Irina Gorbach
Alexander Berger
Edward Melomed

Microsoft® SQL Server™ 2008 Analysis Services

UNLEASHED

SAMS

Microsoft® SQL Server™ 2008 Analysis Services Unleashed

ISBN-13: 978-0-672-33001-8
ISBN-10: 0-672-33001-6

Library of Congress Cataloging-in-Publication Data:

Melomed, Edward.
Microsoft SQL server 2008 analysis services unleashed / Edward Melomed, Alexander Berger, Irina Gorbach.
p. cm.
ISBN 978-0-672-33001-8
1. SQL server. 2. Client/server computing. 3. Relational databases.
I. Berger, Alexander. II. Gorbach, Irina. III. Title.
QA76.9.C55M483 2008
005.75'65–dc22
2008049303

Printed in the United States of America

First Printing December 2008

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Bulk Sales

Pearson offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact:

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales
+1-317-581-3793
international@pearsontechgroup.com

Editor-in-Chief
Karen Gettman

Executive Editor
Neil Rowe

Acquisitions Editor
Brook Farling

Development Editor
Mark Renfrow

Managing Editor
Patrick Kanouse

Project Editor
Jennifer Gallant

Copy Editor
Keith Cline

Indexer
Ken Johnson

Proofreader
Water Crest Publishing, Inc.

Publishing Coordinator
Cindy Teeters

Book Designer
Gary Adair

Compositor
Mark Shirar

Graphics
Laura Robbins

Foreword

I am pleased to see this book being updated for a second edition, to cover the changes in Analysis Services 2008, and also to clarify some of the more difficult material in the first edition. This should make the book even more useful to its target users.

Now that Analysis Services is in its fourth major release, it has become a big, complex product, far removed from the relatively simple first release of a mere decade earlier. To make the most of it, model designers need much more knowledge than is available in the online documentation, which makes books like this all the more necessary. And, of course, now that the product is so widely used, sometimes for quite challenging applications, there is much more experience of the best practices to follow. Some of these are now baked into the product itself, but books like this can provide much more context for their use.

The authors are to be highly commended for putting in the effort to comprehensively update a substantial work like this; I know from my own experience how much extra motivation you need to update an existing publication after just two years, compared to the excitement of creating the first edition. All too often, publications like this remain frozen when new versions of the software they describe are released, leaving users to guess which parts remain true, and which have been superseded. In this case, this second edition actually follows more closely on the heels of Analysis Services 2008 than did the first edition on Analysis Services 2005.

Microsoft is also to be commended for continuing to permit or even encourage the disclosure of this level of detail about one of its major products; with the consolidation of the BI industry, some of the other major vendors have become much less willing to provide detailed information about the inner workings of their products. In any case, I have never known any other OLAP server vendor to be so open.

Users of Analysis Services are fortunate in the range of books available to them: more than for all the other OLAP servers combined. This is clearly the book for the most technical users who really need and want to understand exactly how Analysis Services works.

There are many other books for those just getting started with Analysis Services, or who want a clear 'how do I?' guide. The many application developers who just want to improve their Analysis Services skills will probably find this book overwhelming; there are at least a dozen simpler books to choose from. And, needless to say, this book is definitely not aimed at business users who want to understand what Analysis Services can do for them.

Nigel Pendse
Editor of *The OLAP Report*
Author of *The OLAP Survey*

Introduction

Analysis Services began as the project of a small Israeli firm named Panorama, which had responded to a request from a British publishing company to develop an application that would analyze the data stored in its relational database. By the end of 1994, Panorama developers began work on a more general application that would make it possible for business managers to analyze data with relative ease.

With its first release in 1995, Panorama deployed the application to several dozen customers. As the next release moved the application more deeply into the Israeli market, the Panorama team began to develop a new client/server analytical application. The server would process the data and store it in a proprietary format, and the client would also offer users an easy-to-use, rich graphical interface.

By 1996, the application had come to the attention of Microsoft, which acquired the technology by the end of that same year. In early 1997, a small Panorama team comprised of Alexander Berger, Amir and Ariel Netz, Edward Melomed, and Mosha Pasumansky moved from Tel Aviv to Redmond to start work on the first version of Microsoft OLAP Server. After the move to the United States, the team added new developers Irina Gorbach and Py Bateman.

To make the application attractive to enterprise customers, the team took on the challenge of formalizing and standardizing data exchange protocols, and they eliminated the client side of the application in favor of supporting a variety of third-party client applications. In early 1997, a small group including Alexander Berger retreated to a Puget Sound island to brainstorm the foundation of what would become SQL Server Analysis Services.

That retreat produced a plan for developing a standard protocol for client applications to access OLAP data: OLEDB for OLAP. More important, and more challenging, was the plan for developing a new query language that could access multidimensional data stored in the OLAP server—MDX (Multidimensional Expressions). MDX is a text language similar to SQL. MDX makes it possible to work with a multidimensional dataset returned from a multidimensional cube. From its inception, MDX has continued to change and improve, and now it is the de facto standard for the industry.

The original release plan was to include the OLAP server in the 1997 release of SQL Server 6.5. However, instead of rushing to market, Microsoft decided to give the development team more time to implement MDX and a new OLEDB for OLAP provider. Microsoft's first version of a multidimensional database was released in 1998 as part of SQL Server 7.0. That version was integrated with Microsoft Excel PivotTables, the first client for the new server.

Under the slogan, "multidimensionality for the masses," this new multidimensional database from Microsoft opened the market for multidimensional applications to companies of all sizes. The new language and interface were greeted favorably. The simplicity (and, one could say, elegance) of the design made it possible for users to rapidly become proficient with the new product, including users who weren't database experts. Technology that used to be available only to large corporations was now accessible to medium-sized and small businesses. As a result, the market for new applications that use multidimensional analysis has expanded and flourished in an environment rich with developers who write those applications.

But, of course, we were not satisfied to rest on our laurels. We took on a new goal—turn Analysis Services into a new platform for data warehousing. To achieve this, we introduced new types of dimensions, increased the volume of data the server can process, and extended the calculation model to be more robust and flexible. Even though no additional personnel joined the team for this effort, by the end of 1999 we brought the new and improved Analysis Services 2000 to market.

For the next five years, more and more companies adopted Analysis Services until it became a leader in the multidimensional database market, garnering a 27% market share. Now, multidimensional databases running on OLAP servers are integral to the IT infrastructures of companies of all sizes. In response to this wide adoption of multidimensional database technology, Microsoft has increased the size of the team devoted to OLAP technology in order to continue to develop the platform to meet the requirements of enterprise customers.

For the 2005 release of SQL Server Analysis Services, we started from ground up, rewriting the original (and now aging) code base. We built enterprise infrastructure into the core of the server.

SQL Server 2008 release continues to improve architecture and functionality of Analysis Services. While improving the performance of query execution, it also introduces query language extensions and new management capabilities.

Who Is This Book's Intended Audience?

In this book, we bring you the tools you need to fully exploit Analysis Services and explain the architecture of the system. You'll find all of the coverage of our previous book (just in case you were wondering if you needed to go back and read that one first), including the basic architecture established in Analysis Services 2005, as well as all the improvements introduced in Analysis Services 2008. *Analysis Services Unleashed* gives you a full understanding of multidimensional analysis and the MDX query language. It also exposes all the aspects of designing multidimensional applications and management of the system.

How This Book Is Organized

The book is divided into the following nine parts:

Parts I and II are devoted to a formalized description of the multidimensional model implemented in the new version of the OLAP server. We give you the vocabulary and concepts you'll need to work with this model.

In Part III, we present a detailed discussion of MDX and explanation of the way we use it to query multidimensional data. You'll need a practical grasp of the data model and MDX to take advantage of all the functionality of Analysis Services.

We devote the middle section of the book in Parts IV–VII to the practical aspects of loading and storing data in Analysis Services, as well as methods of optimizing data preparation and data access. In addition, we examine server architecture.

In the last section of the book, Parts VIII–IX, we discuss data access, the architecture of client components, and data protection. In addition, we examine the practical aspects of administering the server and monitoring its activities.

We wish you great success in your work with Analysis Services 2008, and we hope that our humbly offered book is of service to you.

Conventions Used in This Book

Commands, scripts, and anything related to code are presented in a special `monospace` computer typeface. Bold indicates key terms being defined, and italic is used to indicate variables or for emphasis. Great care has been taken to be consistent in letter case, naming, and structure, with the goal of making command and script examples more readable. In addition, you might find instances in which commands or scripts haven't been fully optimized. This lack of optimization is for your benefit, as it makes those code samples more intelligible and follows the practice of writing code for others to read.

Other standards used throughout this book are as follows:

CAUTION

Cautions alert you to actions that should be avoided.

NOTE

Notes give you additional background information about a topic being discussed.

CHAPTER 1

Introduction to OLAP and Its Role in Business Intelligence

IN THIS CHAPTER

- The Multidimensional Data Model

In the past decade, Microsoft SQL Server Analysis Services established itself as one of the leaders in the Business Intelligences systems market. Analysis Services helps managers, employees, customers, and partners to make more informed business decisions by enabling them to analyze information accumulated during a company's day-to-day operations.

Success of Analysis Services and the entire Business Intelligence market was predefined by incredible growth of amounts of data accumulated as a result of everyday functioning of a large number of companies. Today it's hard to imagine a business or an organization that doesn't use an online transaction processing (OLTP) system. OLTP systems provide means to highly efficient execution of a large number of small transactions and reliable access to data stored in the result of the transactions.

The volume of the data stored and processed for one day by an OLTP system could be several gigabytes per day; after a period of time, the total volume of data can reach to the tens and even hundreds of terabytes. Such a large volume of data can be hard to store, but it is a valuable source of information for understanding the way the enterprise functions. This data can prove very helpful for making projections that lead to successful strategic decisions, and for improving everyday decision making.

It's easy to see why analysis of data has become so important to the management of modern enterprises. However, OLTP systems are not well suited to analyzing data. In the past decades, an entire new market has emerged for systems that can provide reliable and fast access for analyzing very large amounts of data: online analytical processing (OLAP).

OLAP enables managers, executives, and analysts to gain insight into data using fast, interactive, and consistent interfaces to a wide variety of possible views of information. For example, with OLAP solution, you can request information about company sales in Europe over the year, then drill down to the sales of computers in September, calculate year-to-date sales or compare revenue figures with those for the same products sold in January, and then see a comparison of TV sets sales in Europe in the same time period.

Because OLAP systems are designed specifically for analysis, they typically don't need to both read and write data. All that is necessary for analysis is reading data. With this emphasis on reading only, OLAP systems enjoy a speed advantage over their OLTP cousins. However, a read-only approach to the database architecture is not the only distinction of the OLAP solution. The following rules distinguish OLAP systems from relational databases:

- **Multidimensional data structures**

  OLAP solutions typically use multidimensional data structures that allow analysts and managers to analyze numeric values from different perspectives, such as time, customers, products, and others.

- **Consistently fast data access**

  Architecture of the system allows constantly fast access to the data. To ensure fast, predictable query times, OLAP solutions typically pre-aggregate data.

- **Intuitive interface**

  Skilled analysts and nontechnical users alike can manipulate and analyze data; they can generate reports without involving their organization's IT department.

- **Complex calculations**

  With multiple dimensions come more complex, cross-dimensional calculations. You might need to calculate the subtotal of sales for the state Washington, for example, to be expressed as a percentage of the whole U.S. sales. Further, this result may be presented as part of a time-series analysis (for instance, current month versus last month, versus a year ago).

The Multidimensional Data Model

The design and development of the multidimensional database—especially Microsoft SQL Server Analysis Services, the system designed and developed by the authors of this book—was inspired by the success of relational databases. If you're already familiar with relational databases, you'll recognize some of the terminology and architecture. But, to understand Analysis Services, you must first understand multidimensional data models, how this model defines the data and processes it, and how the system interacts with other data storing systems, primarily with the relational data model.

The multidimensional data model for Analysis Services consists of three more specific models:

- The conceptual data model
- The application data model
- The physical data model

The Conceptual Data Model

The conceptual data model contains information about how the data is represented and the methods for defining that data. It defines data in terms of the tasks that the business wants to accomplish using the multidimensional database. To define conceptual data model, you use the user specifications for the structure and organization of the data, rules about accessing the data (that is, security rules), and calculation and transformation methods.

In a sense, the conceptual data model serves as a bridge between a business model and the multidimensional data model. The solutions architect is the primary user for the conceptual data model. We use Data Definition Language (DDL) and MDX (Multidimensional Extensions) script for the creation of the conceptual model. You can also use Business Intelligence Development Studio to develop the conceptual data model.

The Application Data Model

The application model defines the data in a format that can be used by the analytical applications that will present data to a user in a way that he can understand and use. The primary user for the application data model is the client application, which exposes the model to the user. The application model is built with the MDX language and XML for Analysis protocol. The chapters of Part 3, “Using MDX to Analyze Data,” contain detailed information about MDX and a few of most commonly used client applications. The chapters of Part 7, “Accessing Data in Analysis Services,” contain information about protocol used by Analysis Services to communicate with client applications.

The Physical Data Model

As in the arena of relational databases, the physical model defines how the data is stored in physical media:

- **Where it is stored**—What drive (or maybe on the network), what types of files the data is stored in, and so on
- **How it is stored**—Compressed or not, how it’s indexed, and so on
- **How the data can be accessed**—Whether it can be cached, where it can be cached, how it is moved into memory, and so on

The database administrator is the primary user for the physical data model. We use XML-based commands for manipulation of data on the physical layer.

Figure 1.1 shows relationships between three parts of multidimensional model.

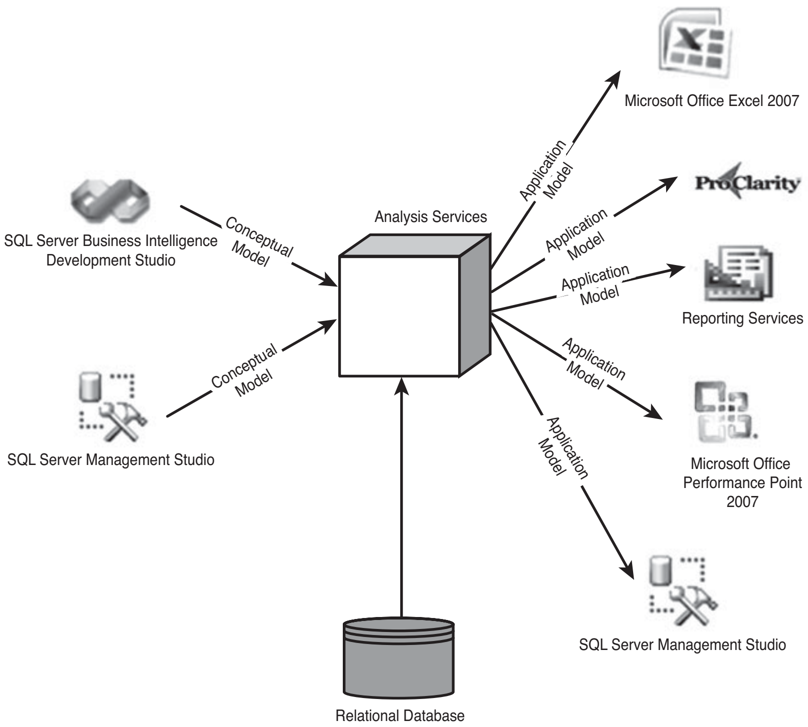

FIGURE 1.1 Submodels of the multidimensional model.

You use SQL Server Business Intelligence Development Studio or SQL Server Management Studio to define a conceptual data model, also known as a Unified Dimensional Model (UDM) or cube. After the conceptual model is defined, you populate it with data by loading/processing the data from the relational database. At this time, you define the physical data model—partitioning scheme of the data, indexing scheme, and so on. The application model of Analysis Services consists of standard data access interfaces. Client applications use those interfaces: XML for Analysis and MDX to communicate with Analysis Services. More than hundred applications available today support the application model of Analysis Services and can work with any Analysis Services cubes.

Unified Dimensional Model

The UDM of Microsoft SQL Server Analysis Services makes it possible for you to set up your system so that different types of client applications can access data from both the relational and the multidimensional databases in your data warehouse, without using separate models for each.

It's been a common industry practice for some time now to build data warehouses that include a relational database for storing data and a multidimensional database for analyzing data. This practice developed because the large volumes of data that multidimensional databases were developed to analyze are typically stored in relational databases. The data would be moved to the multidimensional database for analysis, but relational database would continue to serve as primary storage.

Therefore, it makes sense that the interaction between the stored data and the multidimensional database where it can be analyzed has been an important component of multidimensional database architecture. Our goal for Analysis Services, put simply, is speedy analysis of the most up-to-date data possible.

The speedy and up-to-date parts are what present the challenge. The data in OLTP systems is constantly being updated. But we wouldn't want to pour data directly from an OLTP system into a multidimensional database, because OLTP data is easily polluted by incomplete transactions or incomplete data entered in a transaction. In addition, you don't want your analysis engine to access the OLTP data directly, because that could disrupt work and reduce productivity.

In a data warehouse, OLTP data is typically transformed and stored in a relational database and then loaded into a multidimensional database for analysis. To connect the two databases, you can choose from three methods, each one using a different kind of interaction:

- Relational OLAP (ROLAP), in which no data is stored directly in the multidimensional database. It is loaded from the relational database when it is needed.
- Multidimensional OLAP (MOLAP), in which data is loaded into the multidimensional database and cached there. Future queries are run against the cached data.
- Hybrid OLAP (HOLAP), in which the aggregated data is cached in the multidimensional database. When the need arises for more detailed information, that data is loaded from the relational database.

In earlier versions of Analysis Services, the multidimensional part of the data warehouse was a passive consumer of data from the relational database. The functions of storing data and analyzing data were not only separate, but you had to understand two models—one for accessing a relational database and one for accessing a multidimensional database.

Some client applications would use one model, and others would use the other model. For example, reporting applications traditionally would access the data in a relational database. On the other hand, an analysis application that has to look at the data in many different ways would probably access the data in the multidimensional database, which is designed specifically for that sort of use.

Now, the UDM offers a substantially redefined structure and architecture so that the one model (UDM) serves the purposes of any client application. You no longer have to understand two models; we're providing a unified model. Figure 1.2 shows how many different client applications can use UDM to access data in a variety of different data stores.

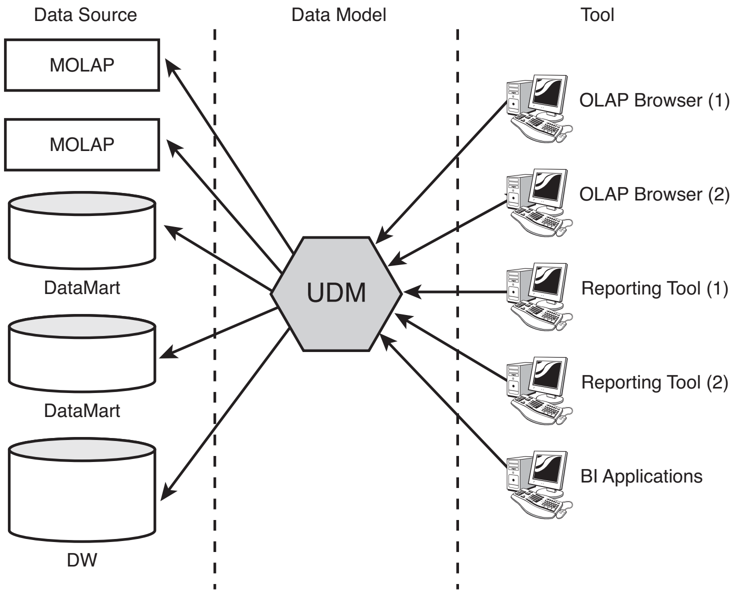

FIGURE 1.2 The UDM provides a unified model for accessing and loading data from varied data sources.

Analysis Services uses *proactive caching* to ensure that the user of the client application is always working with predictable data latency. In essence, proactive caching is a mechanism by which the user can schedule switching from one connection mode (ROLAP, MOLAP, or HOLAP) to another. For example, the user might set his system to switch from MOLAP to ROLAP if the data in the MOLAP system is older than, say, four hours.

With UDM at the center of the multidimensional model, you no longer need to have different methods of data access for different data sources. Before UDM, every system had a number of specialized data stores, each one containing data that was stored there for a limited number of users. Each of these data sources would likely require specific methods of data access for loading data into the multidimensional model. With Analysis Services, all the data of the enterprise is available through the UDM, even if those data sources are located on different types of hardware running different operating systems or different database systems. OLAP now serves as an intermediate system to guarantee effective access to the data.

Basic Concepts

When you start to build a multidimensional model, you think about business entities your organization operates with and about values that you need to analyze. For example, in our fictional organization—a chain of grocery stores known as Food Mart—we operate with warehouses, stores, products, customers, and different currencies, as shown in Figure 1.3. Those business entities became *dimensions* of our multidimensional model. Typically, you want to analyze data in a context of a time periods, and therefore the `Time` dimension is present in almost all multidimensional models. Actual values or facts that you are analyzing, such as sales, costs, and units, are called *measures*.

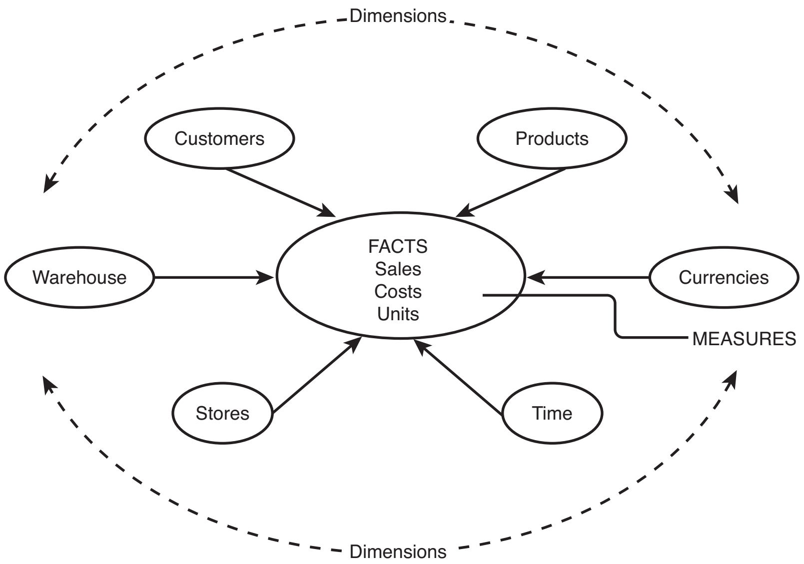

FIGURE 1.3 A multidimensional model consists of dimensions and measures.

Each individual element of the dimension is called a *member*. For example, "Club 1% Milk" is a member of the `Products` dimension, Irina Gorbach is a member of the `Customers` dimension, and January 1997 is a member of the `Time` dimension.

Each business entity usually has multiple characteristics. For instance, a customer can have the following properties: name, gender, city, state, and country. You might look at the products by name, Stock Keeping Unit (SKU), brand, product family, product category, and so on. We call these characteristics of the business entity *dimension attributes*. Figure 1.4 shows dimension attributes.

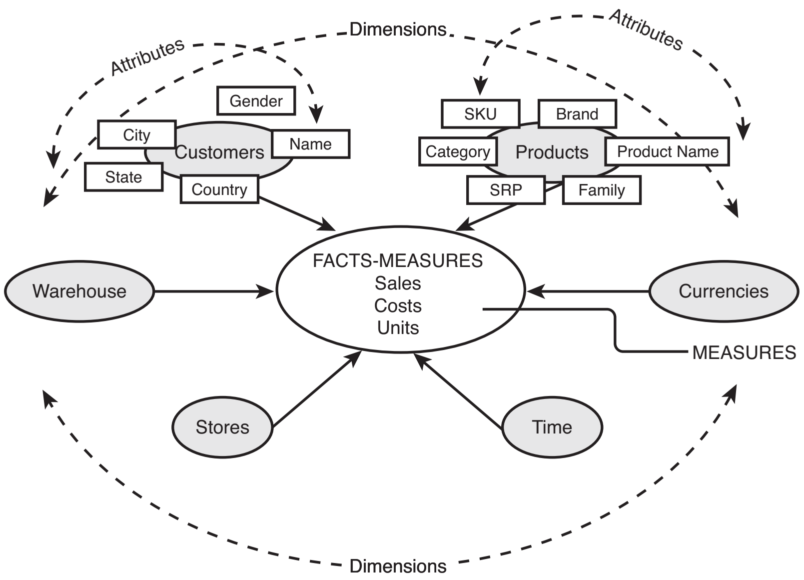

FIGURE 1.4 Each dimension is defined by its attributes.

Dimension attributes are not completely independent from each other. For example, `Year` contains `Quarter`, and `Quarter` contains `Month`. We can say that `Year`, `Quarter`, and `Month` attributes are related to each other.

If members of different attributes have a hierarchical structure, attributes can be organized in a *hierarchy*. For example, you can create the hierarchy `Calendar—Year` > `Quarter` > `Month` within the `Time` dimension, because the year contains quarters and quarters contains months.

After data is loaded in the cube, you can access it with many client applications. Microsoft Excel is one of the most frequently used application. Figure 1.5 shows Excel 2007 exposing data stored in Analysis Services cube.

This Excel spreadsheet demonstrates sales and cost for products in different time periods based on the data stored in the FoodMart 2008 database.

In Chapter 2, “Multidimensional Space,” we explain the terms that we use to describe multidimensional space.

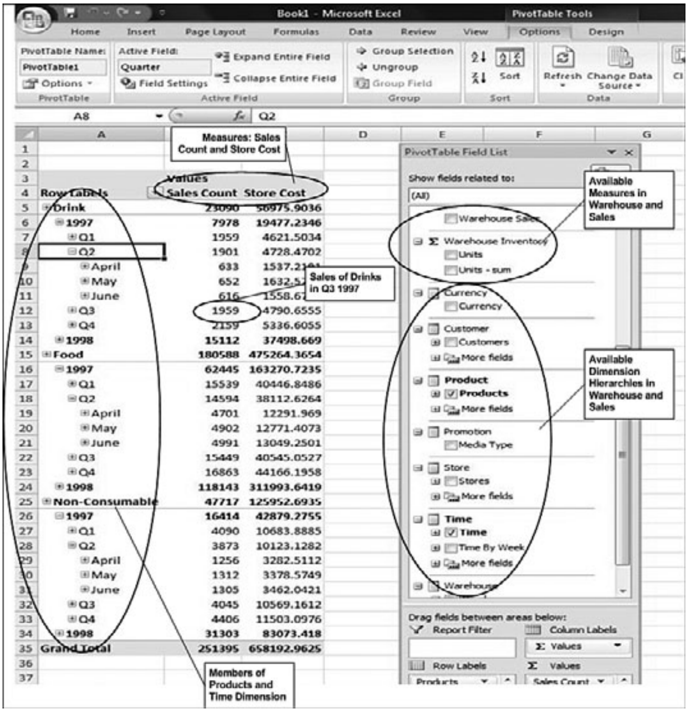

FIGURE 1.5 Accessing data in FoodMart 2008 sample using Excel 2007.

Index

A

B

D

E

F

G

H

J-K

L

M

N

O

P

R

T

V

W

X

This page intentionally left blank