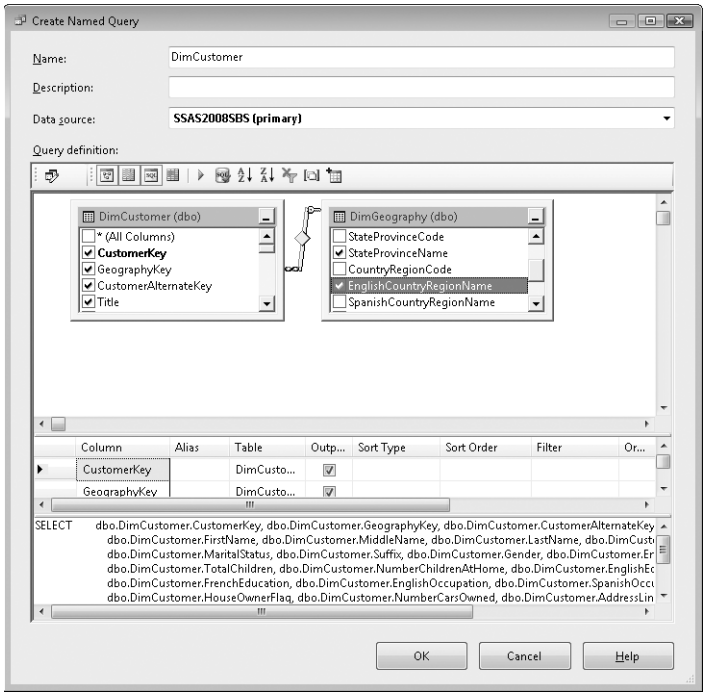


3. In the Add Table dialog box, select DimGeography, click Add, and then click Close.

4. In the DimGeography table, select City, StateProvinceName, and EnglishCountryRegionName. The Create Named Query dialog box now looks like the following image.

Important The SQL query that you enter is passed through to the source database. It must use valid SQL syntax for the RDBMS that is hosting your data. I often use SQL Server Management Studio to write and execute a SQL query and then copy the query and paste it into the Create Named Query SQL pane.

BIDS attempts to validate your SQL query when you close the Create Named Calculation dialog box. Sometimes BIDS is unable to parse valid SQL and raises an error. If you know that your SQL query is valid, you can click the Switch To Generic Query Builder button. Then BIDS will not attempt to validate your SQL query.

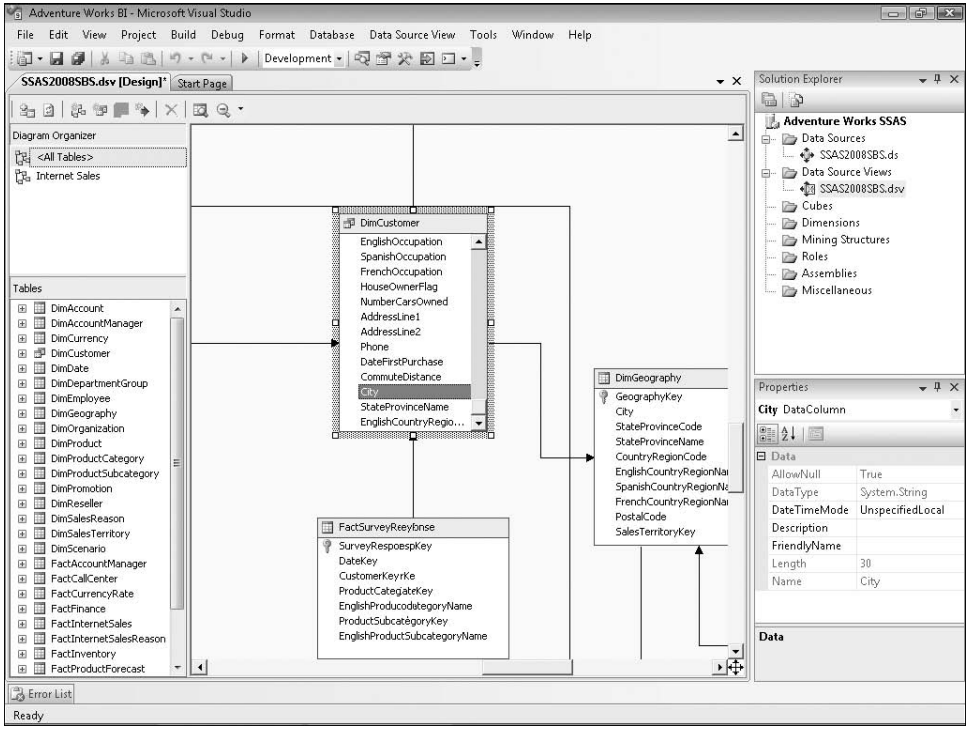


5. Click OK. The icon next to DimCustomer in the table header has changed from a single table to multiple tables, indicating that DimCustomer is now a named query.

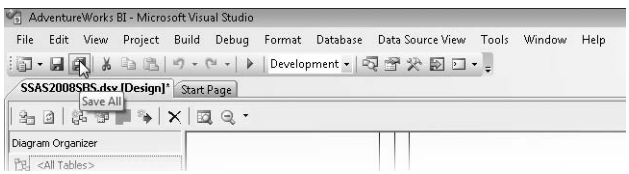
6. In the Diagram Organizer pane, select the <All Tables> diagram and then in the Tables pane, select DimCustomer.

   The changes you made in the Internet Sales diagram have also been made in the <All Tables> diagram. DimCustomer is a named query everywhere it appears in your data source view.

The fields you selected in your Named Query can be found when you scroll within the DimCustomer table:



7. Click the Save All button and close BIDS.



In this chapter, you learned how to create a business intelligence solution and an Analysis Services project using SQL Server Business Intelligence Development Studio. You created a data source that contains the information Analysis Services uses to connect to a database. You then created a data source view that contains the definitions of the tables and views in a source database. Because a data source view that contains many tables can be hard to comprehend, you created a diagram that displayed a subset of the tables in the data source view. The source database was missing some primary keys and foreign key relationships, so you added logical primary keys and relationships to the data source view. Finally, in order to transform the source data, you added a named calculation containing a SQL expression and a named query containing a SQL query to the data source view.

In Chapter 4, "Creating Dimensions," and in Chapter 5, "Creating a Cube," you will create dimensions and cubes based on the metadata contained in the data source view. When you process these dimensions and cubes, Analysis Services will use the information in the data source to connect to the source database.

Chapter 14

Managing Partitions and Database Processing

In this chapter, you will learn how to:

- Configure a dimension to use ROLAP or MOLAP storage.
- Configure a measure group partition to use ROLAP, HOLAP, or MOLAP storage.
- Configure rigid attribute relationships.
- Process a dimension using the Process Update processing option.
- Process a partition using the Process Incremental processing option.
- Configure proactive caching.
- Create, merge, and manage measure group partitions.
- Import an Analysis Services design into a Business Intelligence Development Studio project.

In Chapter 11, "Retrieving Data from Analysis Services," you learned how to use client tools and MDX queries to get data out of Analysis Services and into a PivotTable dynamic view or other report. In order to get data *out of* the Analysis Services database, you had to first put data *into* the Analysis Services database. In previous chapters, every time you were ready to apply your changes, you simply deployed your solution and let the Analysis Services server work its magic to determine how to go about loading the data from the relational data warehouse into the Analysis Services database. How Analysis Services actually processed and stored that data has been a conceptual black box up to this point. Knowledge is power, however, and that adage certainly holds true in the domain of Microsoft SQL Server 2008 Analysis Services. The more you know about what goes on within the black box of the Analysis Services server, the better you'll be able to troubleshoot unusual situations and find solutions to difficult problems.

This chapter explains in broad terms what goes on inside that black box. You will learn about how Analysis Services physically processes and stores data. The explanations are simplified, but they encompass some of the most complex topics in this book. If nothing else, this chapter can give you an appreciation for the sophistication and elegance of the Analysis Services server design.

Working with Storage

An Analysis Services database uses four types of information: the design of database objects (dimensions, cubes, measure groups, and so on), dimension data, detail values, and aggregated values.

Analysis Services always stores the design of database objects in XML files on the Analysis Services server, but it allows you to decide where the dimension data, details values, and aggregated values are stored. You can choose from three storage modes that correspond to different physical storage locations:

- ROLAP (relational OLAP) leaves dimension data and detail values in the relational database. If the relational database is SQL Server Database Engine, Analysis Services can also create indexed views in the relational database to store aggregated values.
- HOLAP (hybrid OLAP) leaves the detail values in the relational fact table but stores aggregated values on the Analysis Services server. HOLAP is not an option for dimension storage.
- MOLAP (multidimensional OLAP) stores dimension data, detail values, and aggregated values on the Analysis Services server.

No matter which storage mode you choose, the database design is stored on the Analysis Services server. It's the database design that makes Analysis Services data appear to be in a cube to a person running a query. That means the storage mode is invisible to client applications—that is, applications that query the cube. The client application always sees the cube. Deciding which storage option to use is primarily based on processing and query performance and has no effect on how client applications will interact with the cube.

Because a client application can't tell which storage mode you have chosen, you can change the storage mode without affecting any client applications. After you specify storage and start using the cube, you can still change your mind later and switch to a different storage type. Because a cube appears to the client application as a single, logical entity, you can use different storage modes for different portions of a cube. To avoid doing that, you must use multiple partitions. You will learn about creating partitions later in this chapter.

Note Regardless of which storage mode you choose, Analysis Services will never allocate storage for missing values. For example, if you have a database that shows you didn't start selling products in Australia until 2010, Analysis Services will use no storage space for detail or aggregated values for Australia in 2008.

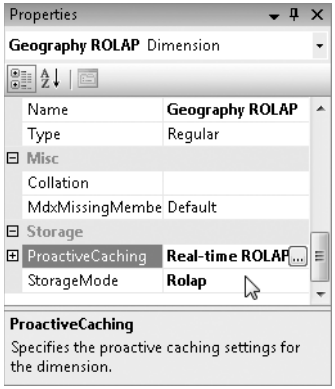
Understanding Dimension Storage Modes

You can choose to store dimension data using ROLAP or MOLAP storage mode. If you choose ROLAP storage, the dimension data will be stored in the source dimension tables. If you choose MOLAP storage, the dimension data will be stored on the Analysis Services server. You can use dimensions with ROLAP or MOLAP storage with partitions that have ROLAP, HOLAP, or MOLAP storage with one restriction: If the partition uses MOLAP storage, it must have one dimension that uses MOLAP storage. If you use a dimension with ROLAP storage with a partition that uses MOLAP storage, any time you execute a query that uses a leaf level member from an attribute in the ROLAP dimension, Analysis Services will have to retrieve the values from the relational database. For example, if Product was a ROLAP dimension and a user included the members of the Product Color attribute hierarchy in a query, Analysis Services would have to retrieve both the dimension data and the fact data directly from the relational database, and you would lose some of the query performance benefits that MOLAP storage provides. The only time you want to use the ROLAP storage mode for a dimension is when a dimension contains an extremely large number of members or when you want a measure group that uses only ROLAP storage for its dimensions and partitions.

In the next procedure you will learn how to change the storage mode to ROLAP for a dimension. Later in this chapter, you will work with cubes that contain partitions using ROLAP and MOLAP storage. You will then browse the cubes and see that to a client application, the cubes appear alike.

Modify dimension storage settings

1. Use Business Intelligence Development Studio (BIDS) to open the AdventureWorks BI solution contained in the C:\Microsoft Press\Analysis Services 2008 SBS\Chapter 14 \AdventureWorks BI folder. The AdventureWorks SSAS database contains two dimensions, Geography MOLAP and Geography ROLAP, which are identical. You will change the storage mode of the Geography ROLAP dimension to Real-time ROLAP. The other three dimensions, Date, Employee, and Product, use MOLAP storage.
2. In Solution Explorer, expand the Dimensions folder, right-click Geography ROLAP.dim, and select View Designer.
3. In the Attributes pane of the Dimension Designer, right-click the Geography ROLAP dimension and select Properties.
4. In the Properties window, select the *ProactiveCaching* property and click the ellipsis button (…) that appears on the right.
5. In the Dimension Storage Settings dialog box, drag the slider to Real-time ROLAP and click OK. The Properties window will now show that the value of the *ProactiveCaching* property is *Real-time ROLAP*, and the value of the *StorageMode* property is *Rolap*.



6. Close the Dimension Designer and save the changes to the Geography ROLAP dimension.

You were able to change the storage mode of the dimension without making any other change to its design. Query performance is the only difference that a user should perceive between a dimension that uses MOLAP storage and one that uses ROLAP storage.

Note In addition to query performance, there is one minor difference in the way Analysis Services will display data related to a ROLAP dimension attribute's unknown member in some special circumstances. If you are considering using ROLAP storage for a dimension that has the *UnknownMember* property enabled, refer to SQL Server 2008 Books Online for more information on this special case.

In the next section, you will learn how to change the storage mode of a measure group partition.

Understanding Partition Storage Modes

Choosing a storage mode is not as difficult as it might seem. Use ROLAP storage if your analytical solution requires real-time data or if you have insufficient disk storage or processing time to process a large volume of data into MOLAP storage. The ROLAP storage with aggregations option is rarely used because aggregations in a relational database can be bulky, and they usually only marginally improve query performance. You might choose the ROLAP storage with aggregations option if you're learning about aggregations and want to physically look at the aggregation tables, but this storage mode is rarely used in practice.

Aggregations in both MOLAP and HOLAP are identical—the only difference is where the detail-level values are stored. If you count the space required by the original warehouse as well as the space needed for the OLAP cubes, MOLAP does consume more storage space than HOLAP because the MOLAP storage option duplicates the values from the fact table. Analysis Services, however, is very efficient in how it stores data. An Analysis Services da-

tabase will often use less than half of the storage space of its source database. With a very large warehouse database, you could process the data into a MOLAP cube and then archive and remove the original warehouse. By using the MOLAP storage option, you could actually end up using a fraction of the original storage space.

If you have a large, permanent warehouse—and if using aggregations can satisfy most queries—you may want to consider HOLAP storage. Queries that must retrieve detail data are slower than if the cube used MOLAP storage, but if they're infrequent, the performance gain might not be worth the incremental storage requirements. In addition, processing a MOLAP cube can take more time than processing a HOLAP cube. While developing an OLAP cube, you may want to use HOLAP storage simply to speed up processing during the time that you process frequently. When you have completed the database design, you can switch to MOLAP storage to maximize query performance.

Note Some descriptions of warehouse technology use the term ROLAP to refer to a relational data warehouse that has a fact table and dimensional tables. This is a different meaning of the term than is used within Analysis Services and corresponds most closely to a ROLAP cube with no aggregations.

The AdventureWorks SSAS database contains two cubes, ROLAP and MOLAP, that both use MOLAP storage. The cubes are identical except that the ROLAP cube contains the Geography ROLAP dimension, whereas the MOLAP cube contains the Geography MOLAP dimension. In the next procedure, you will change the storage mode of one of the partitions in the ROLAP cube. You will then browse the cubes and see that they appear to be identical.

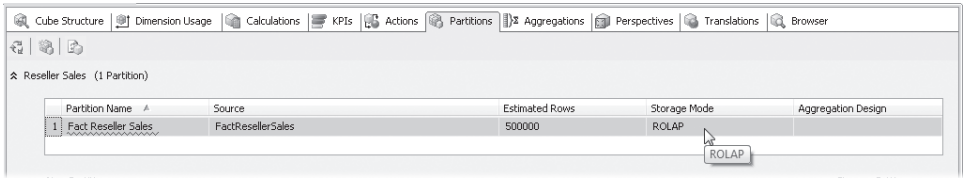
Modify partition storage settings

1. In Solution Explorer, expand the Cubes folder, right-click ROLAP.cube, and select View Designer.
2. In the Cube Designer, click the Partitions tab. The Reseller Sales measure group contains one partition, Fact Reseller Sales. Notice that the partition's storage mode is MOLAP. In the next three steps, you will change the storage mode to ROLAP.
3. Right-click the Fact Reseller Sales partition and click Storage Settings.

Note You can also change the storage mode by clicking the Storage Settings link. However, if you click this link without first selecting a partition, the changes you make won't affect any of the existing partitions. Instead, the storage settings that you specify will become the default settings for new partitions.

4. In the Partition Storage Settings dialog box, drag the slider to Real-time ROLAP and click OK.

5. Verify that the partition's Storage Mode column displays ROLAP, like this:



6. Close the Cube Designer and save the changes to the ROLAP cube. In the next procedure, you will browse these two cubes, so you need to deploy and process the AdventureWorks SSAS database.

7. On the Build menu, select Deploy AdventureWorks SSAS. If the AdventureWorks SSAS database already exists on the server, a dialog box may appear warning that the database will be overwritten. If the warning appears, click Yes. The prior version of the database will be deleted and the current deployment will continue.

You are now ready to browse the cubes and see that they appear identical.

In the next procedure, you will browse the cubes using SQL Server Management Studio (SSMS).

Browse the ROLAP and MOLAP cubes

1. On the Microsoft Windows task bar, click Start, point to All Programs, expand the Microsoft SQL Server 2008 folder, and then select SQL Server Management Studio.

2. In the Connect To Server dialog box, select Analysis Services from the Server Type list. In the Server Name text box, type **localhost** and click Connect.

3. In Object Explorer, expand the Databases folder, expand the AdventureWorks SSAS database, and expand the Cubes folder.

4. In Object Explorer, right-click the ROLAP cube and select Browse.

5. In the Metadata pane, expand the Measures folder, expand the Reseller Sales measure group, and drag the Reseller Order Quantity measure to the totals area of the Report pane.

6. In the Metadata pane, expand the Date dimension and drag the Calendar Year attribute to the columns area of the Report pane.

7. Expand the Geography ROLAP dimension and drag the Geography hierarchy to the rows area of the Report pane. Click the Report pane and then right-click the Area attribute and select Expand Items. The report should look like this:

Drop Filter Fields Here

		Calendar Year				
		CY 2008	CY 2009	CY 2010	CY 2011	Grand Total
Area	Country	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity
⊟ Europe	⊞ France		2,680	7,715	3,953	14,348
	⊞ Germany			4,480	2,900	7,380
	⊞ United Kingdom		2,443	7,060	3,690	13,193
	Total		5,123	19,255	10,543	34,921
⊟ North America	⊞ Canada	2,404	12,727	18,801	7,829	41,761
	⊞ United States	8,431	40,391	59,107	24,819	132,748
	Total	10,835	53,118	77,908	32,648	174,509
⊟ Pacific	⊞ Australia			3,009	1,939	4,948
	Total			3,009	1,939	4,948
Grand Total		10,835	58,241	100,172	45,130	214,378

8. Repeat steps 4 through 7 to create one more report that uses the MOLAP cube. In step 4, select the MOLAP cube. In step 7, use the Geography MOLAP dimension.

You should now have a window in SSMS for the ROLAP cube and a window for the MOLAP cube. The values in both of the reports should be identical. Leave these reports open—you will use them in the next procedure.

Changing Data in a Warehouse

When you process an Analysis Services database, you update the information in the dimensions and measure groups based on the data stored in the data warehouse and the design of your database. If you change the design—for example, if you add a measure or a dimension to a cube—you must process the affected portions of the database. If the data in the data warehouse changes, as it inevitably will, you will also need to process the affected portions of your Analysis Services database.

The information in a data warehouse is almost always time-dependent. That means that at the very least, you'll continually add new time periods to your data warehouse. In time, you might also add additional products or additional geographic regions. When the data warehouse changes, you need to process the database to resynchronize your Analysis Services database with the relational data warehouse.

The FactResellerSales table in the SSAS2008SBS database contains data for six countries through June 2011. In the SSAS2008SBS database, the DimSalesTerritory dimension table includes only the six countries that appear in the fact table. The DimDate dimension table, however, includes months through December 2011. It is not uncommon in a warehouse to include months in the date dimension through the end of the current quarter or year, but to add members to other dimensions only as they are needed.

Included in the C:\Microsoft Press\Analysis Services 2008 SBS\Chapter 14\SQL folder is a SQL script named Update Warehouse 1.sql. This script adds Mexico to the DimSalesTerritory table and an additional row in the fact table for a sale in Mexico for December 2011. Inserting these records into the source database for the cubes simulates, on a very small scale, the load operations that occur regularly in a production data warehouse.

In the next procedure, you'll execute queries that insert data into the warehouse and then observe the effect of changed data on the cubes. You will see that the storage mode for a dimension or measure group partition affects what data is displayed by a cube when source data is changed.

Insert source data

1. On the SSMS File menu, point to Open and select File. In the Open File dialog box, browse to the C:\Microsoft Press\Microsoft Press\Analysis Services 2008 SBS\Chapter 14\SQL folder, select Update Warehouse 1.sql, and click Open.
2. On the Query menu, point to Connection and select Connect. In the Connect To Database Engine dialog box, change the Server Name to **localhost** and click Connect.
3. On the SSMS toolbar, click Execute. After the query executes successfully, close the Update Warehouse 1.sql Query window.
4. Select the ROLAP [Browse] window and on the Browser toolbar, click Reconnect. With Real-time ROLAP storage mode, changes to the source database immediately appear in the cube, because the dimension and partition data is stored only in the source database. The newly added country, Mexico, is displayed on rows in the Geography ROLAP dimension, and the number of items in the newly added order, 122, is displayed at the intersection of Mexico and CY 2011. Aggregate values have also been updated. The total for CY 2011 has increased by 122 to 45,252 and the total for all years and all countries has increased by 122 to 214,500.

Drop Filter Fields Here

		Calendar Year				
		CY 2008	CY 2009	CY 2010	CY 2011	Grand Total
Area	Country	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity
⊟ Europe	⊞ France		2,680	7,715	3,953	14,348
	⊞ Germany			4,480	2,900	7,380
	⊞ United Kingdom		2,443	7,060	3,690	13,193
	Total		5,123	19,255	10,543	34,921
⊟ North America	⊞ Canada	2,404	12,727	18,801	7,829	41,761
	⊞ Mexico				122	122
	⊞ United States	8,431	40,391	59,107	24,819	132,748
	Total	10,835	53,118	77,908	32,770	174,631
⊟ Pacific	⊞ Australia			3,009	1,939	4,948
	Total			3,009	1,939	4,948
Grand Total		10,835	58,241	100,172	45,252	214,500

5. Select the MOLAP [Browse] window and on the Browser toolbar, click Reconnect. The grand total for Order Quantity for all countries is still 214,378, and Mexico doesn't appear in the list of countries. All the values are unchanged from before the warehouse changed.

The MOLAP cube uses MOLAP storage, and it behaves as if you had not changed the data source. When you use MOLAP storage, with or without aggregates, the cube is completely detached from the data warehouse. You can even delete the warehouse database without affecting the Analysis Services database. With MOLAP storage, you must process a dimension or partition to resynchronize with the data warehouse.

Managing Analysis Services Processing

The easiest way to make sure that an Analysis Services database is completely consistent with the data warehouse, and with itself, is to process the entire database. When you process the database, you completely discard all the dimensions and measure group partitions within the database and create new ones. This takes place as a single transaction, which means that

client applications can continue to use the existing cubes until processing is complete. It also means that if an error occurs at any point during the processing, the entire change is rolled back, again ensuring that the database is internally consistent.

Processing the entire database is the simplest option, provided that you have sufficient time and storage space available. Although processing a large database can consume a considerable amount of time, users can continue to access the existing database while data is being updated in a new version of the database. For example, suppose you have an Analysis Services database that you update every day and that requires 10 hours to fully process. Assuming you have sufficient disk space, you could still choose to process the entire database, perhaps by starting the nightly processing after 19:00 or as soon as new data is available. Users would then have access to the updated database by the next morning. You would not have to exclude users from the system or wait for them to leave for the day.

Processing a large database can also consume a considerable amount of disk space. The Analysis Services server not only creates a second copy of all the dimension and partition files created during the transaction, but it also uses additional temporary files to accumulate aggregations, particularly when creating aggregations from a large fact table.

Consequently, some databases are simply too large to process as a single transaction. Analysis Services provides several options for processing individual components of a database. These options allow you to create and manage extremely large databases, but they also require much more work to provide users continuous access to the cubes and to prevent including invalid or inconsistent values in the database.

Processing a Dimension

When you process a dimension, the server creates a map that includes the path for each member of that dimension. Every cube that includes the dimension uses that map. When you process an existing dimension, the map is destroyed and a new map is created. Destroying the dimension map invalidates all the cubes that use the dimension. When you process an entire database, the dimensions are processed first and then all the cubes are processed as well. But when you process a single dimension, you make all the cubes that use that dimension inaccessible to client applications.

Fortunately, you can make certain changes to a dimension without destroying the existing map. If you don't destroy the existing map, you don't invalidate existing cubes. Analysis Services allows you to update a dimension to make changes that don't destroy the dimension map. The most useful change you can then make is adding new members to a dimension. You can also rename, delete, or re-parent members. For example, you could re-parent the Helmets member by changing its category from Accessories to Clothing.

Note If a dimension member has fact data associated with it in the cube, Analysis Services can't actually delete the member permanently during a Process Update. Instead, Analysis Services will "hide" the member in the dimension, leaving the fact data in the cube. The associated fact data will still be included in aggregated values, but it will not be displayed alongside a member in the dimension.

Unfortunately, when you update a dimension, you delete the aggregations related to that dimension. To avoid deleting aggregations when you update a dimension, you need to change the attribute relationship property *RelationshipType* from its default value of *Flexible* to *Rigid*. Changing this property allows you to update a dimension without deleting all of the related aggregations. However, if this property is set to *Rigid* and you try to update a dimension, Analysis Services will fail the operation if it detects changes in the dimension table that aren't allowed. Deleting or re-parenting members of a rigid relationship requires the dimension to be fully reprocessed, which in turn forces each cube using the dimension to be reprocessed.

The Date dimension is a good candidate for rigid attribute relationships, because you should not delete dates, and the month, quarter, and year that a date belongs to shouldn't change. A product dimension with categories and subcategories may not be a good candidate for rigid attribute relationships because products may be deleted or recategorized.

In the next procedure, you will learn how to change the value of the attribute relationship property *RelationshipType* from *Flexible* to *Rigid*.

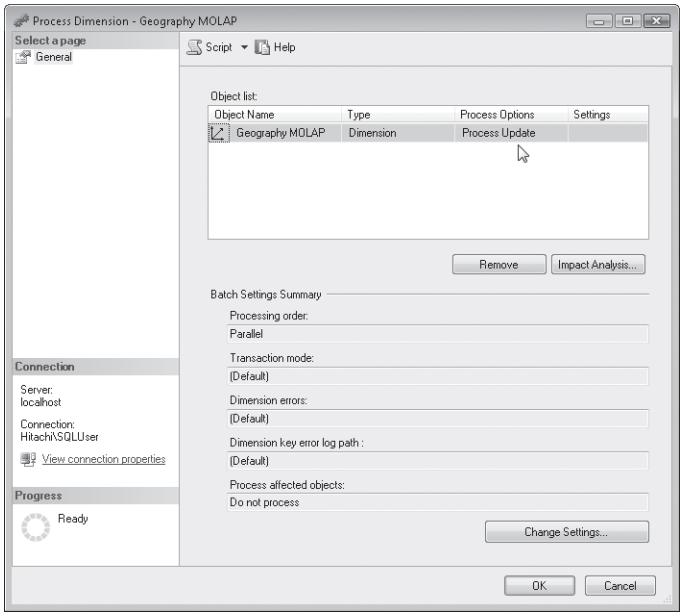
Create rigid attribute relationships

1. Switch to BIDS. In Solution Explorer, right-click Date.dim and select View Designer.
2. In the Attributes pane of the Dimension Designer, point at the Date dimension. Design Warnings will appear, including the warning "Define attribute relationships as 'Rigid' where appropriate." You should be careful about responding to this warning. Only define attribute relationships as Rigid if you are sure that when dimension members are deleted or re-parented, you will be able to fully reprocess the dimension and all cubes that contain the dimension.
3. Switch to the Attribute Relationships tab.
4. In the Attribute Relationships pane, right-click the Date – Month attribute relationship and select Edit Attribute Relationship.
5. In the Edit Attribute Relationship dialog box, change the Relationship Type to Rigid. Click OK. The Date - Month attribute relationship is now displayed with a solid arrow indicating that it is a rigid attribute relationship.
6. Repeat steps 4 and 5 for all of the other Date dimension attribute relationships.
7. Close the Dimension Designer and save the changes to the Date dimension.

In the procedure titled "Insert Source Data" earlier in the chapter, you inserted Mexico into the DimSalesTerritory dimension table, but Mexico has not yet appeared in the Geography MOLAP dimension. In the next procedure, you will update the dimension using SQL Server Management Studio so that it will contain this new member.

Update a dimension

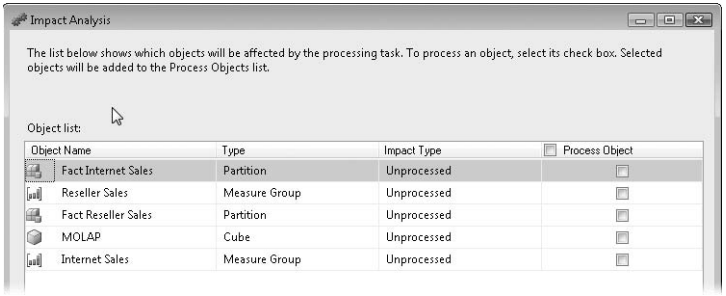
1. Switch to SQL Server Management Studio (SSMS). In Object Explorer, expand the Dimensions folder, right-click the Geography MOLAP dimension, and select Process. The Process Dimension dialog box is displayed.



Notice that the Process Options value is *Process Update*. Analysis Services suggests the processing option with the lowest impact that synchronizes the dimension with the database design and the data warehouse. In this case, because the dimension has already been processed and no structural changes are detected in the dimension design, Analysis Services determines that full processing is not required, and suggests an update instead.

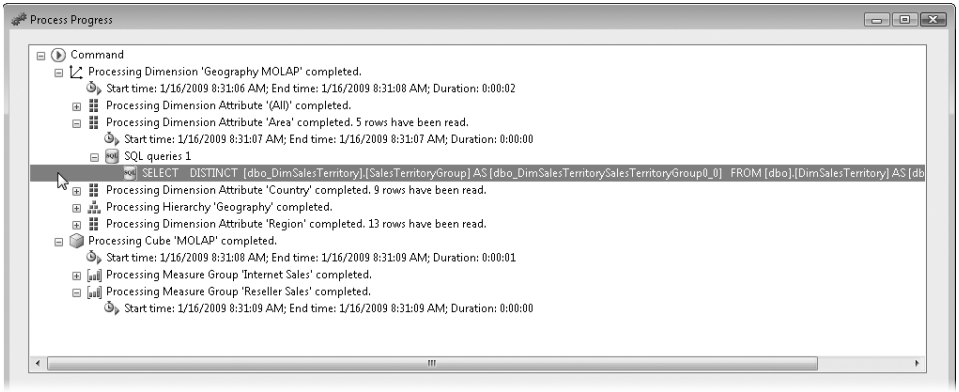
The processing option to update the dimension, Process Update, is appropriate when you make changes to the data in the dimension table. However, you will change this value to *Process Full* to observe the potential impact on other objects in the database. Notice the current setting for Process Affected Objects, visible at the bottom of the dialog box, is Do Not Process. Thus, any objects dependent on this dimension, such as partitions in the cube, will not be processed. You can review the impact of processing the Geography MOLAP dimension on other objects in the database by using the Impact Analysis feature.

2. In the Process Dimension dialog box, select Process Full from the Process Options list.
3. Click Impact Analysis. The Impact Analysis dialog box shows that the MOLAP cube and all of its measure group partitions would need to be processed if you were to perform a Process Full on the Geography MOLAP dimension. You can select objects in the Process Object column so that they will be processed immediately after the dimension is processed.



4. In the Impact Analysis dialog box, click Cancel. To avoid the need to process the MOLAP cube, change the processing options so that the dimension is updated.
5. In the Process Dimension dialog box, select Process Update from the Process Options list. Click Impact Analysis. The Impact Analysis dialog box shows that when you update a dimension, you don't need to process any other objects. You are now ready to update the dimension.
6. In the Impact Analysis dialog box, click Cancel. In the Process Dimension dialog box, click OK. The Process Progress dialog box appears, displaying the steps Analysis Services takes as it processes the dimension. When processing is complete, the Status will change to Process Succeeded. You can view additional information, including the SQL queries used to select data from the dimension table, by expanding the steps and substeps.

   The "Process Cube 'MOLAP' completed" step makes it appear that the MOLAP cube was processed. However, if you expand the substeps, you will see that no SQL query was executed, meaning that no data was loaded into any of the partitions.



7. In the Process Progress dialog box, click Close. Now that the Geography MOLAP dimension has been updated, you should be able to see the new member, Mexico, when you browse the MOLAP cube.
8. Select the MOLAP [Browse] window. On the Browser toolbar, click Reconnect. Mexico doesn't appear. Because the MOLAP cube was not processed, no values are associated with Mexico. You will be able to see Mexico if you choose to display empty cells.
9. Right-click the Report pane and select Show Empty Cells. The report should now display Mexico as a member of the Country attribute, like this:

Drop Filter Fields Here

		Calendar Year				
		CY 2008	CY 2009	CY 2010	CY 2011	Grand Total
Area	Country	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity
⊟ Europe	⊞ France		2,680	7,715	3,953	14,348
	⊞ Germany			4,480	2,900	7,380
	⊞ United Kingdom		2,443	7,060	3,690	13,193
	Total		5,123	19,255	10,543	34,921
⊟ NA	⊞ NA					
	Total					
⊟ North America	⊞ Canada	2,404	12,727	18,801	7,829	41,761
	⊞ Mexico					
	⊞ United States	8,431	40,391	59,107	24,819	132,748
	Total	10,835	53,118	77,908	32,648	174,509
⊟ Pacific	⊞ Australia			3,009	1,939	4,948
	Total			3,009	1,939	4,948
Grand Total		10,835	58,241	100,172	45,130	214,378

You will now learn about the options available to you when you process a cube, measure group, or partition. You will then complete a procedure that loads the Mexico sales order into the MOLAP cube.

Processing a Cube

When you click the Process command for a cube that has already been processed, the default processing option is always Process Full. When you fully process a cube, Analysis Services checks to see whether any design changes have been made to any of the dimensions used by the cube. If changes have been made, the server processes the dimensions before processing the cube. The server then generates a set of temporary files containing replacement data for the cube. As soon as processing has completed successfully, the server deletes the current files for the cube and renames the temporary files with the permanent names.

The Process Data option for a cube is virtually identical to the Full Process option. In both cases, the server generates all the files for a new cube, swapping the files into place when the processing is complete. The only real difference is that the Process Data option doesn't check to see whether you have made any changes to the dimension schema. It processes the cube using the existing dimension files.

Another option, Process Incremental, is both powerful and dangerous. The Process Incremental option allows you to process additional fact records into a cube. Analysis Services does not have the capability to identify which fact records are new. When you incrementally process a cube, you must provide Analysis Services with a SQL query that identifies the new records or identifies a table or view that contains the new records. You must be very careful to ensure that you do not include records that have already been processed into the cube and that you do not exclude any fact records. You can't use Process Incremental to delete or update records that have already been processed into the cube.

Process Incremental creates new cube files—precisely as if you were using the Process Full option. When the processing is complete, however, the server doesn't replace the old files with the new ones. Rather, it merges the two sets of files, creating a third set of cube files. Finally, it deletes all but the third set of files and renames those files to become the final cube files. One implication of this operation is that for a single cube, the Process Incremental option might actually require more disk space than the Process Full option because it creates three sets of files, rather than just two. A more important implication is that if you use the Process Incremental option using a fact table that includes values already stored in the cube, those values will be double-counted after you process the cube. An alternate option, Process Data, simply clears out the data in the cube structure and reloads data from the fact table as defined for each partition.

In the procedure titled "Insert Source Data" earlier in this chapter, you inserted a sales order from Mexico on December 31, 2011, into the FactResellerSales fact table, but this sales order has not yet appeared in the MOLAP cube. In the next procedure, you will use the Process Incremental processing option to add this record to the cube.

Incrementally process a partition

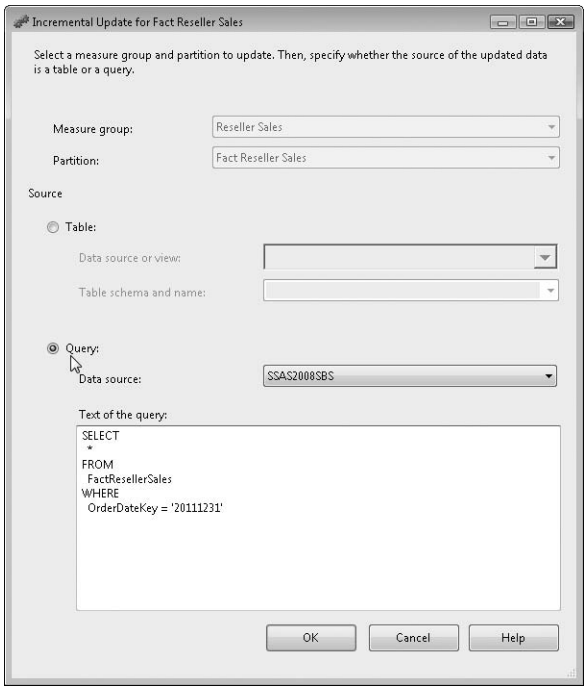
1. In the SSMS Object Explorer, expand the MOLAP cube, expand the Measure Groups folder, expand the Reseller Sales measure group, and expand the Partitions folder.
2. Right-click the Fact Reseller Sales partition and select Process.
3. In the Process Partition dialog box, select Process Incremental from the Process Options list. In the Settings column, click Configure. In the Incremental Update dialog box, you can select the table or view that contains the fact records that you want to add to the partition or you can enter a SQL query that selects the additional fact records.

Important Analysis Services does not have the capability of identifying duplicate or missing fact records. When you perform an incremental process, you must ensure that the table, view, or SQL query that you use contains all of the additional records that should be processed into the cube and doesn't contain any fact records that have already been processed into the cube.

4. In the Incremental Update dialog box, select Query. In the Text Of The Query text box, enter the following SQL query, or you can copy the code from the file C:\Microsoft Press\Analysis Services 2008 SBS\Chapter 14\SQL\Incremental Process.txt.

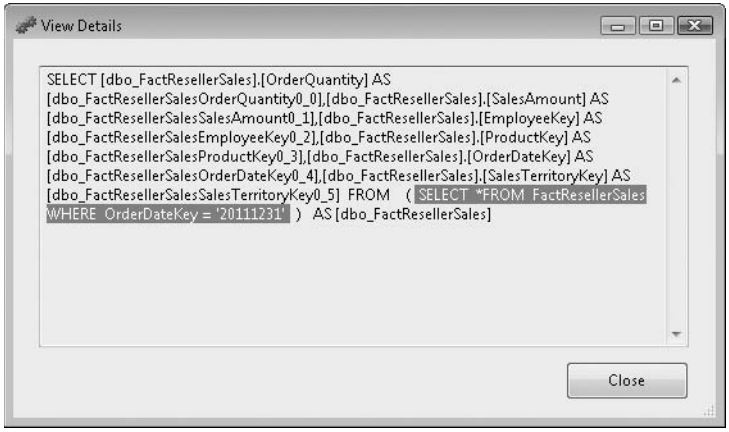
```
SELECT
  *
FROM
  FactResellerSales
WHERE
  OrderDateKey = '20111231'
```

The Incremental Update dialog box should look like the following image.



5. Click OK. In the Process Partition dialog box, click OK. The Process Progress dialog box will appear and display the steps Analysis Services takes as it processes the partition. When processing is complete, the Status will change to Process Succeeded. You can view additional information, including the SQL query used to select data from the fact table, by expanding the steps and substeps.

6. In the Process Progress dialog box, expand all of the steps and substeps and then select the substep that begins SELECT [dbo_FactResellerSales].[OrderQuantity] AS. Click View Details. The View Details dialog box displays the SQL query that Analysis Services used to select records from the fact table. The SQL query you entered in step 4 appears as a subquery.



7. In the View Details dialog box, click Close, and then click Close in the Process Progress dialog box.

8. Select the MOLAP [Browse] window. On the Browser toolbar, click Reconnect. The report shows that the sales order from Mexico has been added to the MOLAP cube. The number of items in the newly added order, 122, is displayed at the intersection of Mexico and CY 2011. Aggregate values have also been updated. The total for CY 2011 has increased by 122 to 45,252 and the total for all years and all countries has increased by 122 to 214,500.

Drop Filter Fields Here

		Calendar Year				
		CY 2008	CY 2009	CY 2010	CY 2011	Grand Total
Area	Country	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity	Reseller Order Quantity
⊟ Europe	⊞ France		2,680	7,715	3,953	14,348
	⊞ Germany			4,480	2,900	7,380
	⊞ United Kingdom		2,443	7,060	3,690	13,193
	Total		5,123	19,255	10,543	34,921
⊟ NA	⊞ NA					
	Total					
⊟ North America	⊞ Canada	2,404	12,727	18,801	7,829	41,761
	⊞ Mexico				122	122
	⊞ United States	8,431	40,391	59,107	24,819	132,748
	Total	10,835	53,118	77,908	32,770	174,631
⊟ Pacific	⊞ Australia			3,009	1,939	4,948
	Total			3,009	1,939	4,948
Grand Total		10,835	58,241	100,172	45,252	214,500

As you've learned in this chapter, keeping an Analysis Services database up to date can be challenging. You have to consider when to process changes to database objects relative to changes in the data warehouse, how processing one object impacts other objects, and how the processing option you use affects user queries. In the next section, you will learn about proactive caching, an Analysis Services feature that simplifies database processing by managing the details for you.

Proactive Caching

The data that Analysis Services needs to respond to a query can exist in one of three storage locations, also called *caches*. Analysis Services can respond to a query most quickly if the data exists in memory. If the data is not in memory, Analysis Services can use the second-quickest option and retrieve the data from MOLAP disk storage. If the data is not available from memory or MOLAP storage, the slowest option is to retrieve the data from ROLAP storage—that is, Analysis Services has to query the source relational database.

A potential problem arises when the data in the source relational database changes. These changes cause the ROLAP cache to be out of sync with the memory and MOLAP caches. To get the three caches back in sync, you must process the dimensions and partitions that use MOLAP or HOLAP storage, and the memory cache must be cleared and repopulated. Because processing dimensions and partitions takes time and computer resources, you need to decide when you want this processing to occur and what cache(s) should be used to respond to queries while the caches are out of sync. Fortunately, Analysis Services provides proactive caching, which simplifies managing these issues.

To configure proactive caching, you need to consider the following questions:

- Should processing of the cube occur in fixed intervals of time or only when data in the underlying source has changed?
- While the cube is being processed, how should user queries be resolved—from the most recent version of the cube (which might contain old data) or from the underlying source relational database (which contains new data, but might be slower to return query results)?
- If processing will be triggered by changes to the source database, how should Analysis Services be notified of a change?

In the next procedure, you will review the available options and configure proactive caching for a partition.

Configure partition proactive caching

1. Switch to Business Intelligence Development Studio (BIDS). In Solution Explorer, right-click MOLAP.cube and select View Designer. In the Cube Designer, click the Partitions tab.
2. On the Partitions tab, expand the Internet Sales measure group, right-click the Fact Internet Sales partition, and select Storage Settings.
3. In the Partition Storage Settings dialog box, drag the slider to Scheduled MOLAP. Click Options. Notice that Proactive Caching is enabled, and the cache is configured to rebuild in one-day intervals. In this case, the MOLAP storage for this partition will automatically update once every day, unless you change the setting to a shorter interval. Unfortunately, you can change only the frequency of the rebuild, not the time of day that the rebuild takes place. The Rebuild Interval can be as often as every second or it can be set as an interval of many days.

   By using Scheduled MOLAP, you force a periodic update of the MOLAP cache, whether or not new data has appeared in the warehouse. This may result in more frequent processing of the partition than necessary if data is not regularly added to the warehouse. On the other hand, the partition may not be processed frequently enough if the intervals are too long relative to the frequency of updates to the warehouse.
4. Click Cancel. In the Partition Storage Settings dialog box, drag the slider to Automatic MOLAP. Click Options. Automatic MOLAP also enables proactive caching, but updates the cache only when the data changes, instead of on a periodic basis. The default settings start cache processing after a 10-second Silence Interval, with a Silence Override Interval of 10 minutes. The Silence Interval setting prevents processing from starting until data updates in the warehouse have completely stopped for the specified interval. This situation is analogous to waiting for rush hour traffic to end. If your warehouse is routinely updated during the first 10 minutes of every hour, that period of time is the rush period. As soon as 10 seconds (or the amount of time specified by the Silence Interval) have elapsed, the cache is processed.

But what if there are so many updates to the fact table during the rush period that processing cannot start after the expected 10-minute duration? This Silence Override Interval setting tells the Analysis server to go ahead and start processing the cache if 10 minutes have elapsed and data is still being added to the warehouse. The server uses a snapshot to isolate the records that it will include in the update process from the records that are added to the warehouse after processing has begun.

5. Click the Notifications tab. You use the notification options to specify the conditions that indicate that data has changed in the warehouse. The default value is SQL Server. Any dimension and fact tables used for the current partition are monitored for changes, but you can also specifically identify tables to be monitored. Alternatively, you can choose to have a client application send notification of changes by using the Client Initiated option, or you can choose to poll specific tables by using the Scheduled Polling option. Scheduled Polling is useful if your warehouse is stored in a relational database other than SQL Server. For more information about polling queries, refer to SQL Server Books Online.

Note In order to use SQL Server notifications, either the data source or the Analysis Services service account must be configured to use an account that is either a member of the sysadmin server role or has been granted Alter trace permissions on the SQL Server database server.

6. Click Cancel. In the Partition Storage Settings dialog box, drag the slider to Medium-Latency MOLAP and click Options. The settings on this page are similar to the Automatic MOLAP settings, but now the Drop Outdated Cache option is enabled with a default value for Latency of 4 hours. With medium-latency MOLAP, the Analysis Services server will eliminate the MOLAP cache if it hasn't been processed within the past four hours (or the period that you specify if you change the default value).

   When the MOLAP cache is dropped by the server, any queries will be answered from the relational data source (ROLAP cache) until a new MOLAP cache is created. Notice that the Bring Online Immediately option is enabled for medium-latency MOLAP. This option tells the Analysis server that as soon as it drops the outdated cache, queries should be resolved from the relational data source until the new MOLAP cache is available. This setting is useful when processing has started but has not completed within a desired period of time, providing users with relatively current data in response to queries. If you disable this option and you have the Drop Outdated Cache option enabled, queries cannot be answered until the new MOLAP cache is created. If you prefer to continue answering queries with the old cache while a new cache is being built, you must disable the Drop Outdated Cache option.

7. Click Cancel. In the Partition Storage Settings dialog box, drag the slider to Low-latency MOLAP. Click Options. The only difference between medium-latency MOLAP and low-latency MOLAP is that the *Latency* property changes from 4 hours to 30 minutes.

8. Click Cancel. In the Partition Storage Settings dialog box, drag the slider to Real-time HOLAP. Click Options. The storage mode has now been changed to HOLAP. In HOLAP storage, only summarized aggregations are stored in MOLAP storage. All queries that require detail data will be sent directly to the relational database.

   Because the Silence Interval value is 0, the MOLAP aggregations will be rebuilt immediately when Analysis Services is notified of a change to the data. The Drop Outdated Cache option causes the Analysis Services memory cache to be cleared as soon as there is a change to the data source fact table.

9. Click Cancel. In the Partition Storage Settings dialog box, drag the slider to Real-time ROLAP. Click Options. The storage mode has now changed to ROLAP. The Update The Cache When Data Changes option has been cleared, because no MOLAP cache will be created. Similar to HOLAP storage, the Drop Outdated Cache option causes the Analysis Services memory cache to be cleared as soon as there is a change to the data source fact table.

10. Click Cancel. In the Partition Storage Settings dialog box, drag the slider to Automatic MOLAP. Click OK. You have configured the partition to use Automatic MOLAP proactive caching to manage processing. The partition will automatically be processed starting 10 seconds after the last update to the source database. If 10 minutes pass since the first update to the source database and there has not been a 10-second silence interval, processing will begin.

11. Close the Cube Designer and save the changes to the MOLAP cube.

You can also configure proactive caching for dimensions. In the next procedure, you will configure Automatic MOLAP proactive caching for the Geography MOLAP dimension.

Configure dimension proactive caching

1. In Solution Explorer, right-click Geography MOLAP.dim and select View Designer.
2. In the Attributes pane of the Dimension Designer, right-click the Geography MOLAP dimension and select Properties.
3. In the Properties window, select the *ProactiveCaching* property and click the ellipsis button that appears on the right.
4. In the Dimension Storage Settings dialog box, drag the slider to Automatic MOLAP. Click Options. The same options are available for dimension and partition proactive caching.
5. Click OK. In the Dimension Storage Settings dialog box, click OK.
6. Close the Dimension Designer and save the changes to the Geography MOLAP dimension. In the next procedure, you will have an opportunity to see proactive caching in action, so you need to deploy your changes to the Analysis Services server.
7. On the Build menu, select Deploy AdventureWorks SSAS.

In the next procedure, you will create a report using Internet sales data. You will then execute a SQL script that will insert Japan into the DimSalesTerritory table and add a record for a sale in Japan to the FactInternetSales table. You will then refresh the report and see the order for Japan, even though you haven't processed the cube.

Insert source data and browse the cubes

1. Switch to SSMS. In the MOLAP [Browse] window, click Reconnect and then remove Reseller Order Quantity from the Report pane.
2. In the Metadata pane, expand the Internet Sales measure group and drag Internet Order Quantity to the Report pane totals area. Notice that Japan is not one of the countries listed in the report.

Drop Filter Fields Here

		Calendar Year				
		CY 2008	CY 2009	CY 2010	CY 2011	Grand Total
Area	Country	Internet Order Quantity	Internet Order Quantity	Internet Order Quantity	Internet Order Quantity	Internet Order Quantity
⊟ Europe	⊞ France	59	233	2,291	2,799	5,382
	⊞ Germany	76	233	2,254	2,897	5,460
	⊞ United Kingdom	96	265	2,966	3,382	6,709
	Total	231	731	7,511	9,078	17,551
⊟ NA	⊞ NA					
	Total					
⊟ North America	⊞ Canada	47	226	3,086	3,783	7,142
	⊞ Mexico					
	⊞ United States	341	861	8,511	10,811	20,524
	Total	388	1,087	11,597	14,594	27,666
⊟ Pacific	⊞ Australia	394	859	5,335	6,384	12,972
	Total	394	859	5,335	6,384	12,972
Grand Total		1,013	2,677	24,443	30,056	58,189

3. On the SSMS File menu, point to Open and select File. In the Open File dialog box, browse to the C:\Microsoft Press\Analysis Services 2008 SBS\Chapter 14\SQL folder, select Update Warehouse 2.sql, and click Open.
4. On the Query menu, point to Connection and select Connect. In the Connect To Database Engine dialog box, change the Server Name to **localhost** and click Connect.
5. On the SSMS toolbar, click Execute. After the query executes successfully, close the query window. Proactive caching has been configured to automatically process the dimension after a 10-second quiet interval, so you may need to wait just a brief moment before refreshing the report.
6. Back in the MOLAP [Browse] window, click Reconnect on the Browser toolbar. The report shows that the sales order from the Internet order from Japan that has been added to the MOLAP cube. The number of items in the newly added order, 1, is displayed at the intersection of Japan and CY 2011. Aggregate values have also been updated. The total for CY 2011 has increased by 1 to 30,057, and the total for all years and all countries has increased by 1 to 58,190.

Drop Filter Fields Here

		Calendar Year				
		CY 2008	CY 2009	CY 2010	CY 2011	Grand Total
Area	Country	Internet Order Quantity	Internet Order Quantity	Internet Order Quantity	Internet Order Quantity	Internet Order Quantity
⊟ Europe	⊞ France	59	233	2,291	2,799	5,382
	⊞ Germany	76	233	2,254	2,897	5,460
	⊞ United Kingdom	96	265	2,966	3,382	6,709
	Total	231	731	7,511	9,078	17,551
⊟ NA	⊞ NA					
	Total					
⊟ North America	⊞ Canada	47	226	3,086	3,783	7,142
	⊞ Mexico					
	⊞ United States	341	861	8,511	10,811	20,524
	Total	388	1,087	11,597	14,594	27,666
⊟ Pacific	⊞ Australia	394	859	5,335	6,384	12,972
	⊞ Japan				1	1
	Total	394	859	5,335	6,385	12,973
Grand Total		1,013	2,677	24,443	30,057	58,190

Tip If you want to work through the procedures in this section a second time, you first need to delete the records that were inserted by the Update Warehouse 1.sql and Update Warehouse 2.sql scripts. Included in the C:\Microsoft Press\Analysis Services 2008 SBS \Chapter 14\SQL folder is a SQL script named Delete.sql that will do this for you.

Working with Partitions

Partitions make it possible for you to create extremely large cubes. You can effectively create small, medium-sized, and even remarkably large cubes without using partitions. But partitions are useful when you need to create very large, enterprise-wide applications. For that reason, the ability to manage multiple partitions is available only with Microsoft SQL Server 2008 Enterprise Edition.

Understanding Partition Strategies

Each Analysis Services measure group consists of at least one partition. You design storage modes and aggregations at the partition level. Whether a measure group contains only a single partition or many partitions, the process of designing storage is the same.

One of the benefits of creating multiple partitions is that you can design different storage for different portions of the measure group. For example, say that you have one partition that contains information for the current year and one previous year. You access this information frequently, so you specify multidimensional OLAP (MOLAP) storage with aggregations to provide a 30 percent performance gain. A second partition contains values for the third, fourth, and fifth years. These years are usually accessed only at a summary level (if at all), and the relational warehouse is also occasionally accessed, so you specify hybrid OLAP (HOLAP) storage, with aggregations that provide a 15 percent performance gain. A third partition contains several previous years. Those years are infrequently accessed and the relational warehouse is never used, so you specify MOLAP storage with aggregations to the 5 percent performance level and then archive the relational warehouse to tape.

A second major benefit of creating partitions is that you can process a partition independently of the rest of the cube. As a fairly extreme example, suppose that you have an Analysis Services cube used to monitor manufacturing activities and you want to update the information in that cube every 10 minutes. You don't have time to completely process the database every 10 minutes. By putting the current day into a separate partition, you can process that partition every 10 minutes, without having to process the rest of the cube. In effect, creating a partition for the current day is like performing an incremental update on the cube, except that you can completely replace the values in that one partition every 10 minutes, guaranteeing consistency with the relational data source.

A client application has no awareness of—let alone control over—partitions used on the server. You can modify the design of partitions without affecting any client application. The most important task when creating partitions is to make sure that each appropriate value from the fact table (or fact tables) makes it into one and only one partition.

Creating Partitions

When you're creating partitions, make sure each partition gets unique data. Otherwise, it's easy to double-count values in multiple partitions. The dangers of creating partitions are similar to the dangers of executing an incremental update on a cube. This similarity is not coincidental. In fact, when you perform an incremental update on a cube, the Analysis Services server creates a new partition, loads values into the new partition, and then merges the two partitions. Analysis Services provides two techniques to avoid double-counting:

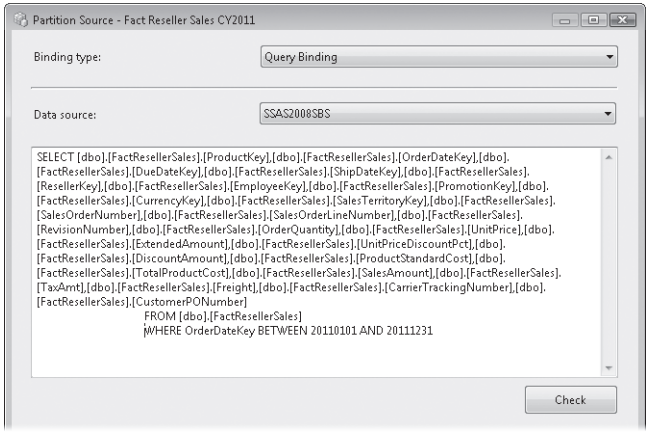
- Create a separate fact table for each partition.
- Specify a filter (a SQL WHERE clause) to restrict rows from the fact table.

In this section, you will create three partitions in the AdventureWorks cube Reseller Sales measure group. One partition will contain data for CY 2011, another for CY 2010, and a final partition will contain data for all prior years. In the first procedure, you will modify the existing partition so that it contains the most current data, and then in the following procedure you will create two partitions to contain data from prior years.

Modify a partition

1. Switch to BIDS. In the BIDS Solution Explorer, right-click AdventureWorks.cube and select View Designer. In the Cube Designer, click the Partitions tab.
2. In the Reseller Sales measure group, right-click the Fact Reseller Sales partition and select Properties. In the Properties window, change the name of the partition to **Fact Reseller Sales CY 2011**.
3. In the Properties window, select the *Source* property and then click the ellipsis button that appears on the right. You can ensure that the data in a partition is unique by having each partition select data from a different table or by having each partition select records using a SQL *SELECT* statement that includes a WHERE clause. The WHERE clause for each partition must filter a unique set of fact records for each partition. In the Partition Source dialog box, you choose whether the partition will be retrieving data from a table or whether it will use a SQL query.
4. In the Partition Source dialog box, select Query Binding from the Binding Type list. When you select Query Binding, the *SELECT* statement that the partition is currently using appears followed by WHERE. You need to complete the WHERE clause so that the partition selects data for CY 2011.

5. In the Query text box, change the WHERE clause to **WHERE OrderDateKey BETWEEN 20110101 AND 20111231**. The Partition Source dialog box should look like this:



6. Click Check. This will check the SQL query syntax, but it is up to you to ensure that partitions do not contain duplicated data by setting the WHERE clause appropriately. In the Partition Filter dialog box, click OK. In the Partition Source dialog box, click OK.

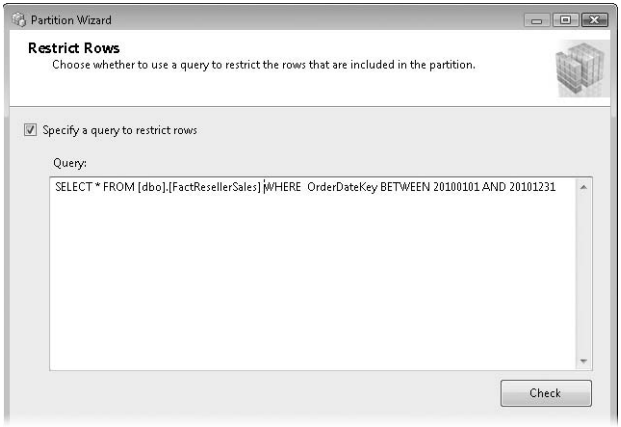
Important Changing a partition source from table binding to query binding breaks the link between partition and the data source view. If you later make changes to the fact table in the data source view, those changes *will not* get propagated to the partition. You will need to manually update each partition.

Now that you have modified the original partition so that it only contains CY 2011 data, you can create two other partitions to contain the rest of the fact table data.

Create partitions

1. On the Partitions tab of the Cube Designer, click the New Partition link in the Reseller Sales measure group section.
2. On the Welcome page of the Partition Wizard, click Next. On the Specify Source Information page, the Measure Group value is Reseller Sales. This value defines the measure group for the new partition. The Look In value defines which data source or data source view contains the source table for the partition. Any tables that match the structure of the measure group's source table will be displayed. If multiple tables are selected from the Available Tables list, a new partition is created for each table.
3. In The Available Tables list, select FactResellerSales. Click Next.
4. On the Restrict Rows page, select Specify A Query To Restrict Rows. In the Query text box, change the SQL *SELECT* statement to **SELECT * FROM [dbo].[FactResellerSales]**

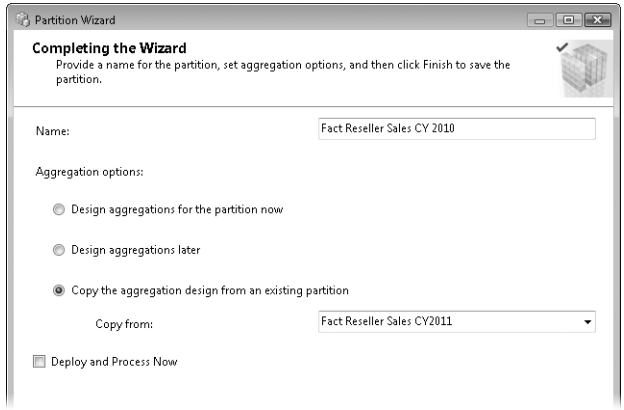
WHERE OrderDateKey BETWEEN 20100101 AND 20101231. The Restrict Rows page of the wizard should look like this:



Tip If your partition is selecting data from a single fact table and there are no named calculations, in the SQL query you should replace the list of column names with *. By eliminating the use of specific column names, you will not be required to modify the partition if you later modify the columns in the fact table.

5. Click Check. In the Partition Filter dialog box, click OK. On the Restrict Rows page of the wizard, click Next. If you have a very large Analysis Services database, you can allocate a measure group's partitions across multiple Analysis Services servers. This allows multiple servers to share the processing and query load. To learn more, see the SQL Server Books Online article "Creating and Managing a Remote Partition." You can also allocate a measure group's partitions across multiple storage locations. The Processing And Storage Locations page of the Partition Wizard allows you to configure these options. You'll accept the default options for this procedure.
6. On the Processing And Storage Locations page, click Next.
7. On the Completing The Wizard page, change the name of the partition to **Fact Reseller Sales CY 2010**.

   A new partition needs to have an aggregation design. You can choose to create a new aggregation design using the Aggregation Design Wizard, copy an aggregation design from an existing partition, or create or copy an aggregation design later. For this partition, you will copy the aggregation design used by the Fact Reseller Sales CY 2011 partition.
8. Select Copy The Aggregation Design From An Existing Partition. This will copy the aggregation design from the partition in the Copy From list. The CY 2010 partition will be assigned the 30 Percent AggregationDesign aggregation design, the same one used by the CY 2011 partition. The Completing The Wizard page should look like the following.



9. Click Finish. The Reseller Sales measure group now contains two partitions. Notice that both partitions are using the 30 Percent AggregationDesign. In the next step, you will create one more partition that will contain the data from all prior years. Because this historical data will be queried less frequently than the more recent data, you can choose to have fewer aggregations. With fewer aggregations, the partition can be processed more quickly and will use less storage space.

10. Repeat steps 1 through 9 to create one more partition with the properties shown in the following table.

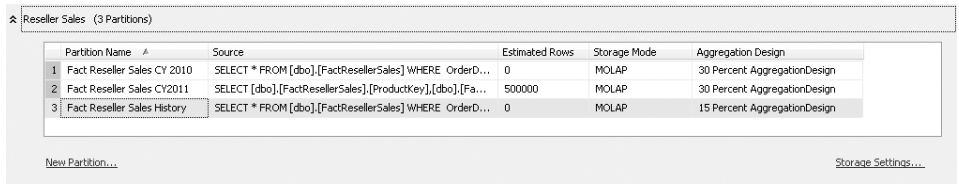
SQL Query	**SELECT * FROM [dbo].[FactResellerSales] WHERE OrderDateKey < 20100101**
Name	**Fact Reseller Sales History**
Aggregation Option	Design Aggregations Later

11. In the Cube Designer, click the Aggregations tab.

12. On the Aggregations tab, right-click 15 Percent AggregationDesign in the Reseller Sales measure group section and select Assign Aggregation Design.

13. In the Assign Aggregation Design dialog box, select Fact Reseller Sales History from the Destination Partitions list.

14. Click OK and then click the Partitions tab. The Reseller Sales measure group should now contain three partitions. The CY 2010 and CY 2011 partitions should be using the 30 Percent AggregationDesign, and the history partition should be using the 15 Percent AggregationDesign.



One benefit of having multiple partitions is that you can process the partitions individually, or you can take advantage of parallel processing and process several partitions at the same time. In the next procedure, you will demonstrate these capabilities.

Process partitions

1. In the Cube Designer, right-click the Fact Reseller Sales CY 2011 partition on the Partitions tab and then select Process. A dialog box may appear with the warning that the server content appears to be out of date. Click Yes and the project will be deployed to the server. When deployment is complete, the Process Partition dialog box will appear.
2. In the Process Partition dialog box, click Run. When processing is complete, the status will change to Process Succeeded.
3. In the Process Progress dialog box, click Close, and then click Close in the Process Partition dialog box.
4. Click the Browser tab of the Cube Designer.
5. In the Browser tab Metadata pane, expand the Measures folder, expand the Reseller Sales measure group, and drag Reseller Order Quantity to the Report pane totals area.
6. In the Metadata pane, expand the Date dimension and drag Calendar Year to the Report pane rows area. Right-click the Report pane and select Show Empty Cells.

   Because you processed the Fact Reseller Sales CY 2011 partition and did not process the other partitions, the report only displays CY 2011 data.

Drop Filter Fields Here	
	Drop Column Fields Here
Calendar Year	Reseller Order Quantity
CY 2008	
CY 2009	
CY 2010	
CY 2011	45,252
Grand Total	45,252

   You will need to process the other two partitions in the Reseller Sales measure group so that you can see all four years of data.
7. Click the Partitions tab. Click the Fact Reseller Sales CY 2010 partition. Hold down Ctrl and click the Fact Reseller Sales History partition so that both partitions are selected. On the Partitions tab toolbar, click Process. In the Process Object(s) dialog box, click Run.
8. In the Process Progress dialog box, click Close, and then click Close in the Process Partition dialog box.
9. Click the Browser tab and click Reconnect. The report is refreshed and now displays data for four years.

Drop Filter Fields Here	
	Drop Column Fields Here
Calendar Year	Reseller Order Quantity
CY 2008	10,835
CY 2009	58,241
CY 2010	100,172
CY 2011	45,252
Grand Total	214,500

Consider the situation described earlier in the section titled "Understanding Partition Strategies," where you created a new partition each day for a manufacturing cube. Each month you would create up to 31 additional partitions in the cube. Simply keeping the partitions straight would be extraordinarily confusing. One solution is to use only two partitions: one for the current day and one for all previous time. Each night, merge the current day partition with the previous time partition, and then create a new current day partition for the next day. Merged partitions don't run significantly faster than separate partitions, but they can be much easier to manage.

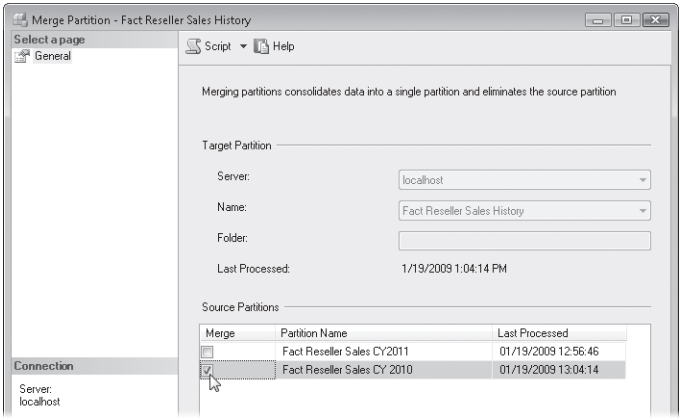
In the next procedure, you will learn how to merge the CY 2010 partition with the partition containing historical data.

Merge partitions

1. Switch to SSMS. Expand the AdventureWorks cube, expand the Measure Groups folder, expand the Reseller Sales measure group, and expand the Partitions folder.
2. Right-click the Fact Reseller Sales History partition and select Merge Partitions.

Important The partition you select before opening the Merge Partition dialog box is the partition that will be retained after the merge.

3. In the Merge Partition dialog box, select Fact Reseller Sales CY 2010 from the Source Partitions list. The Merge Partitions dialog box should look like this:



4. Click OK. Right-click the Partitions folder and select Refresh. You can see that the Fact Reseller Sales CY 2010 partition no longer appears in the list of the Reseller Sales measure group partitions. The data that was in the CY 2010 partition has been merged into the history partition.

Important Merging partitions does not update the partition query, nor does it combine fact tables. If you think you might reprocess the partition, you need to update the partition query or move the records from the fact table corresponding to the source partition into the fact table corresponding to the target partition.

You will need to modify Fact Reseller Sales History partition so that it selects all fact records dated prior to January 2011.

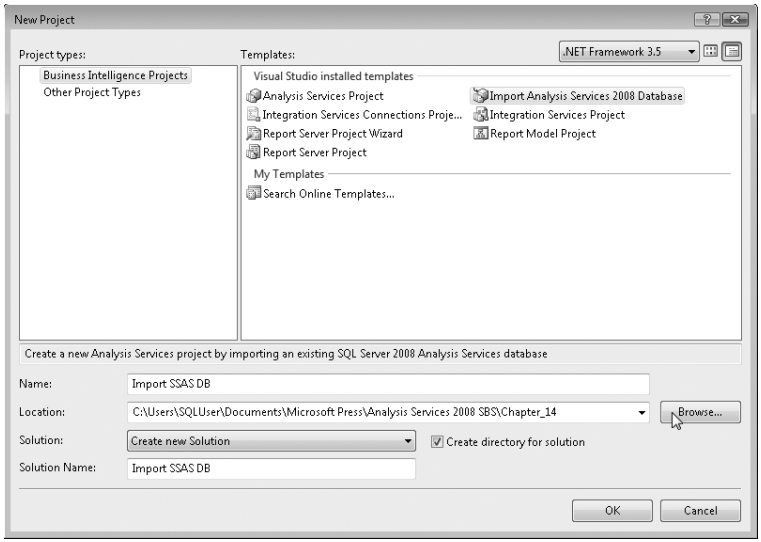
5. Right-click the Fact Reseller Sales History partition and select Properties.
6. On the General page of the Partition Properties dialog box, select the *Source* property and click the ellipsis button that appears on the right.
7. Change the WHERE clause of the SQL query to **WHERE OrderDateKey < 20110101**.
8. Click OK. In the Partition Properties dialog box, click OK. Close SSMS.

Because the partitions have been merged in SQL Server Management Studio, the AdventureWorks SSAS database that is deployed on the Analysis Services server is no longer synchronized with the solution that is currently open in BIDS. Whenever changes are made to an Analysis Services database in SSMS, you should import those design changes into a new project in BIDS. Then use the new project for any future design changes that will be deployed back to the Analysis Services server.

In the next procedure, you will learn how to import an Analysis Services database design into a new project in BIDS.

Import an Analysis Services database design

1. On the File menu in BIDS, point to New and select Project.
2. In the New Project dialog box, verify that the project type is Business Intelligence Projects. In the Templates pane, select Import Analysis Services 2008 Database.
3. Change the project name to **Import SSAS DB** and change the location to C:\Microsoft Press\Analysis Services 2008 SBS\Chapter 14. The New Project dialog box should look like the following image.



When you click OK, the AdventureWorks BI solution will be closed and the Import Analysis Services Database Wizard will start.

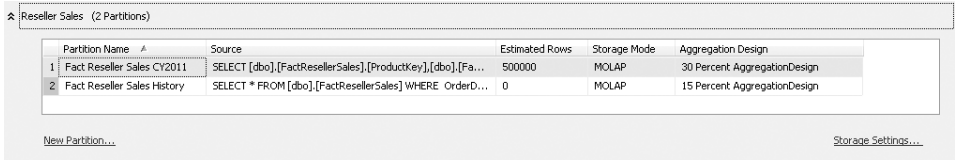
4. Click OK. On the Welcome page of the Import Analysis Services Database Wizard, click Next.

5. On the Source Database page, enter **localhost** in the Server text box and then select AdventureWorks SSAS from the Database list.

6. Click Next. On the Completing The Wizard page, click Finish. You should look at the partitions in the Reseller Sales measure group to confirm that the changes you made in SSMS have been imported into this new project.

7. In the Cubes folder of Solution Explorer, right-click AdventureWorks.cube and select View Designer.

8. In the Cube Designer, click the Partitions tab. The Reseller Sales measure group should contain only the Fact Reseller Sales CY 2011 and Fact Reseller Sales History partitions.



9. Close BIDS.

In this chapter, you learned that Analysis Services has three storage modes: ROLAP, HOLAP, and MOLAP. ROLAP mode leaves dimension, detail, and aggregate data in the source relational database; MOLAP stores dimension, detail, and aggregate data in proprietary data structures on the Analysis Services server; and HOLAP leaves detail data in the source relational database and stores aggregate data on the Analysis Services server. Cube data is stored in measure group partitions. A measure group can have multiple partitions. Partitions in a measure group can have different storage modes and can use different aggregation designs. You can add, delete, or merge the partitions in a measure group.

You also learned that when the data in a source database is modified, you need to reprocess MOLAP dimensions and partitions to bring those changes into an Analysis Services database. The most simple and reliable method is to fully process all affected dimensions and partitions. However, if the source database is large, time might not allow for a full process. Analysis Services allows you to update dimensions and to use incremental processing to add new source data into a measure group partition. Keeping a source database and an Analysis Services database in sync can become quite complicated. Analysis Services provides proactive caching that allows you to automate Analysis Services processing.