CHAPTER 1

What Is Business Intelligence?

This chapter presents a blueprint for understanding the exciting potential of SQL Server 2005's BI technologies to meet your company's crucial business needs. It describes tools, techniques, and high-level implementation concepts for BI.

This chapter covers:

- Defining Business Intelligence
- Understanding BI from an end-user perspective
- Understanding the business problems BI addresses

Just What Is BI?

Business Intelligence (*BI*) is defined in many ways. Often particular vendors "craft" the definition to show their tools in the best possible light. For the purposes of this book, Microsoft's vision of BI using SQL Server 2005 is defined as

> *Business Intelligence is a method of storing and presenting key enterprise data so that anyone in your company can quickly and easily ask questions of accurate and timely data. Effective BI allows end users to use data to understand why your business got the particular results that it did, to decide on courses of action based on past data, and to accurately forecast future results.*
>
> *BI data is displayed in a fashion that is appropriate to each type of user, i.e. analysts will be able to drill into detailed data, executives will see timely summaries, and middle managers will see data presented at the level of detail that they need to make good business decisions. Microsoft's BI uses cubes, rather than tables, to store information and presents information via reports. The reports can be presented to end users in a variety of formats: Windows applications, Web Applications, and Microsoft BI client tools, such as Excel or SQL Reporting Services.*

Figure 1-1 shows a sample of a typical BI physical configuration. You'll note that Figure 1-1 shows a Staging Database Server and a separate BI server. Although it is possible to place all components of BI on a single physical server, the configuration shown in the figure is the most

typical for the small-to-medium BI projects that I've worked on. You may also need to include more servers in your project, depending on scalability and availability requirements. You'll learn more about these concepts in Chapter 13.

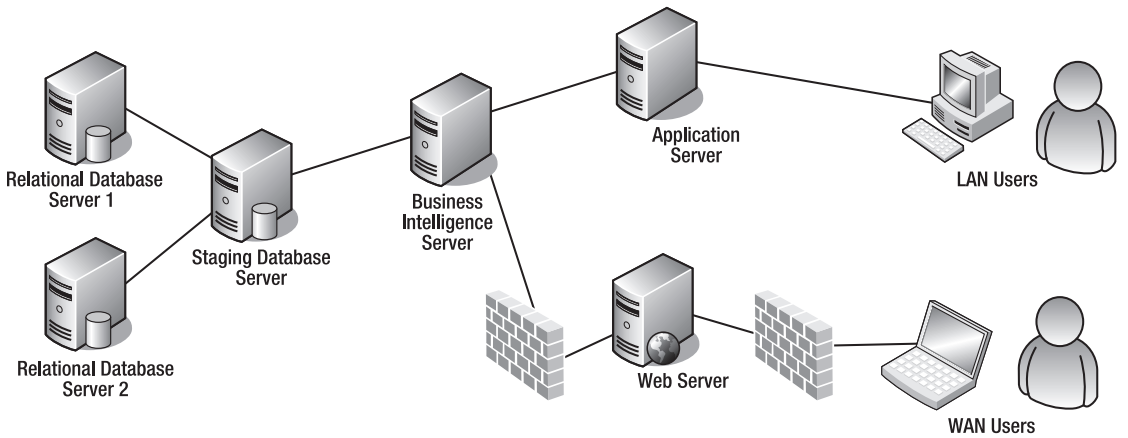


Figure 1-1. *An enterprise BI configuration*

In addition to the term *business intelligence*, there are several other terms commonly used in discussing the technologies depicted in Figure 1-1:

Data warehouse: A single structure that usually, but not always, consists of one or more cubes. Data warehouses are used to hold an aggregated, or rolled-up and read-only view, of the majority of an organization's data; sometimes this structure includes client query tools.

■ **Tip** Data warehousing is not new. The most often quoted spokespeople from the world of data warehousing theory are Bill Inmon and Ralph Kimball. Both have written many articles and books and have very popular Web sites talking about their experience with data warehousing solutions using products from many vendors.

To read more about Ralph Kimball's ideas on Data Warehouse design modeling, go to `http://www.ralphkimball.com`. I prefer the Kimball approach to modeling (rather than the Inmon approach) and have had good success implementing Kimball's methods in production BI projects.

Data mart: A defined subset of a data warehouse, often a single cube from a group (see Figure 1-2). The single cube represents one business unit (for example, marketing) from a greater whole (that is, the entire company). Data marts were the basic unit of organization in Analysis Services 2000 due to limitations in the product; this is no longer the case for SSAS 2005 (Sequel Server Analysis Services). Now data warehouses consist of usually just one cube.

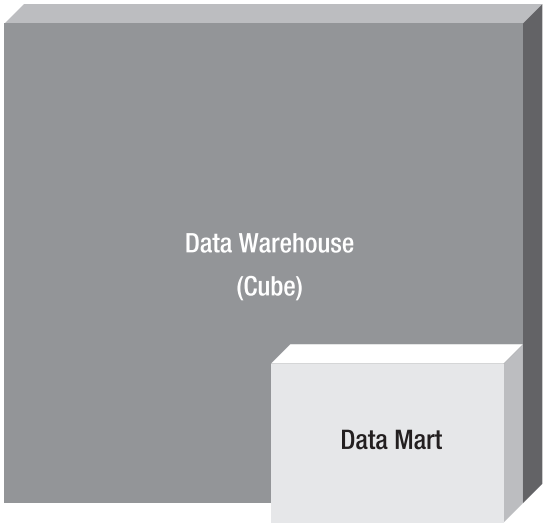


Figure 1-2. *Data marts are subsets of enterprise data (warehouses) and are often defined by time, location, or department.*

Cube: A storage structure used by classic data warehousing products in place of many (often normalized) tables. Rather than using tables with rows and columns, cubes use dimensions and measures (or facts). Also, cubes will usually present data that is aggregated (usually summed), rather than each individual item (or row). This is often stated this way: cubes present a summarized, aggregated view of enterprise data, as opposed to normalized table sources that present detailed data. Cubes are populated with a read-only copy of source data (or production data). In some cases, cubes contain a complete copy of production data; in other cases, cubes contain subsets of source data. The data is moved from source systems to the destination cubes via ETL (Extract, Transform, and Load) processes. We will discuss cube dimensions and facts in greater detail in Chapter 2.

■ **Note** Another name for a cube or set of cubes is an *online analytical processing* (OLAP) system. Some writers actually use the terms *data warehouse*, *cube*, *OLAP*, and *DSS* interchangeably. Another group of terms you'll hear associated with OLAP are MOLAP, HOLAP, and ROLAP. These terms refer to the method of storing the data and metadata associated with a SSAS cube. The acronyms stand for multidimensional OLAP, hybrid OLAP, or relational OLAP. Storage methods are covered in detail in Chapter 7.

Decision Support System (DSS): This term's broad definition can mean anything from a read-only copy of an online transaction processing (OLTP) database to a group of OLAP cubes or even a mixture of both. If the data source consists only of an OLTP database, this store is usually highly normalized. One of the challenges of using an OLTP store as a source for a DSS is the difficulty in writing queries that execute quickly and with little overhead on the source system.

This challenge is due to the level of database normalization. The more normalized the OLTP source, the more joins that must be performed on the query. Executing queries that use many joins places significant overhead on the OLTP store. Also, the locking behavior of OLTP databases is such that large read queries can cause significant contention (or waiting) for resources by end users. Yet another complexity is the need to properly index the tables in each query. This book is focused on using the more efficient BI store (or OLAP cube) as a source for a DSS system.

NORMALIZATION VS. DENORMALIZATION

What's the difference between normalization and denormalization? Although entire books have been written on the topic, the definitions are really quite simple. *Normalization* means reducing duplicate data by using keys or IDs to relate rows of information from one table to another, for example, customers and their orders. *Denormalization* means the opposite, which is deliberately duplicating data in one or more structures. Normalization improves the efficiency of inserting, updating, or deleting data. The fewer places the data has to be updated, the more efficient the update and the greater the data integrity. Denormalization improves the efficiency of reading or selecting data and reduces the number of tables the data engine has to access or the number of calculations it has to perform to provide information.

Defining BI Using Microsoft's Tools

Microsoft entered the BI market when it released OLAP Services with SQL Server 7.0. It was a quiet entry, and Microsoft didn't gain much traction until its second BI product release, SQL Server 2000 Analysis Services.

Since its first market entry, Microsoft has taken the approach that BI should not be for the few (business analysts and possibly executives) but for *everyone* in the organization. This is a key differentiator from the competitor's BI product suites. One implementation of this differentiation is Microsoft's focus on integrating support for SSAS into its Office products—specifically Excel. Excel 2003 can be used as a SSAS client at a much lower cost than third-party client tools. Microsoft has expanded the support for SSAS features in Excel 2007. The tools and products Microsoft has designed to support BI (from the 2000 release onward) have been targeted very broadly. In typical Microsoft fashion, they've attempted to broaden the BI usage base with each release.

The Microsoft vision for BI is ambitious and seems to be correctly positioned to meet market demand. In the first year of release, the market penetration of Microsoft's 2005 toolset for BI grew at double the average BI toolset rate, approximately 26% as compared to the overall BI market rate of growth, which was around 12%.

If you're completely new to BI, it's important for you to consider the possibilities of BI in the widest possible manner when beginning your project. This means planning for the largest possible set of end-user types, that is, analysts, executive managers, middle managers, *and* all

other types of end users in your organization. You must consider (and ask your project supporters and subject matter experts [SMEs]) which types of end-user groups need to see what type of information and in what formats (tabular, chart, and so on).

If you have experience with another vendor's BI product (for example, Cognos, Informatica, or Essbase), you may find yourself rethinking some assumptions based on use of those products because Microsoft's BI tools are not copies of anything already on the market. Although some common functionality exists between Microsoft and non-Microsoft BI tools, there is also a large set of functionality that is either completely new or implemented differently than non-Microsoft BI products. This is a particularly important consideration if you are migrating to Microsoft's BI from a non-Microsoft BI vendor. I've seen several Microsoft BI production solutions that were needlessly delayed due to lack of understanding of this issue. Whether you are migrating or entirely new to BI, you'll need to start by considering the products and technologies that can be used in a Microsoft BI solution.

What Microsoft Products Are Involved?

As of this writing, the most current Microsoft products that support BI are the following:

SQL Server 2005: This is the preferred staging and, possibly, source location for BI solutions. Data can actually be retrieved from a variety of data stores (Oracle, DB2, and so on), so a SQL Server installation is not strictly required to build a Microsoft BI solution. However, due to the integration of some key toolsets that are part of nearly all BI solutions—for example, SSIS or SQL Server Integration Services, which is usually used to perform the ETL of source data into the data warehouse—most BI solutions will include at least one SQL Server 2005 installation. Another key component in many BI solutions is SQL Server Reporting Services (SSRS). When working with SQL Server to perform OLAP administrative tasks, you will use the management interface, which is called SQL Server Management Studio (SSMS).

Sequel Server Analysis Services 2005 (SSAS): This is the core server in Microsoft's BI solution. SSAS provides storage for the data used in cubes for your data warehouse. This product may or may not run on the same physical server as SQL Server 2005. I will detail how to set up cubes in Chapters 4, 5, 6, 7, 10, and 13. Figure 1-3 shows the primary tool—Business Intelligence Development Studio (BIDS) —that you'll use to develop cubes for Analysis Services. You'll note that BIDS opens in a Visual Studio (VS) environment. A full VS installation is *not* required to develop cubes for SSAS. If you do not have VS on your development machine, when you install SSAS, BIDS will install as a stand-alone component. If you do have VS on your development machine, then BIDS will install as a component (really a set of templates) into your existing VS instance.

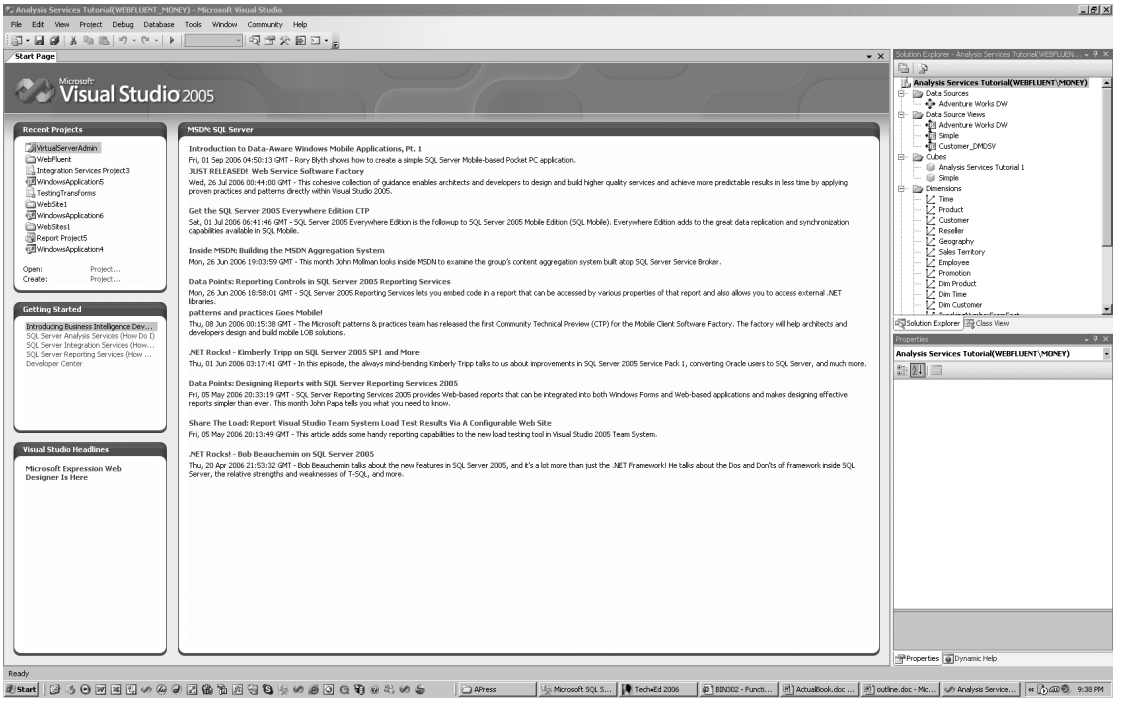


Figure 1-3. *You use the Business Intelligence Development Studio (BIDS) to implement BI solutions.*

Data Mining Using SSAS: This is an optional component included with SSAS that allows you to create data mining structures. These structures include data mining models. *Data mining models* are objects that contain source data (either relational or multidimensional) that have been processed using a particular type of data mining algorithm. These algorithms either classify (group) only or classify and predict one or more column values. Although data mining was available in Analysis Services 2000, Microsoft has significantly enhanced the capabilities of this tool in the 2005 release, for example in the 2000 release there were only two data mining algorithms available, in the 2005 release there are nine algorithms. I will provide an overview of data mining in general, and the capabilities available in SSAS for implementing data mining in Chapter 11.

SQL Server 2005 Integration Services (SSIS): This toolset is a key component in most BI solutions that is used to import, cleanse, and validate data prior to making the data available to the Analysis Services for reporting purposes. It is typical to use data from many disparate sources (relational, flat file, XML, and so on) as source data to a data warehouse. For this reason, a sophisticated toolset, such as SSIS is used to facilitate the complex data loads that are often common to BI solutions. As stated earlier, this functionality is often called ETL (Extract, Transform, and Load) in a BI solution. In SQL Server 2000, the available ETL toolset was named Data Transformation Services (DTS). SSIS has been completed re-architected in this release of SQL Server. Although there is some overlap in functionality, SSIS really is a new release, as compared to DTS, for Microsoft. I will discuss the use of SSIS in Chapters 3, 8, and 9.

SQL Server 2005 Reporting Services (SSRS): This is an optional component for your BI solution. Microsoft has made many significant enhancements in the most current version that makes using SSRS an attractive part of a BI solution. The most important of which is the inclusion of a visual query designer for SSAS cubes, which facilitates rapid report creation by reducing the need to write manual queries against cube data. I will discuss reporting clients, including SSRS, in Chapter 12.

Excel 2003 or 2007: This is another optional component for your BI solution. Many companies already own Office 2003, so use of Excel as a BI client is often attractive for its low cost and (relatively) low training curve. I will compare various client solutions in Chapter 12. Office 2007 is released as of the writing of this book; I will provide a "first look" at new features for Excel 12 (or 2007) in Chapter 14.

■ **Tip** Connecting to an OLAP data source from Excel also requires that MS-Query be installed. MS-Query is listed under optional components on the Office installation DVD.

SharePoint Portal Server 2003 or Microsoft Office SharePoint Server 2007 (MOSS): This is yet another optional component to your BI solution. Most easily used in conjunction with SSRS, using the freely available SSRS Web parts, SharePoint can expand the reach of your BI solution. As mentioned previously, I will detail options using different BI clients in Chapter 12. Office 2007 has a planned release of early spring 2007. SharePoint Services will have many significant enhancements related to BI solutions, which are discussed in Chapter 14.

■ **Note** A Web part is a pluggable UI showing some bit of content. It is installed globally in the SharePoint Portal Server Web site and can be added to a portal page by any user with appropriate permissions.

Visio 2003 or 2007: This is my favorite modeling tool for BI projects. It is optional as well; you can use any tool that you are comfortable using. Sections in Chapter 2 that concern modeling for OLAP include sample Visio diagrams. As with other products in the Office suite, Microsoft has increased the BI integration capabilities with Visio 2007.

ProClarity (acquired by Microsoft in 2006): This is a high-end client tool. Prior to its acquisition, ProClarity was my recommended business analyst tool of choice. ProClarity, as you might imagine, is currently undergoing quite a transition as it becomes part of Microsoft. Microsoft has announced that all ProClarity functionality will be integrated into a new product. This product is called Performance Point Server (PPS). PPS is currently in CTP (Community Technology Preview) release (and set for final release in late 2007). I'll provide an update in Chapter 14.

■ **Note** Microsoft has added significant BI integration into Office 2007—particularly for Excel 2007, SharePoint 2007 (now called Microsoft Office SharePoint Server, or MOSS), and for the renamed Business Scorecards Manager Server (which will be called Performance Point Server). Microsoft has further announced that PPS will include the next release of ProClarity, which means that ProClarity will no longer be available as a stand-alone product.

The capability and feature differences between SSAS editions (standard, enterprise, and so on) for the products in the BI suite are highlighted in Chapter 2, and key feature differences are discussed throughout the entire book. These differences are significant and affect many aspects of your BI solution design, such as the number of servers, number and type of software licenses, and server configuration.

You may be thinking at this point, "Wow, that's a big list. Am I required to buy (or upgrade to) all of those Microsoft products to implement a BI solution for my company?" The answer is no, the only server that is *required* is the SSAS. Many companies also provide tools that can be used in a Microsoft BI solution. Although I will occasionally refer to some third-party products, I will primarily focus on using Microsoft's products and tools to build a BI solution in this book.

BI Languages

An additional consideration is that you will use at least three languages when working with SSAS. The first, which is the primary query language for cubes, is *not* the same language used to work with SQL Server data (T-SQL). The query language for SSAS is called MDX. SSAS also includes the capability to build data mining structures. To query the data in these structures, you'll use yet another language—DMX. Finally, Microsoft introduces an administrative scripting language in SSAS 2005—XMLA. Here's a brief description of each language.

MDX (Multidimensional Expressions): This is the language used to query OLAP cubes. Although this language is officially an open standard, and some vendors outside of Microsoft have chosen to adopt parts of it into their BI products, the reality is that very few developers are proficient in MDX. A mitigating factor is that the need for you to manually write MDX in a BI solution can be relatively small—not nearly as much T-SQL as you would manually write for a typical OLTP database. However, retaining developers who have at least a basic knowledge of MDX is an important consideration in planning a BI project. MDX is introduced in Chapter 10.

Figure 1-4 shows a simple example of an MDX query in SQL Server Management Studio (SSMS).

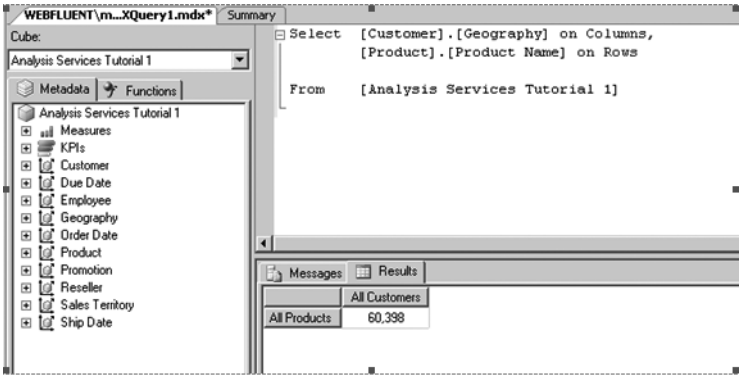


Figure 1-4. *The MDX query language is used to retrieve data from SSAS cubes. Although MDX has a SQL-like structure, MDX is far more difficult to master. This is due to the complexity of the SSAS source data structures—cubes.*

DMX (Data Mining Extensions): This is the language used to query data mining structures (which contain data mining models). Although this language is officially an open standard, and some vendors outside of Microsoft have chosen to adopt parts of it into their BI products, the reality is that very few developers are proficient in DMX. A mitigating factor is that the need for DMX in a BI solution is relatively small (again, not nearly as much T-SQL as you would manually write for a typical OLTP database). Also, Microsoft's data mining interface is heavily wizard driven, more than creating cubes (which is saying something!). However, retaining developers who have at least a basic knowledge of DMX is an important consideration in planning a BI project that will include a large amount of data mining. DMX is introduced briefly in Chapter 11.

XMLA (XML for Analysis): This is the language used to perform administrative tasks in SSAS. Here are some examples of XMLA tasks: viewing metadata, copying, backing up databases, and so on. Although this language is officially an open standard, and some vendors outside of Microsoft have chosen to adopt parts of it into their BI products, the reality is that very few developers are proficient in XMLA. A mitigating factor is that Microsoft has made generating XMLA scripts simple. In SSMS, when connected to SSAS, you can right-click any SSAS object and generate XMLA scripts using the GUI interface. XMLA is introduced in Chapter 13.

Because I've covered so many acronyms is this section, and I'll be referring to these products by their acronym going forward in this book, a quick list is provided in Figure 1-5.

Acronym	Name
BI	Business Intelligence
BIDS	Business Intelligence Development Studio
BSM	Business Scorecards Manager Server
DMX	Data Mining Extensions
DSS	Decision Support System
HOLAP	Hybrid OLAP
MDX	MultiDimensional Expressions
MOLAP	Multidimensional OLAP
MOSS	Microsoft Office SharePoint Server
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
PPS	Performance Point Server
ROLAP	Relational OLAP
SPS	SharePoint Portal Server
SSAS	SQL Server Analysis Services
SSIS	SQL Server Integration Services
SSMS	SQL Server Management Studio
SSRS	SQL Server Reporting Services
VS	Visual Studio
XMLA	XML for Analysis Services

Figure 1-5. *For your convenience, the various BI acronyms used in this book are listed here.*

Understanding BI from an End User's Perspective

You may be wondering where to start at this point. Your starting point depends on the extent of involvement you and your company have had with BI technologies. Usually you will either (a) be completely new to BI; (b) be new to SSAS 2005, that is, you are using SSAS 2000; or (c) be new to Microsoft's BI, that is, you are using another vendor's products to support BI. If BI is new to you and your company, then a great place to start is with the end user's perspective of a BI solution. To do this, you will use the simplest possible client tool for SSAS—an Excel pivot table. This is a great way to familiarize not only yourself, but also other members of your team and your executive sponsors about basic BI concepts.

■ **Note** If you have experience with basic BI end-user tools (particularly pivot tables), you may want to skip to the next chapter.

Demonstrating the Power of BI Using Excel 2003 Pivot Tables

Although this may seem like a strange way to showcase a suite of products that is as powerful as Microsoft's BI toolset, my experience has shown over and over that this simple approach is quite powerful.

There are two ways to implement the initial setup. Which you choose will depend on the amount of time you have to prepare and the sophistication level of your audience. The first approach is to create a cube using the sample database (AdventureWorksDW) that Microsoft

provides with SSAS. Detailed steps for using the first approach are provided later in this chapter. The second approach is to take a very small subset of data from your company and to use it for a demonstration or personal study. If you want to use your own data, you'll probably have to read a bit more of this book to be able to set up a basic cube using your own data.

The rest of this chapter will get you up and running with the included sample. At this point, we are going to focus simply on clicks, that is "click here to do this." We are not yet focusing on the "why" at this point. The rest of the chapters will explain in detail just what all this clicking actually does and why you click where you're clicking.

Building the First Sample—Using AdventureWorksDW

To use the SQL Server 2005 AdventureWorksDW sample database as the basis for building a SSAS cube, you'll need to have at least one machine with SQL Server 2005 and SSAS installed on it. While installing, make note of the edition of SQL Server that you are using (you can use the Developer, Standard, or Enterprise editions) because you'll need to know the particular edition when you install the sample cube files.

If you're installing SQL Server, remember to choose the option to install the sample databases. This option is *not* selected by default. If SQL Server is already installed, you can download (and install) the sample database AdventureWorksDW. You will use AdventureWorksDW rather than AdventureWorks as the source database for your first SSAS OLAP cube because the former is modeled in a way that is most conducive to easy cube creation. Chapter 2 details what modeling for SSAS cubes consists of and how you can apply these modeling techniques to your own data.

■ **Tip** The AdventureWorksDW database comes on the source media for SQL Server 2005. To install it, you can either rerun setup, or, if you don't have access to the source media, you can download the sample database from `http://www.microsoft.com/downloads/details.aspx?FamilyID=E719ECF7-9F46-4312-AF89-6AD8702E4E6E&displaylang=en`. This URL includes detailed instructions for installing this sample database after you have downloaded it.

To create the sample cube, you will use the sample AdventureWorks Analysis Services project. The sample consists of a set of physical files that contains metadata that SSAS uses to structure the sample Adventure Works cube. As mentioned earlier, you'll work with these sample files in BIDS. The sample is available in the Standard Edition and the Enterprise Edition. You will select the sample file from the directory that matches the edition that you have installed. There are significant feature differences between the two editions, which you will learn about in detail as you work through the available features in this book.

■ **Note** The Developer Edition has an identical feature set to the Enterprise Edition (for the purposes of your development, demonstration, or personal review). If you have installed the Developer Edition, then select the sample from the Enterprise Edition folder.

How to Deploy the Standard Edition Version of the Sample Cube

To deploy the standard edition of the sample cube:

1. Open the SQL Server Business Intelligence Development Studio (BIDS) from the Start menu.
2. From the BIDS Menu, click File ➤ Open ➤ Project/Solution.
3. Browse to `C:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks Analysis Services Project\Standard`, select the file `Adventure Works DW Standard Edition.sln`, and click Open. This dialog box is shown in Figure 1-6.

Figure 1-6. *To install the SSAS sample cube, select the folder with the edition name that matches the edition of SSAS that you have installed and then double-click AdventureWorks.sln to open the solution in BIDS.*

4. Set the connection string to the server name where you deployed AdventureWorksDW by right-clicking on the Adventure Works.ds data source in Solution Explorer. Click the Edit button on the General tab in the Data Source Designer dialog box to change the connection string. This setting is shown in Figure 1-7.

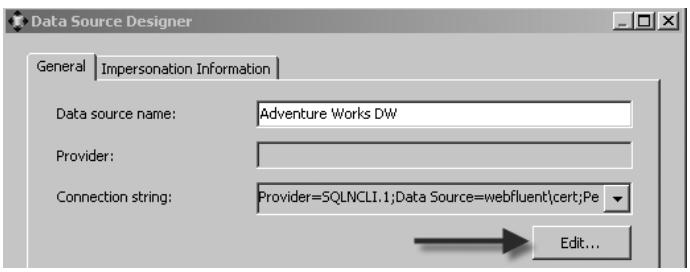


Figure 1-7. *When deploying the sample, be sure to verify that the connection string information is correct for your particular installation.*

■ **Note** If you are using the Enterprise Edition, you can follow these steps as well. Simply select the files from the sample Enterprise folder from the path listed next.

Be sure to test the connection as well. You do this by clicking on the Test Connection button on the bottom of the Connection Manager dialog box as shown in Figure 1-8.

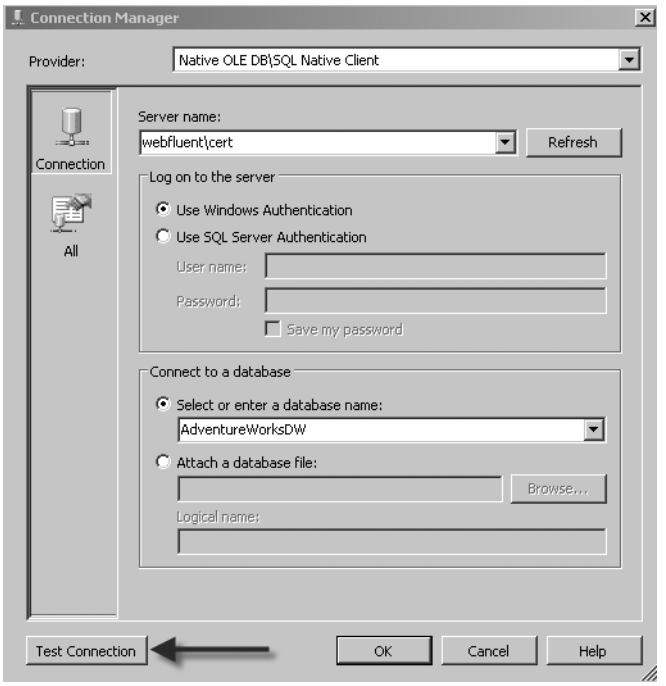


Figure 1-8. *You'll want to test the connection to the sample database, AdventureWorksDW, as you work through setting up the sample SSAS database.*

5. Right-click the name of the project (Adventure Works DW Standard Edition) in Solution Explorer, and then click on Properties from the context menu. You must verify the name of the Analysis Services instance that you intend to deploy the sample project to. The default is localhost. If you are using localhost, then you do not need to change this setting.

   You can also use a named server instance, as shown in Figure 1-9. In that case, in the project's Properties Pages dialog box, click on Deployment, and set the target sever name to the computer name and instance name separated by a backslash character where you have deployed SSAS (see Figure 1-9).

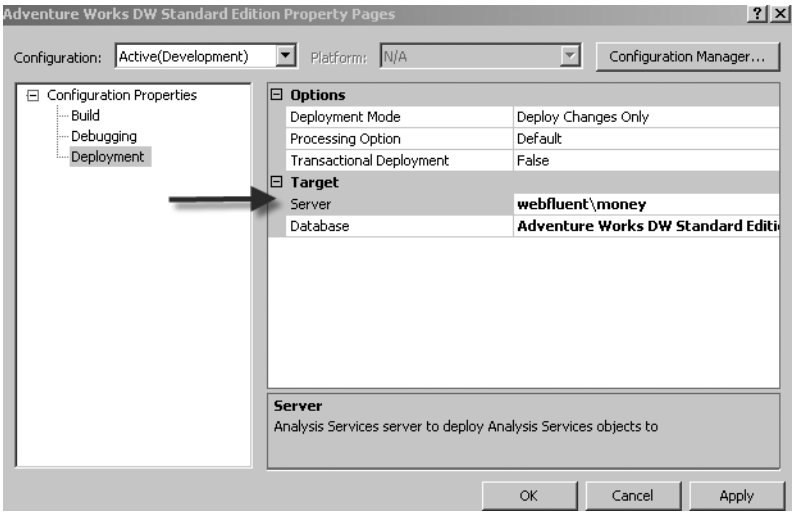


Figure 1-9. *Before deploying the sample SSAS project, right-click the solution name in BIDS, and then click Properties. In the properties sheet, verify the SSAS instance name.*

6. From Solution Explorer, right-click the Adventure Works DW Standard Edition project name, and then click on Deploy. This will process the cube metadata locally and then deploy those files to the Analysis Services instance you configured in the previous step.

After clicking deploy, wait for the "deployment succeeded" message to appear at the bottom right of the BIDS window. This can take up to 5 minutes or more depending on the resources available to complete the processing. If the deployment fails (which will be indicated with a large red X in the interface, read the messages in the Process Database dialog box to help you to determine the cause or causes of the failure. The most common error is incorrectly configured connection strings.

Now you are ready to take a look at the sample cube using the built-in browser in BIDS. This browser looks much like a pivot table so that you, as a cube developer, can review your work prior to allowing end users to connect to the cube using client BI tools. Most client tools contain some type of pivot table component, so the included browser in BIDS is a useful tool for you. To view the sample cube using the built-in cube browser in BIDS, perform the following steps:

1. In Solution Explorer, expand the Cubes folder, and then double-click the Adventure Works cube to open the BIDS cube designer work area (see Figure 1-10).

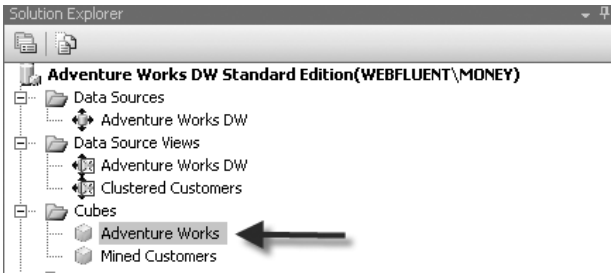


Figure 1-10. *To view the sample cube in BIDS, double-click the cube name in Solution Explorer.*

2. In the cube designer work area (which appears in the center section) of BIDS, on the AdventureWorks main tab, click on the Browser subtab as shown in Figure 1-11.

Figure 1-11. *The cube designer interface has nine tabs. To browse a cube, you click on the Browser tab. The cube must have been successfully deployed to the server to browse it.*

3. Now you can drag and drop items from the cube (dimensions and facts) onto the viewing area. This is very similar to using a pivot table client to view a cube. The functionality is similar, by design, to BI client tools such as Excel pivot tables; however, there are some built-in limitations (for example, on the number of levels of depth you may browse in a dimension), and the Browser tab, like all of BIDS, is designed for cube designers and *not* for end users.

■ **Note** You may be wondering what the dimensions and facts (or measures) are that you see onscreen? We will review these concepts in more detail in Chapter 2, however, as an introduction, you can think of *facts* as important business values (for example daily sales amount or daily sales quantity), and you can think of *dimensions* as attributes (or detailed information) related to the facts (for example, which customers made which purchases, which employees made which sales, and so on).

Spend some time in the BIDS browser interface exploring; drag and drop different items onto the display surface and around the display surface. Also, try right-clicking on the design surface to find many interesting built-in options to display the information differently.

You can use Figure 1-12 as a starting point. The Order Count measure is displayed in the data area, the Calendar Year hierarchy from the Date dimensions is displayed on the columns axis, the Country hierarchy from the Geography dimension is displayed on rows, the Employee Department attribute from the Employees dimension is dis-

played as a filter, and the Product Model Categories hierarchy from the Product dimension is set to filter the browser results to include only measure values where the Product Model Category is equal to Bikes.

■ **Tip** To remove any measures or dimensions from the browse area, click on the item you want to remove and drag it back over the tree listing of available objects.

Figure 1-12 is a view of the sample Adventure Works cube. Note that you can place dimension members and hierarchies on the rows, columns, or filter axis and that you can view measures in the area labeled Drop Total or Detail Fields Here.

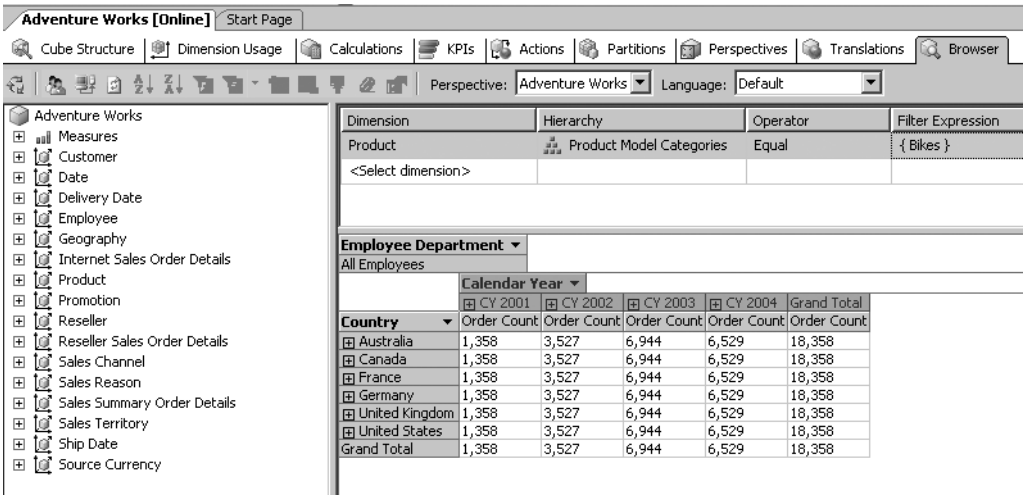


Figure 1-12. *The BIDS cube browser uses a pivot table interface to allow you to view the cube that you have built (or, in this case, simply deployed) using the BIDS cube designer.*

■ **Note** If you are wondering whether you can view sample data mining models in BIDS, the answer is yes. The AdventureWorks samples include data mining structures. Each structure contains one or more data mining models. Each mining model has one or more viewers available in BIDS. Data mining is a deep topic, so I'll spend all of Chapter 11 discussing the mining model types and BIDS interfaces. Also, Excel 2003 does *not* support the display of SSAS mining structures. Excel 2007, however, does, so I'll discuss these features in Chapter 14.

How to Connect to the Sample Cube Using Excel 2003

Now that you've set up and deployed the sample cubes, you will probably want to experience an end user's perspective. An easy way to do this is with a pivot table in Excel 2003:

1. Open Excel 2003.
2. Select Data ➤ Pivot Table.
3. On the PivotTable Wizard Step 1, select Connect to External Data Source.
4. On the PivotTable Wizard Step 2, click the Get Data button as shown in Figure 1-13.

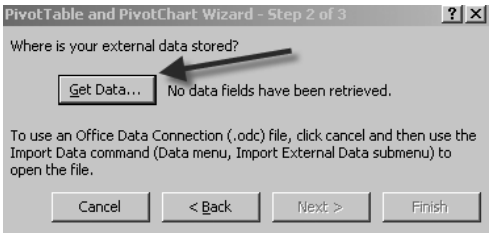


Figure 1-13. *When connecting to a SSAS cube in Excel, you must configure the connection to the SSAS server by clicking on the Get Data button on Step 2 of the PivotTable wizard.*

5. In the Choose Data dialog box, select the OLAP Cubes tab, and then select <new>.
6. In the Create New Data Source dialog box, name your connection, select Microsoft OLE DB Provider for Analysis Services 9.0 in the Select an OLAP provider for the database you want to access box, and then click the Connect button (see Figure 1-14).

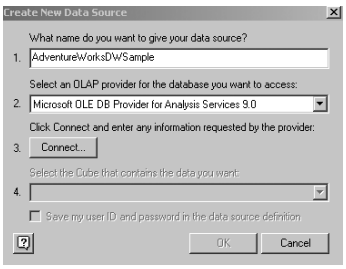


Figure 1-14. *When you are configuring your connection to the SSAS cube, be sure to select the OLE DB Provider for Analysis Services 9.0.*

7. In the first Multidimensional Connection 9.0 dialog box, enter the instance name of the Analysis Services where you deployed the sample project, and then click Next.
8. In the second Multidimensional Connection 9.0 dialog box, click on the name of your sample project (Adventure Works DW Standard [or Enterprise] Edition) in the list of databases to select it. Click Finish. You are returned from the MS Query dialog boxes back to the Create New Data Source dialog box (shown in the previous figure).

9. In this dialog box, click on the 4. Select the Cube that contains the data you want drop-down list box, select AdventureWorks, and click OK. This will return you to the Choose Data Source dialog box. Click OK.

10. You are now returned to the PivotTable Wizard Step 2. Click Next to advance to Step 3. On the Step 3 dialog box, click the Layout button as shown in Figure 1-15.

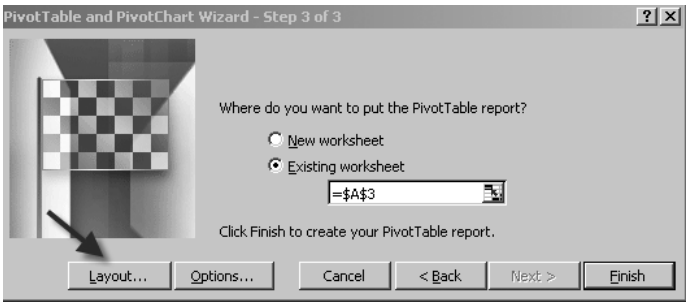


Figure 1-15. *In Step 3 of the PivotTable wizard, you'll click on the Layout button to display the area to drag and drop your dimensions or measures onto the pivot table layout surface.*

11. On the PivotTable Wizard layout, drag the items that you want to show on the rows, columns, and center area. Figure 1-16 shows a sample. The dimensions are listed first in the list of items, and the measures are listed at the end. It is a bit difficult to read the dimension and measure names in this page of the wizard because the fixed button size truncates the dimension and measure names. If you try to drag an item to a layout area where it cannot be displayed (for example, drag a measure to the column area), then the Layout wizard will not allow you to drop that item. The dialog box provides visual hints to help you lay out your pivot table correctly.

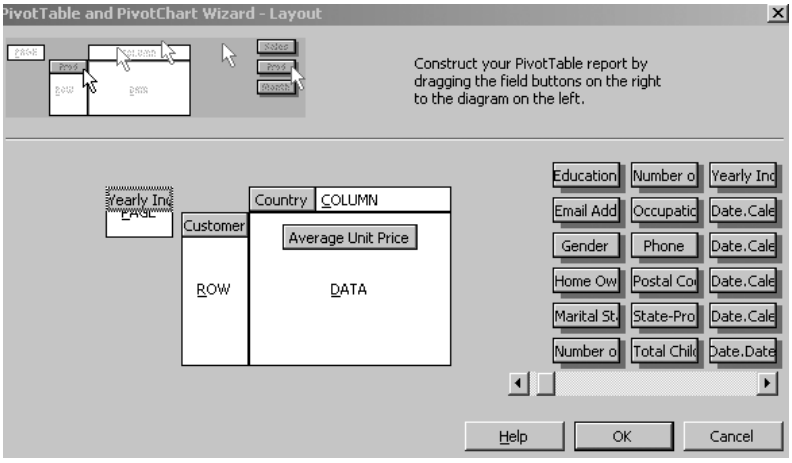
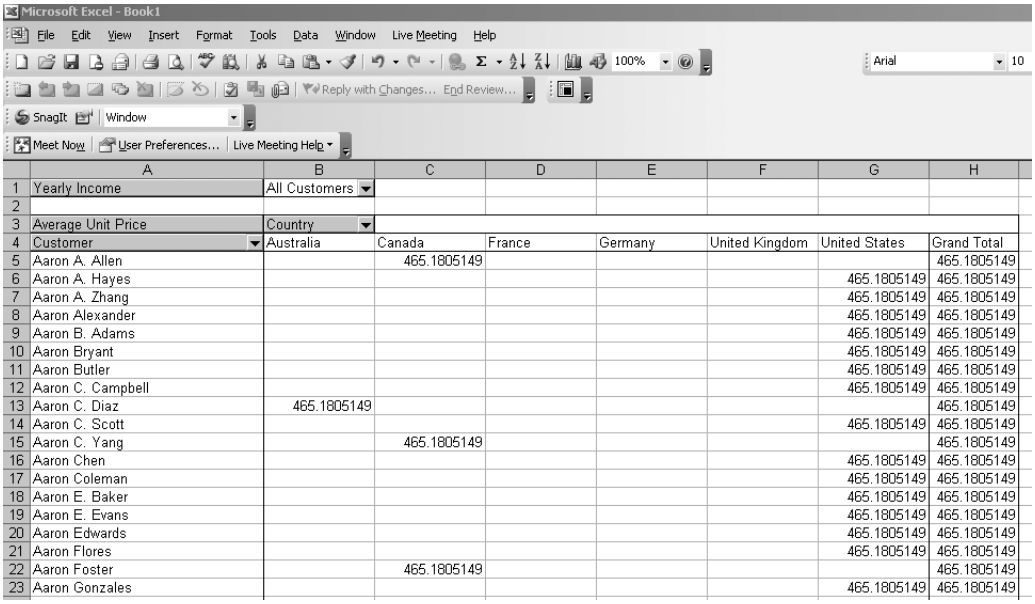


Figure 1-16. *Using the Layout dialog box, you drag and drop dimensions and measures onto the layout area. Drag only measures to the DATA area.*

12. Click OK and Finish. Your pivot table will look somewhat similar to Figure 1-17. If you want to remove items, simply drag the (grey) headers out of the pivot table area. The cursor will change to a red X when the item can be removed from the pivot table. If you want to add items, display the pivot table toolbar (View ➤ Toolbars), and click the last button to show the pivot table field list on the screen. When that list is visible, you can drag items to the pivot table to make their values visible.



	A	B	C	D	E	F	G	H
1	Yearly Income	All Customers						
2								
3	Average Unit Price	Country						
4	Customer	Australia	Canada	France	Germany	United Kingdom	United States	Grand Total
5	Aaron A. Allen		465.1805149					465.1805149
6	Aaron A. Hayes						465.1805149	465.1805149
7	Aaron A. Zhang						465.1805149	465.1805149
8	Aaron Alexander						465.1805149	465.1805149
9	Aaron B. Adams						465.1805149	465.1805149
10	Aaron Bryant						465.1805149	465.1805149
11	Aaron Butler						465.1805149	465.1805149
12	Aaron C. Campbell						465.1805149	465.1805149
13	Aaron C. Diaz	465.1805149						465.1805149
14	Aaron C. Scott						465.1805149	465.1805149
15	Aaron C. Yang		465.1805149					465.1805149
16	Aaron Chen						465.1805149	465.1805149
17	Aaron Coleman						465.1805149	465.1805149
18	Aaron E. Baker						465.1805149	465.1805149
19	Aaron E. Evans						465.1805149	465.1805149
20	Aaron Edwards						465.1805149	465.1805149
21	Aaron Flores						465.1805149	465.1805149
22	Aaron Foster		465.1805149					465.1805149
23	Aaron Gonzales						465.1805149	465.1805149

Figure 1-17. *After you've completed configuring the connection to your SSAS sample cube using the PivotTable wizard in Excel, the result appears to the end user as a regular pivot table.*

■ **Tip** Would you like to practice a bit more with pivot tables? Microsoft has a site where you can download and work with 25 different sample pivot tables: `http://office.microsoft.com/en-us/assistance/HA010346331033.aspx`.

You may also want to create a pivot chart. Some people simply prefer to get information via graphs or charts rather than rows and columns of numbers. As you begin to design your BI solution, it is very important to consider the needs of all the different types of users of your solution. To create a pivot chart, simply display the pivot table toolbar and click on the Chart Wizard button. Figure 1-18 is a sample of a pivot chart.

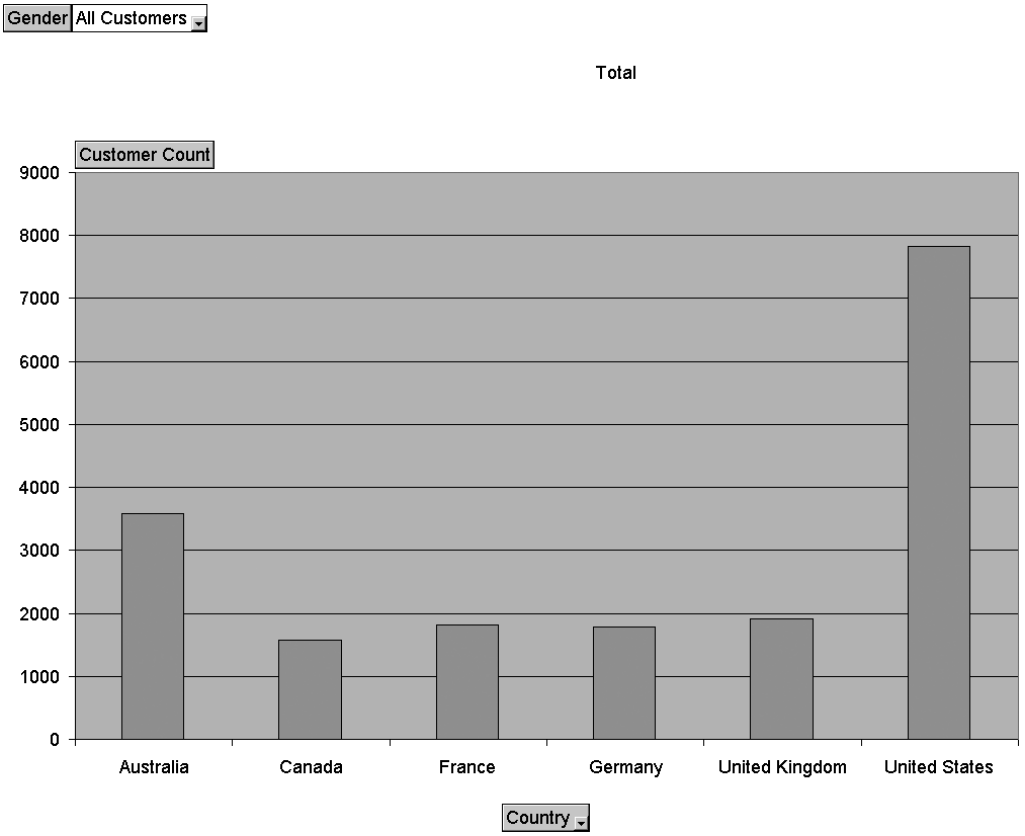


Figure 1-18. *The method used to create a pivot chart using SSAS cube data is similar to that used when creating a pivot table.*

Understanding BI Through the Sample

Now that your pivot table is set up, what exactly are you trying to understand by working with it? How is a pivot table that gets its data from a SSAS cube different from any other Excel pivot table? Here is a list of some of the most important BI (or OLAP) concepts:

- BI is *comprehensive* and flexible. A single, correctly designed cube can actually contain all of an organization's data, and importantly, this cube will present that data to end users consistently. To better understand this concept, you should try working with the AdventureWorksDW sample cube as displayed using the Excel pivot table to see that multiple types of measures (both Internet and Retail Sales) have been combined into one structure.

  Most dimensions apply to both groups of measures, but not all do. For example, there is no relationship between the Employee dimensions and any of the measures in the Internet Sales group because there are no employees involved in these types of sales. Cube modeling is now flexible enough to allow you to reflect business reality in a single cube.

In previous versions of SSAS and in other vendor's products, you would've been forced to make compromises such as creating multiple cubes or being limited by structural requirements. This lack of flexibility in the past often translated into limitation and complexity in the client tools as well.

- BI is *accessible* (intuitive for all end users to view and manipulate). To better understand this aspect of BI, try demonstrating the pivot table based on the SSAS sample cube to others in your organization. They will usually quickly understand and be impressed (some will even get excited!) as they begin to see the potential reach for BI solutions in your company.

  Pivot table interfaces reflect the way many users think about data, which is "what are the measures (or numbers) and what attributes (or factors) created these numbers?"

  Some users may request a simpler interface than a pivot table (that is, a type of "canned report"). Microsoft provides client tools, such as SSRS, which facilitate that type of implementation. It is important for you to balance this type of request, which entails manual report writing by you, versus the benefits available to end users who can use pivot tables. In my experience, most BI solutions include a pivot table training component for those end users who haven't worked much with pivot tables before.

- BI is *fast to query*. After the initial setup is done, queries can easily run 1000% faster in an OLAP database than in an OLTP database. Your sample won't necessarily demonstrate the speed of query in and of itself. However, it is helpful to understand that the SSAS server is highly optimized to provide a far superior query experience (than to provide a typical relational database) because the SSAS engine itself is actually designed to quickly fetch or calculate aggregated values. We will dive into the details on this topic in Chapter 7 of this book.

- BI is *simple to query***.** End users simply drag items into and around the pivot area; developers write very little query code manually. It is important to understand that SSAS clients (like Excel) automatically generate MDX queries when users drag and drop dimensions and measures onto the design surfaces. This is a tremendous advantage as compared to traditional OLTP reporting solutions where T-SQL developers must manually write all of the queries.

- BI provides accurate, near real-time, *summarized* information. This will improve the quality of business decisions. Also with some of the new features available in SSAS, most particularly Proactive Caching, cubes can have latency that is only a number of minutes or even seconds. We'll discuss configuring real-time cubes in Chapter 7.

  Also, using drilldown, users who need to see the detail (that is, the numbers behind the numbers) can do so. *Drilldown* is, of course, implemented in pivot tables via the simple "+" interface that is available for all (summed) aggregations in the AdventureWorksDW sample cube.

- BI *improves ROI* by allowing more end users to make more efficient use of enterprise information so many companies have all the information they need. The problem is that the information is not accessible in formats that are useful for the people in the company to use as a basis for decision making in a timely way.

It's really just that simple; OLAP (or BI) solutions simply give businesses a significant competitive advantage by making more data available to more end users so that those users can make better decisions in a more timely way. What's so exciting about BI is that Microsoft has made it possible for many companies who couldn't previously afford to implement any type of BI solution to be able to "play" in this space by including all of the core BI tools and technologies needed to implement cubes in the box with SQL Server. As previously stated, it is important to understand which features require the Enterprise Edition of SQL Server or Analysis Services. We will review feature difference by edition in detail throughout this book.

In addition to broadening BI's reach by including some BI features in both the Standard and Enterprise Editions of SQL Server, Microsoft is also providing some much needed competition at the enterprise level by including some extremely powerful BI features in the Enterprise Editions of SQL Server and SSAS. I'll talk more about that at the end of this chapter.

Understanding the Business Problems that BI Addresses

As you learn more about SSAS capabilities, you can begin to match some of the strengths of the BI toolset available in SQL Server 2005 (and companion BI Microsoft products) to current challenges you and your company may be facing when working with your enterprise data. I call these challenges "pain points" and list some OLAP (or BI) solutions to commonly seen challenges in Table 1-1.

Table 1-1. *List of Business Pain Points and OLAP Solutions*

Pain Point	OLAP Solutions
Slow-to-execute queries	Use cubes that are optimized for read-only queries and can be 1000% faster to return query results than OLTP systems due to the efficiency of the SSAS engine to aggregate data.
General system slowdowns	Greatly reduce locking overhead from OLTP source systems. (OLAP systems do not use locks, except during processing.)
Manual query writing	Allow end users to click to query (click and drag on pivot table), which eliminates the wait time associated with traditional T-SQL, where end users must usually request reports, which results in developers manually writing queries against OLTP systems.
Disparate data sources	Combine data into central repositories or cubes using ETL packages that can be automated to run on a regular basis.
Invalid/inconsistent report data	Based on data that has been cleaned and validated (prior to cube load) using the ETL toolset available in SSIS. Cubes provide a consistent and unified view of enterprise data all across the enterprise.
Data is not available to all users	Designed to be accessed by all business users.
Too much data	Can use data mining (along with the other tools, that is, cubes, available in SSAS) to find patterns in large amounts of data automatically. SSAS now contains nine different types of data mining (called algorithms) to help you group, correlate, and predict data values.

Reasons to Switch to Microsoft's BI Tools

In addition to providing a great suite of tools for companies that are just getting started with BI, Microsoft's 2005 release of the BI tools has also targeted companies that are using other vendor's BI products. Microsoft has done this by providing a raft of enterprise features in its data warehousing products. Many of these features are available only in the Enterprise edition of the various BI products, that is, Analysis Services, SQL Server Integration Services, and SQL Reporting Services.

■ **Tip** The comprehensive feature comparison list for each edition of SQL Server 2005 is available at `http://www.microsoft.com/sql/prodinfo/features/compare-features.mspx`.

Here's a list of BI-specific features that require the Enterprise edition of Analysis Services 2005:

- Advanced Business Analytics
- Proactive Caching
- Advanced Data Management
- Writeback
- Advanced Data Mining Tuning
- Advanced SSIS Transforms
- Text Mining Support

These features will be reviewed in more detail in the subsequent chapters of this book.

Also, Microsoft has built its BI tools so that they will integrate with other vendor's products. It is quite common, for example, to use SSAS to create cubes from Oracle or DB2 data sources. Another example is to use SSRS with a mainframe or an Informix source data. Microsoft is aggressively adding support for interoperability across the entire suite of BI tools. Interestingly, the number one BI vendor for Oracle is no longer Oracle, even though Oracle does provides a rich set of BI tools. Due to lower cost, ease of implementation, and a comparable feature set, SSAS now takes first place as an Oracle-sourced BI solution provider.

Another compelling aspect of Microsoft's BI offering is the inclusion of intelligent wizards and GUI tools that allow you to get up and running quickly and easily. The catch, however, is that the use of these tools and wizards is heavily dependent on your understanding and implementation of basic OLAP modeling concepts. We will look at that topic in the next chapter.

Summary

This chapter covered basic data warehousing terms and concepts, including OLAP, BI, dimension, and fact. We then reviewed the process and procedures you use to quickly set up a sample SSAS cube using BIDS. You worked with the AdventureWorksDW sample, connecting to it with an Excel 2003 pivot table, to give you a quick view of an OLAP solution from an end user's perspective.

In the next chapter, we'll dig deeper into OLAP concepts and learn about basic modeling for cubes.