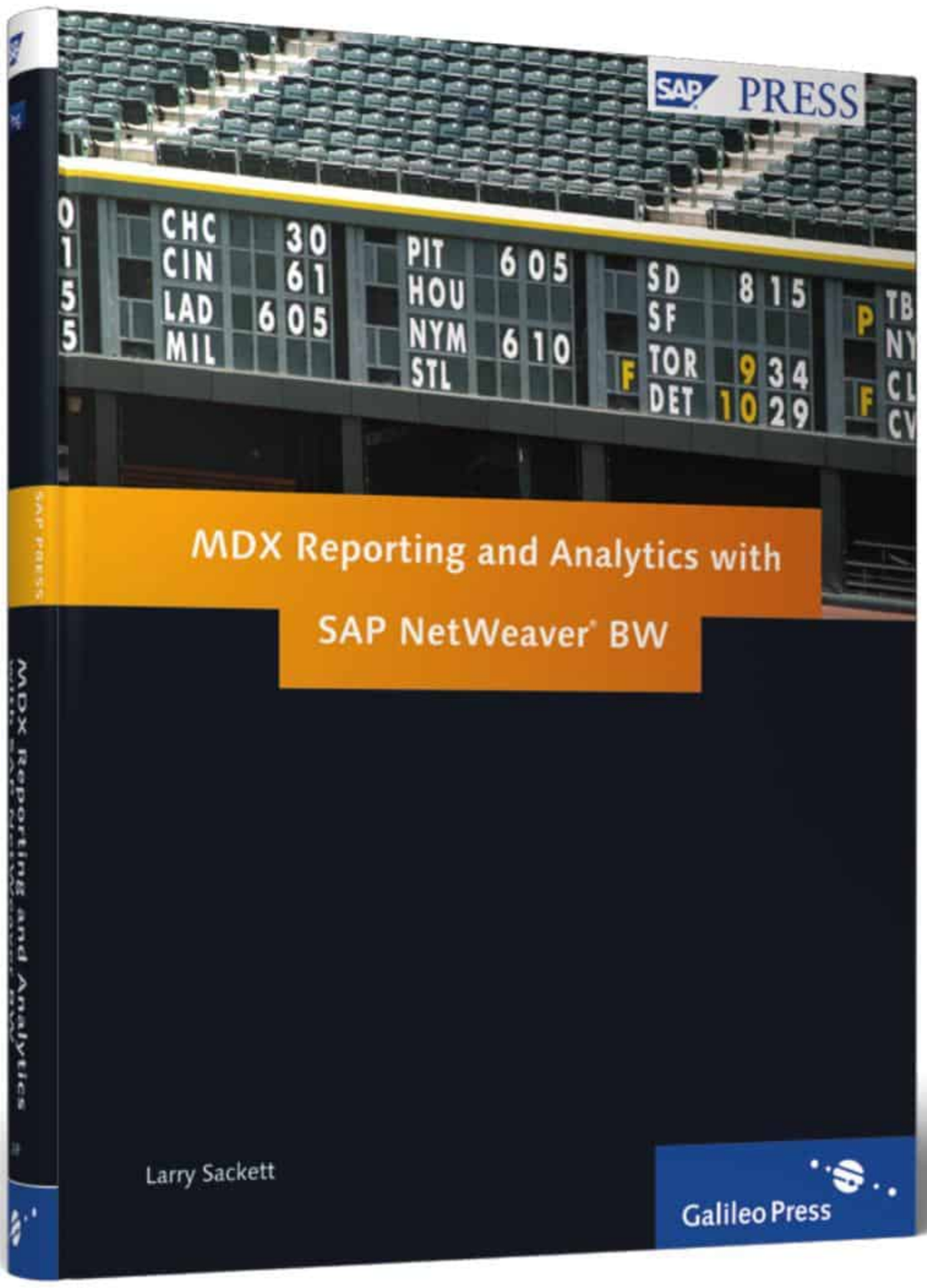
Larry Sackett

MDX Reporting and Analytics with SAP NetWeaver® BW



Galileo Press

Bonn • Boston

Contents at a Glance

Contents

MDX is an extraordinarily powerful query and calculation language for both advanced analytics and self-service reporting. It significantly extends the capabilities of SAP NetWeaver BW, making InfoCube data available and useful to users from the boardroom to the shop floor.

1 Why MDX Matters

1.1 The Business Case for MDX

At the end of the day, business intelligence is about getting the right information into the right hands at the right time to make faster, more profitable business decisions.

- A manufacturer's product manager wants to identify the company's 10 best customers and, in turn, the top 5 products purchased by these customers.
- A financial analyst wants to determine the percent contribution to sales for each of his company's products to their product subcategory, each subcategory to the product category, and ultimately each product category, subcategory, and product to overall sales.
- A sales manager is curious to know the sales rank for each member of her sales team for sales year-to-date versus their ranking for year-to-date sales for the same period last year.
- An inventory manager has to track changes to inventory as of the closing period for each of the past 12 months, and he'd also like to know mid-month if any stock-keeping units (SKUs) are dropping below predetermined inventory levels.
- A company's CEO wants to identify potentially at-risk customers, those that have substantially reduced spending in current periods compared to earlier periods.
- A risk management officer for an investment bank has to compare dozens of risk measures from the most current close of business to the previous close of business, and earlier dates, for compliance reports and to quickly spot anomalies.

All of these individuals want the ability to get this information through self-service web portals, dashboards, or automated daily emails. Equally important, they want the ability to interact with the data, drill down from summary to detail level, pivot rows and

columns, swap or nest dimensions to view results from different perspectives, or slice data to focus on individual products, customers, or time periods.

MDX is uniquely suited to these purposes. That's why, in less than a decade, MDX has become the de facto standard for reporting and analysis in business intelligence (BI) systems. It is supported by virtually every major vendor of BI servers, including Microsoft, SAP, Cognos (IBM), Hyperion (Oracle), Infor, and SAS.

1.2 What Makes MDX Unique

MDX, which stands for MultiDimensional eXpressions, has become popular for two primary reasons: the growth of the BI industry itself, and MDX's unique ability to navigate, query, and perform calculations against multidimensional structures.

The growth of BI has been fueled by the recognition that business users need fast and intuitive access to company information to make more profitable business decisions. This industry encompasses data warehouses, OLAP databases, and reporting and analysis tools.

OLAP stands for online analytical processing, and OLAP databases are usually referred to as *cubes*. The basis of an OLAP cube is the multidimensional database model, which consists of measures and dimensions joined through a central fact table. Multidimensional databases are also called *star schemas* because dimensions are arrayed around a fact table in a star-like structure (Figure 1.1). They are designed to give users direct access to information that is understandable, quick to find, and easy to use.

SAP NetWeaver BW Enhanced Star Schema

While still encompassing fact tables and dimensions, SAP NetWeaver BW enhances the basic star schema model with links from dimension tables to master data (texts and attributes) and external hierarchies. Dimension tables in an InfoCube don't actually contain data; instead they contain surrogate ID (SID) links to the master data and hierarchies. This is covered in more detail in Chapter 11.

Multidimensional OLAP databases are fundamentally different from the OLTP (online transaction processing) databases used in transaction systems. Instead of the star schema, an OLTP database has a schema that looks more like an intricate spider web, with hundreds of tables joined together through a complex set of primary and foreign key relationships. Designers of OLTP systems, such as SAP, worry more about making database updates and inserts lightning-fast than making them easy for reporting. Data in relational tables is often labeled using coding techniques that make the field names short and unique but that lack descriptions to make the tables more understandable to business users.

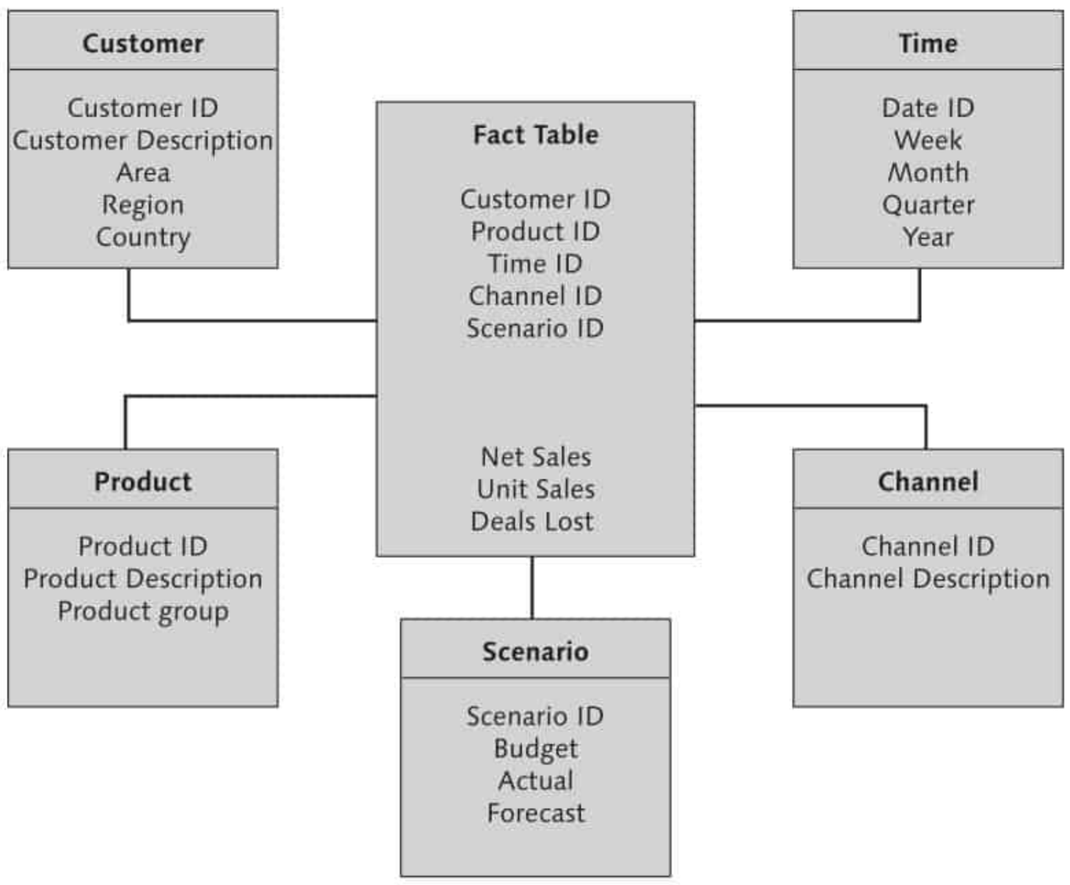


Figure 1.1 Simple Star Schema Database Design

Designers of OLAP systems are more concerned with the information and reporting needs of business users. OLAP systems are designed to:

- **Extract information of value** to business users from the source transaction systems.
- **Transform the information** so that it is easily understood, including creating new measures, adding intelligent descriptions to codes, and standardizing information, for example, putting addresses into U.S. Postal Service format.
- **Load the data** into a multidimensional database so that users can view measures by a wide variety of dimension characteristics, such as sales by product, region, time, and sales channel.
- **Frequently update** information in the OLAP database, typically on a daily basis, but also hourly, and in some cases, within minutes of a transaction.
- **Make it fast and easy** for end users to get information relevant to their day-to-day jobs.

Many dimensions in an OLAP database have multilevel hierarchies, making it possible to drill down from summary data to more detailed information. Examples of multilevel dimension hierarchies include time, with drill down from year to quarter to month to day; customer, with drill down from county to region to area to customer; and product, with drill down from product group to product.

This emphasis on multilevel hierarchies and drill down leads to the second reason for MDX's popularity: It is uniquely able to navigate, query, and perform calculations against multidimensional structures. These include OLAP cubes, dimensions, hierar-

chies, levels, attributes, members, and measures. Although it is theoretically possible to query an OLAP cube using SQL, SQL has no inherent knowledge of dimension hierarchies and cannot easily navigate to the parent or children of a particular dimension member.

MDX, however, recognizes all kinds of dimension relationships: from ancestors to descendants, from members to siblings, and from the dimension root (the All level) to the leaf level (most granular data). If you know a particular dimension member, then MDX lets you determine all of these relationships with very little coding. For example, if you know a particular product, you can query for that product's subcategory or its category with just the Parent or Ancestor MDX functions.

Meta data functions such as Parent, Children, Descendants, and ParallelPeriod enable you to relate levels and members without explicit reference, much like how a cell in an Excel spreadsheet can be related to another cell.

MDX consists of more than 150 expressions that cover mathematical, statistical, logical (Boolean), conditional, member, set, time series, and navigation functions. MDX expressions can be combined to create advanced calculations and filters, such as identifying at-risk customers who did business with you on a regular basis during the past 26 weeks, but have not during the most recent 4.

MDX includes several functions for time period comparisons. For example, ParallelPeriod compares results from a current period to the same period a year ago. PeriodsToDate analyzes data from a particular period in the past through the most current period. LastPeriods lets a user select a set of periods from a current period to a period in the past, for example, the past 3, 6, or 12 months. Other MDX time functions include YTD, QTD, and MTD, which are shorthand for the PeriodsToDate function.

In addition to functions that enable navigation of multidimensional structures, MDX includes a wide variety of functions that support reporting and analytics. Here's a sampling:

- The Filter function lets you select members of a dimension based on Boolean expressions, such as all customers that spent more than $500 in the past month. TopCount helps you determine your most popular products based on measures such as net sales or units sold. Related functions include TopSum, TopPercent, BottomCount, BottomSum, and BottomPercent.
- CrossJoin makes it possible to nest members from different dimensions on the columns and rows of a crosstab report. The Order function sorts dimension members by measures, such as an ordered list of products by their profitability.
- Logical functions include Intermediate If (IIF) to test whether a Boolean operation is true or false. Boolean operators are And, Or, Not, and Xor. IsEmpty is used to test whether data exists for a particular member in a cube. NOT and IsEmpty frequently

are used together to determine the last day for which data has been entered in a cube.

- Set functions help you select the Ancestor of a member at a given level of a multi-level hierarchy. Descendants find all of the children of a particular member at a given level of a dimension hierarchy.
- One of the most useful MDX functions is CurrentMember. Currentmember provides implicit reference to dimension members on the axes of a crosstab or to dimensions sliced in an MDX WHERE clause. What makes CurrentMember so powerful is that you can create named sets or calculated members that specify the current member of a dimension that is sliced in a dimension, with no explicit reference to that member. That makes it possible for users to simply slice on a member to change a calculation without having to modify the underlying MDX code. We'll explore the use of CurrentMember, PrevMember, and NextMember to *generalize* MDX expressions in Chapter 9.
- In addition to the full range of mathematical operators and Boolean functions, MDX includes statistical functions such as Correlation, Covariance, Standard Deviation, Variance, and several linear regression functions.

MDX functions are like Lego® bricks. Just as you can combine Lego bricks to build an infinite number of different robots, you can combine MDX functions to build different expressions that react to users' actions and needs.

1.3 How MDX Navigates External SAP NetWeaver BW Hierarchies

Much of the power of MDX lies in its ability to navigate up, down, and across hierarchies. ParallelPeriod is an excellent example of an MDX function that navigates up, down, and across hierarchies. Here's an example of how this function compares May 2005 with May 2004:

```
ParallelPeriod([Time].[Year], 1, [Time].[Month].[May 2005]
```

The first element of the expression, [Time].[Year], sets the level at which you want the function to look back. You could just as easily specify the quarter.

The second element, 1, tells the expression how many periods to look back — one year in this case.

The third element, [Time].[Month].[May 2005], identifies the reference period.

May is the fifth member in the set of months for 2005. The ParallelPeriod function will go up the set of month members for 2005 until it reaches the year level, go back one year to 2004, and then go back down the set of month members for 2004 until

it reaches the fifth member, [May 2004]. ParallelPeriod and other functions such as PeriodsToDate, Descendants, and Ancestors need a hierarchy structure such as Year, Quarter, and Month in which to operate. Many SAP NetWeaver BW InfoCubes have *flat* dimensions with an [ALL Level 00] and a second [Level 01] that is a list of members, as illustrated in Figure 1.2.

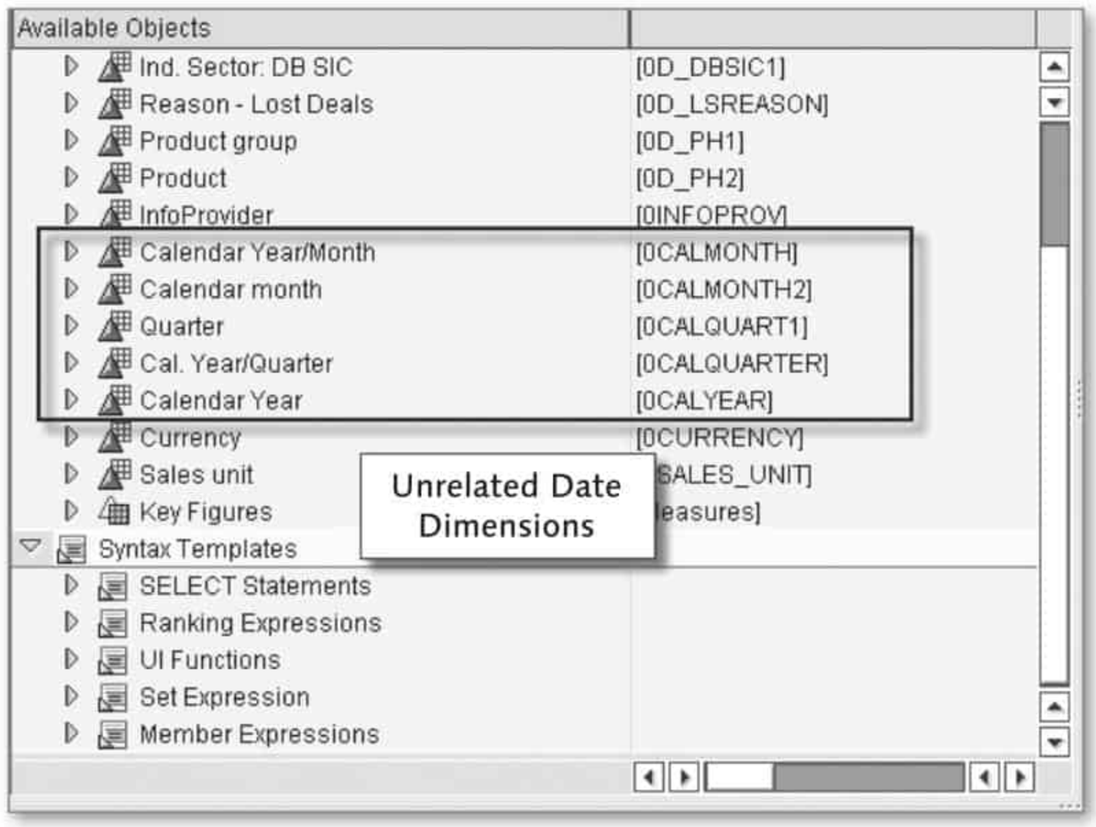


Figure 1.2 Unrelated Flat Date Dimensions

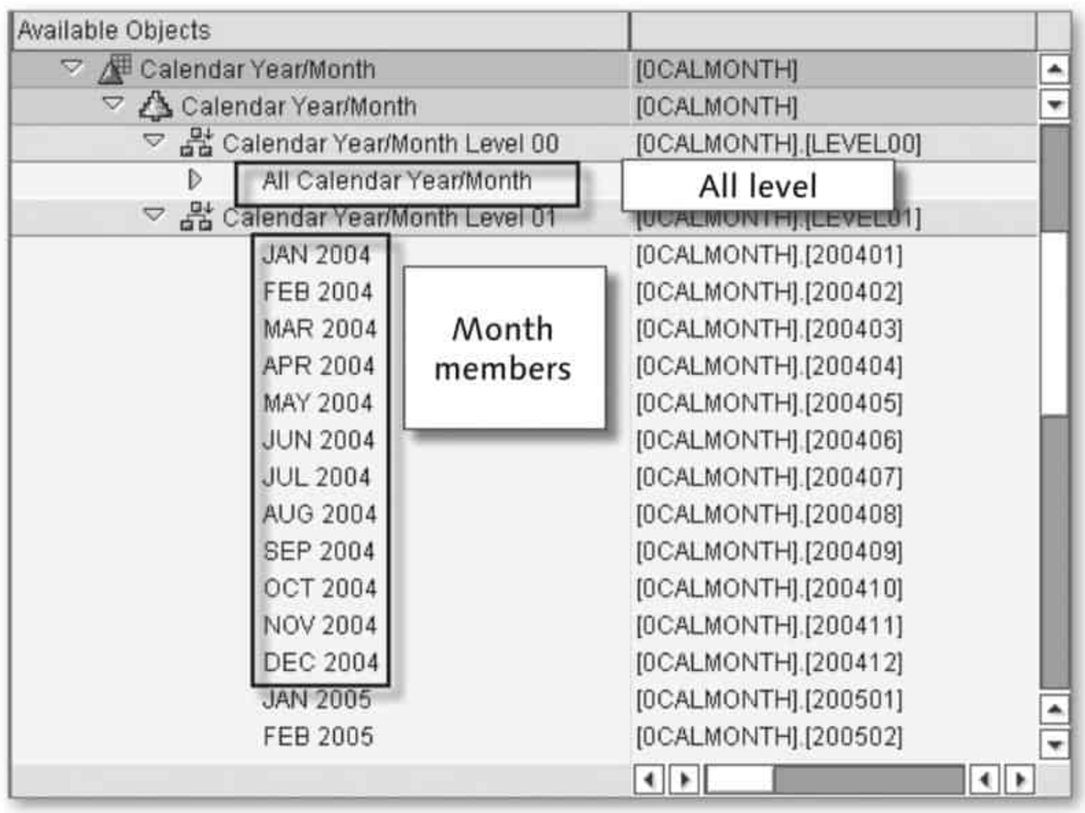


Figure 1.3 Flat Dimension [0CALMONTH] Only Contains Months with No Linkage to Quarters or Years

Figure 1.2 also shows that there are a variety of date characteristics in the DalSegno Company Reporting Cube. However, because there is no structure in the cube that links

these independent date characteristics, MDX can't navigate from one to the other. Figure 1.3 shows a flat dimension, [0CALMONTH], in which you have the All Level and the dates with no other levels.

Fortunately, SAP NetWeaver BW includes of series of prebuilt virtual time hierarchies that can easily be activated for basic date characteristics. These are called *virtual* time hierarchies because they are stored in main memory rather than in the database tables. An example of an activated virtual time hierarchy is shown in Figure 1.4. The virtual time hierarchy creates a structure that MDX can navigate from month to quarter to year. Activating and using the virtual time hierarchies is covered in Chapter 11.

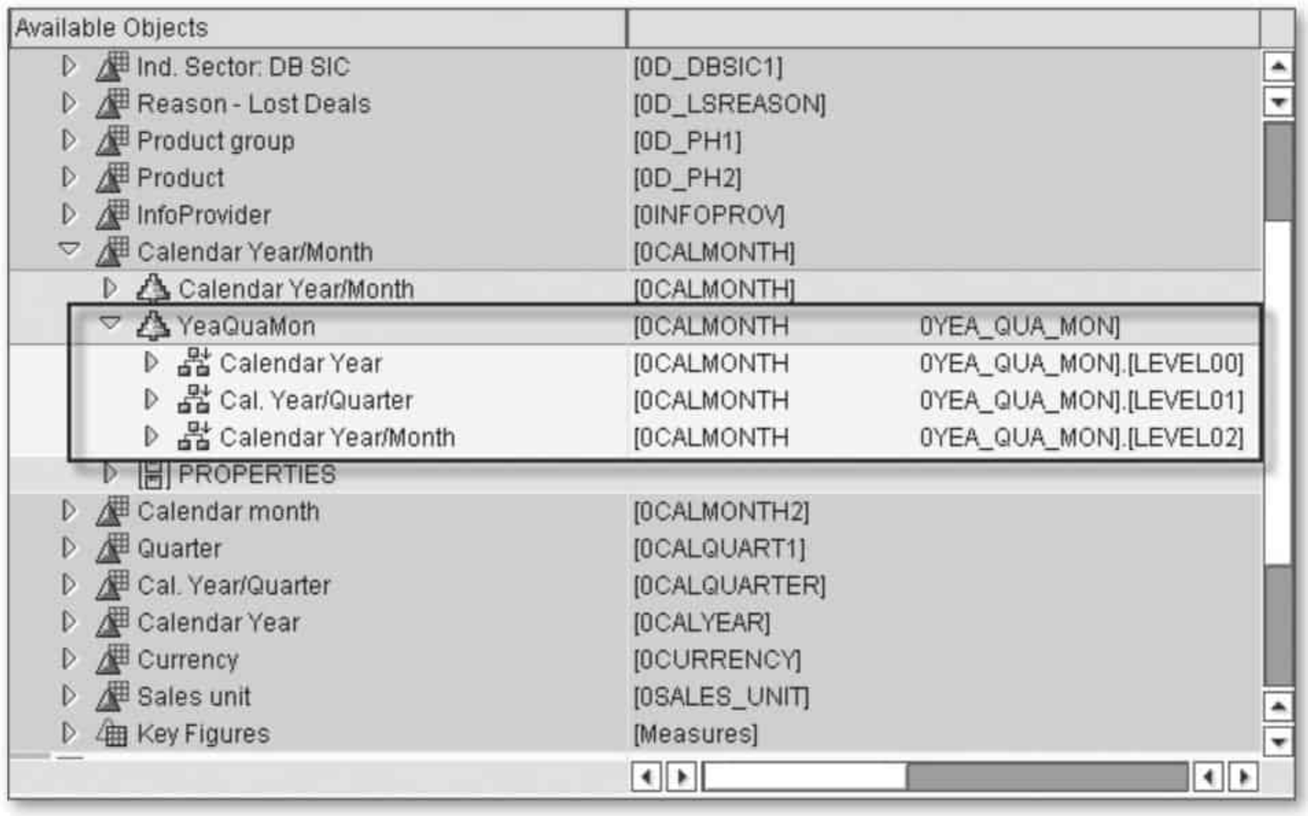


Figure 1.4 Virtual Time Hierarchy for Year, Quarter, and Month

With the virtual time hierarchy in place, the example in Figure 1.5 shows how the ParallelPeriod function works in the SAP NetWeaver BW MDX Test Editor.

❶ The ParallelPeriod expression references the Calendar Year in the Year, Quarter, Month (YEAQUAMONTH) hierarchy and looks back to the previous year (2004).

❷ The .CurrentMember generalizes the ParallelPeriod expression so that it will refer to each of the month members shown in the result set in the lower left side of the screenshot. The .Currentmember function will be discussed in more detail throughout this book.

❸ NON EMPTY removes any empty cells in the result set.

❹ The Descendants function is used to generate a list of all of the children (months) of the year 2005.

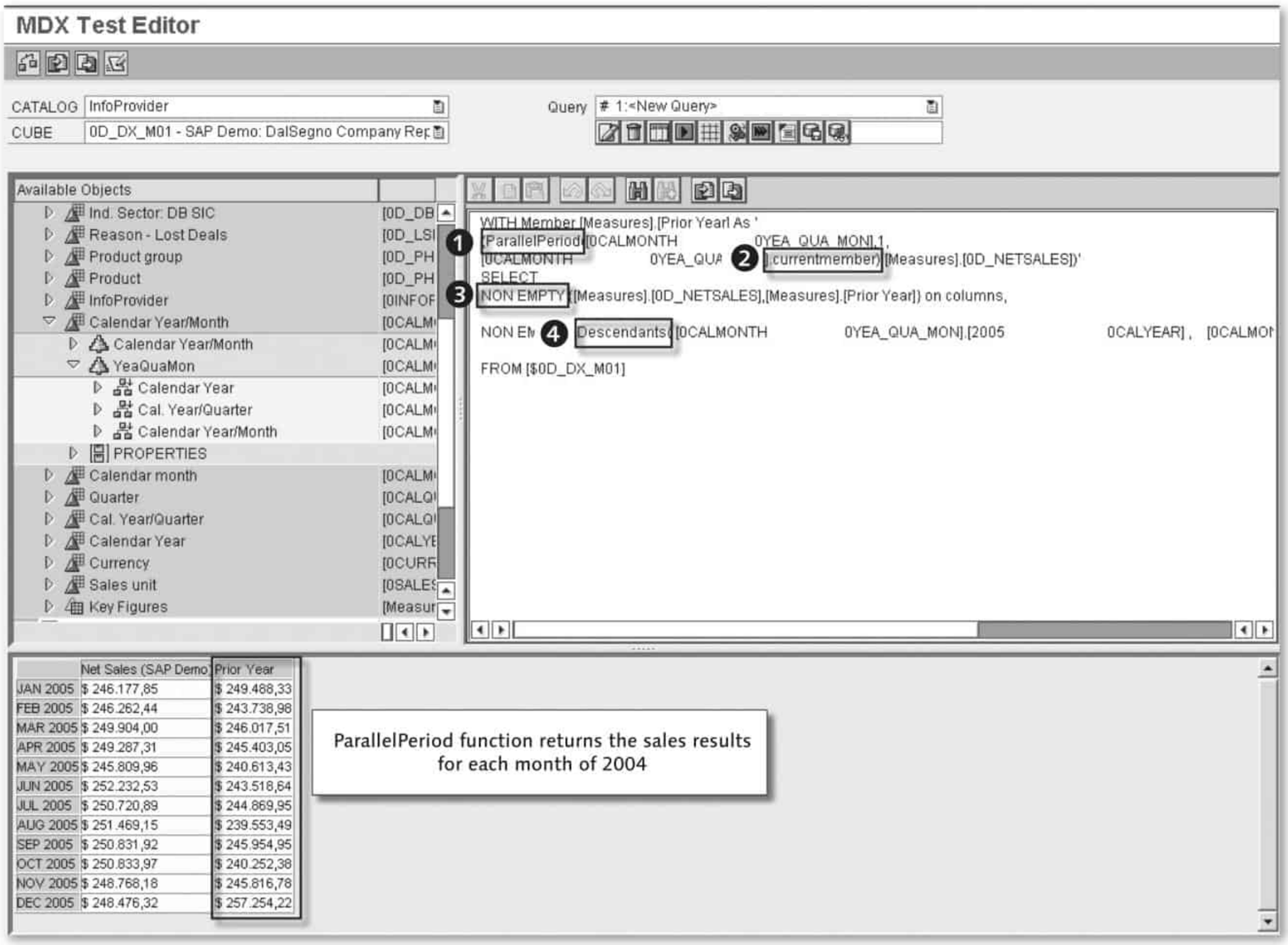


Figure 1.5 ParallelPeriod Used with the .CurrentMember function

When the full MDX query is executed, the ParallelPeriod function loops over each month of 2005, which are referenced by .CurrentMember, and returns the sales values for the corresponding months in 2004.

Note

As with all the code illustrations or samples shown in this book, the complete MDX expressions can be found on this book's web page at *www.sap-press.com.*

The DalSegno reporting cube also contains separate dimensions for product group and product, and for country, region, area, and customer. The characteristics have to be combined in an external hierarchy to enable MDX navigation.

Non-time hierarchies can be imported from SAP transaction systems, or can be built within SAP NetWeaver BW. They are called external hierarchies because they are maintained in external database tables. An example of an external hierarchy for product is shown in Figure 1.6. The process of creating an external hierarchy is more complex than simply activating a virtual time hierarchy and is described in depth in Chapter 11.

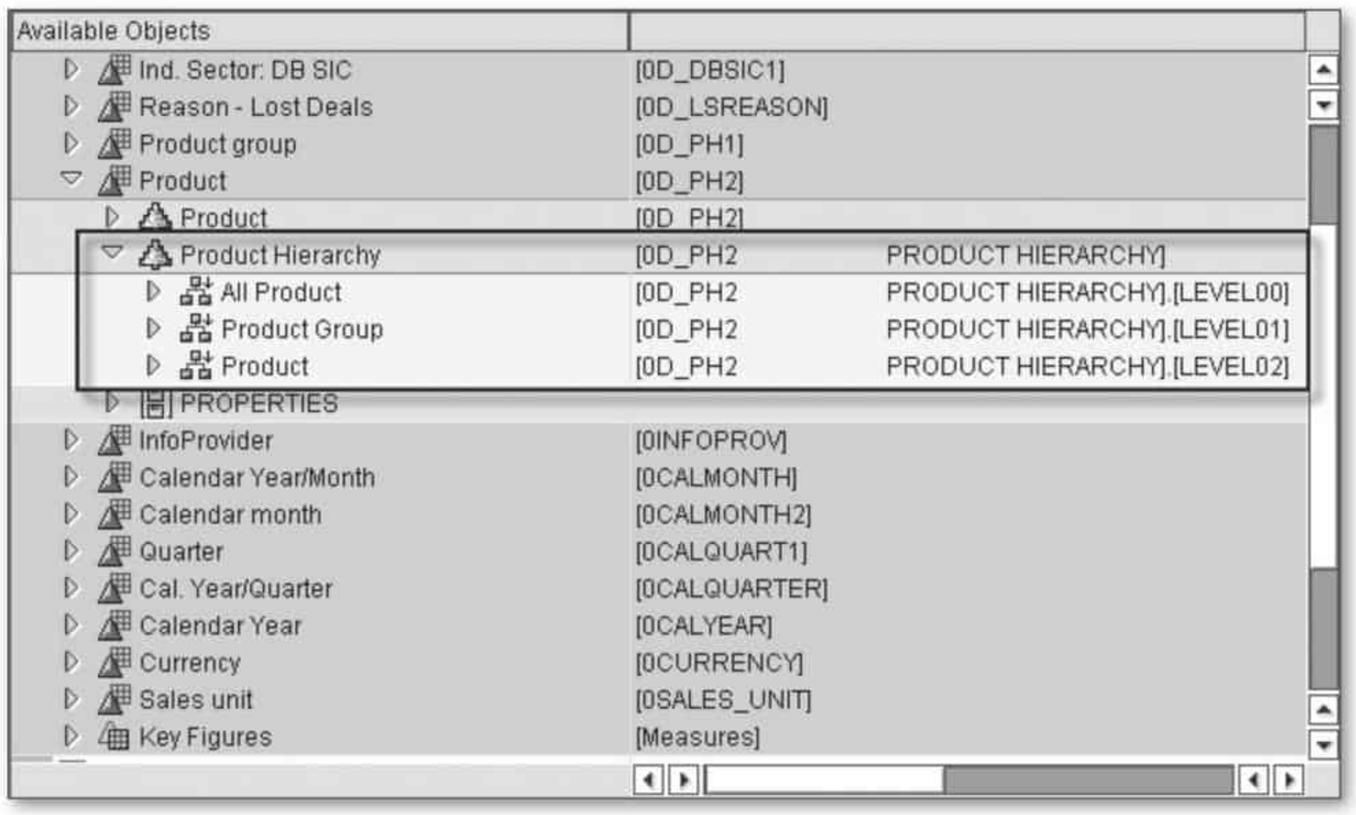


Figure 1.6 External Hierarchy for Product

1.4 History of MDX with SAP

MDX originated in the mid-1990s with Panorama, a small Israeli software firm that has since relocated to Toronto, Canada. Panorama had developed an OLAP server and wrote the first iteration of the MDX calculation engine and programming language. In 1996, Microsoft was exploring options for an OLAP server to be part of SQL Server 7.0. In July of that year, Microsoft representatives met with Panorama and quickly decided to acquire the OLAP Server and the query and calculation language.

In January 1997, a small team of Panorama developers, led by Amir Netz, joined Microsoft as part of the acquisition and relocated to Microsoft's Redmond, WA, campus. Mosha Pasumansky, a member of the relocated team, is credited by Netz as being the major developer of what was to become MDX.

One of the new team's first assignments was to define the MDX language, which became part of Microsoft's OLE DB for OLAP (ODBO) standard. The ODBO standard is built on Microsoft's OLE DB (Object Linking and Embedding, Database) interfaces for database access. Pasumansky did much of the work in defining MDX. He originally referred to MDX as Multi-Dimensional eXtensions because it was defined based on its mapping to the SQL query language.

In 1998 Microsoft released SQL Server 7.0, with Panorama's original OLAP server retooled as SQL Server 7.0 OLAP Manager. (Later this was renamed Analysis Services, with the addition of data mining algorithms in SQL Server 2000). Microsoft published

the ODBO standard in 1997. The standard defined the interfaces for connecting to multidimensional data stores, and the definitions of MDX functions, syntax, and SQL mappings. The standard was quickly embraced by OLAP server vendors such as Hyperion and SAS.

The published standard includes a series of mandatory and optional MDX elements. For this reason, and because of the way the SQL mappings can be interpreted, there are now different "dialects" of MDX. MDX in Microsoft Analysis Services works somewhat differently than in SAP NetWeaver BW. The same is true of the Hyperion OLAP engine owned by Oracle. This book focuses on the dialect in SAP NetWeaver BW.

1.5 SAP Adopts ODBO Standard

SAP also adopted the ODBO standard, and turned to Vancouver, Canada--based Simba Technologies to build an ODBO provider for SAP NetWeaver BW. SAP concurrently built an MDX engine to work with the Simba ODBO provider. The combination created an open interface that enabled third-party BI vendors to connect to the BW system. The work was completed in late 1998. Simba continues to work with SAP to this day on upgrades to the ODBO provider, including an updated ODBO interface for Microsoft Excel 2007 to SAP NetWeaver BW. ODBO is a Windows-based interface for component object model (COM) applications.

In April 2001 the first specification for an Internet-based interface was released: XML/A (Analysis). XML/A is an extension of ODBO that supports XML via HTTP/SOAP protocols for Web services. The second version of the XML/A specification was published in November 2002. SAP wrote its own XLM/A interface based on the specification.

SAP also added its own interface, OLAP BAPI, which executes remote function calls (RFCs) to SAP NetWeaver BW and SAP NetWeaver BW 3.X servers. The OLAP BAPI is the primary interface for the BusinessObjects Enterprise XI frontend BI tools.

All three interfaces — ODBO, XML/A, and OLAP BAPI — go to the same MDX processor in SAP NetWeaver BW (Figure 1.7). Therefore, MDX functions and syntax are exactly the same regardless of the interface.

Starting in 2006, the SAP NetWeaver BW development team and Panorama engineers worked together to design enhancements to the MDX engine. SAP and Panorama state that nearly 40 enhancements have been made since then. SAP has implemented all mandatory elements of the original ODBO standard and the majority of the optional elements. The result is a very robust set of MDX-based interfaces to SAP NetWeaver BW.

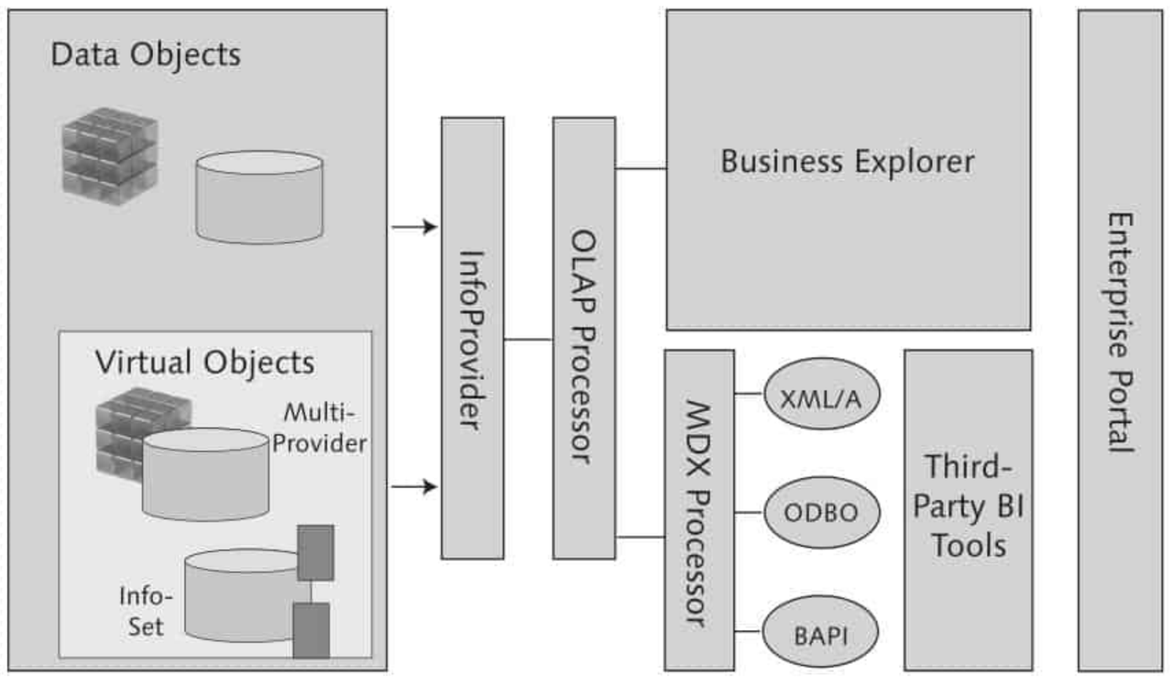


Figure 1.7 Three MDX-Based Interfaces to SAP NetWeaver BW

Panorama rewrote its client interface and MDX query engine to work directly with the Simba ODBO interface and the SAP NetWeaver BW MDX processor. Several demonstrations later in this book show how MDX queries in NovaView work with SAP NetWeaver BW, in addition to MDX queries in the SAP NetWeaver BW MDX Test Editor.

Note

SAP NetWeaver BW supports all of the mandatory elements of the ODBO specification, and the majority of optional ones. SAP added MDX functionality to support variables defined in SAP BEx Query Cubes, also referred to as InfoQueries. SAP also added numeric functions such as ArcSin, ArcCos, ArcTan, DIV, Log, Log10, and TRUNC that are not part of the ODBO specification. These additional functions are not discussed in this book.

1.6 MDX with BusinessObjects

Following the acquisition of BusinessObjects by SAP, the newly combined companies outlined a roadmap for integrating BusinessObjects BI tools with SAP NetWeaver BW. The roadmap was published in January 2008 and updated in May 2009.

The suite of BI tools has been grouped into a common, unified BI platform called SAP BusinessObjects Enterprise XI (currently R3.1). This includes Crystal Reports for formatted reporting, SAP BusinessObjects Web Intelligence for self-service ad hoc reporting, SAP BusinessObjects Predictive Workbench for enterprise mining and predictive analysis, Xcelsius for enterprise dashboards, SAP BusinessObjects Intelligent Search for search and discovery, SAP BusinessObjects Explorer (formerly Polestar) for data exploration,

SAP BusinessObjects Text Analysis for semantic analysis and sentiment extraction, and SAP BusinessObjects Voyager for OLAP analysis.

SAP will continue to invest in SAP NetWeaver BW, SAP NetWeaver BW Accelerator, SAP NetWeaver Master Data Management, and SAP extraction capabilities. SAP has stopped development but will continue to support the SAP Business Explorer (BEx) tools, including the SAP BEx report designer, analyzer (Excel and web), and web application designer. The SAP BEx query designer will continue to be supported indefinitely.

Crystal Reports is the suggested enterprise reporting solution. It can connect directly to SAP NetWeaver BW via the OLAP BAPI for both InfoCubes and InfoQueries. It is offered as a premium to SAP NetWeaver BW customers. The road map includes a lightweight version of Crystal Reports, to be released in 2010. Crystal Reports extracts data from SAP NetWeaver BW via the MDX interface. It includes an internal MDX query editor.

The future tool for OLAP analysis is Pioneer, a combination of the existing SAP BusinessObjects Voyager and the SAP BEx (Excel and web) analyzer. It will be available as a web client and an Excel client. The client will be based on Ajax (Web 2.0) technology and BI Consumer Services as a universal access layer for SAP and non-SAP multidimensional data. The first shipment of Pioneer is expected in 2010. Later versions of Pioneer will leverage the relational and OLAP universes in the SAP BusinessObjects Enterprise XI semantic layer via the OLAP BAPI.

Customers have to license SAP BusinessObjects Enterprise, which provides the infrastructure for most of the client tools, such as universe repository, scheduling services, caching services, and so on. SAP BusinessObjects Enterprise must run on a server separate from SAP NetWeaver BW. There are no plans to integrate the two products.

The SAP BEx analyzer and web analyzer will be phased out. Xcelsius will be the primary product for single, stand-alone dashboards. Xcelsius connects to SAP NetWeaver BW via SAP BusinessObjects Enterprise and OLAP BAPI. At some point it will be integrated with Pioneer. Later, SAP BusinessObjects will add direct connectivity to SAP NetWeaver BW (without the use of SAP BusinessObjects Enterprise). Xcelsius will be integrated with SAP NetWeaver Visual Composer.

The SAP BEx report designer will be replaced by Crystal Reports "light" as part of the SAP NetWeaver BW license. Information broadcasting will be extended for distribution of Crystal Reports and Pioneer workspaces and workbooks.

For web- and Excel-based OLAP analysis, SAP and Business Objects recommend staying with the SAP BEx analyzer and web analyzer for new products because they provide superior OLAP features compared to SAP BusinessObjects Voyager. Projects requiring Excel OLAP integration should be done with the SAP BEx analyzer 7.0. SAP customers will be able to upgrade to Pioneer Excel.

Customers can still use the SAP BEx web application designer, but the roadmap recommends shifting to SAP NetWeaver Visual Composer, which will be integrated with Xcelsius.

1.7 Alternatives to SAP BEx and SAP BusinessObjects

This chapter would not be complete without mentioning alternatives to both the SAP BEx suite and SAP BusinessObjects.

Any third-party BI tool that adheres to the ODBO or XML/A specification can connect to SAP NetWeaver BW via those two interfaces. Companies that have already invested in OLAP reporting and analysis tools such as Cognos or MicroStrategy do not have to convert to SAP BusinessObjects. By design, the SAP NetWeaver BW development team intends to maintain the open interfaces because they recognize that some SAP clients already have substantial investments in third-party tools.

Microsoft and SAP maintain a close collaboration with products such as Duet for the Microsoft Office suite and Microsoft Reporting Services (packaged with SQL Server 2005 and 2008). The two companies partnered in the development of an SAP BW XML/A provider, which connects Reporting Services to NetWeaver BW via a special XML/A interface. Reporting Services can connect to SAP BEx–defined InfoQueries or directly to InfoCubes. The SAP NetWeaver BW XML/A provider "flattens" MDX queries for use in tabular and matrix reports. The provider also incorporates SAP NetWeaver BW variables.

Microsoft Excel 2007 also connects to SAP NetWeaver BW via the Simba Technologies ODBO provider, which was updated to provide virtually the identical functionality of Excel 2007 as when it's connected to Microsoft Analysis Services 2005. Microsoft made substantial enhancements to Excel 2007 pivot tables, making it a viable BI tool with SAP NetWeaver BW. Users can drag and drop columns, rows, data values, and filters into pivot tables defined in the InfoQuery or available in an InfoCube. Chapter 13 of this book describes how Excel 2007 works with SAP NetWeaver BW.

Panorama's NovaView server and BI tools connect to SAP NetWeaver BW via the ODBO interface. As noted earlier in this chapter, Panorama rewrote the NovaView client interface to the ODBO provider and optimized its MDX query generator to work natively with the MDX processor in SAP NetWeaver BW. According to the SAP NetWeaver BW development team, Panorama is the only third-party vendor to go to this length to work with the SAP NetWeaver BW system. NovaView's entire user interface is based on MDX functionality. Chapter 15 covers NovaView reporting and analytics using SAP NetWeaver BW.

Microsoft Analysis Services 2005 or 2008 can extract data from SAP NetWeaver BW InfoCubes through SAP NetWeaver BW's Open Hub Destination. Using Analysis Ser-

vices can be an attractive option for companies that make heavy use of the Microsoft SQL Server BI suite. Microsoft's Integration Services, part of the SQL Server BI suite, includes an SAP NetWeaver BW connector. The downside of this approach is that none of the SAP NetWeaver BW security settings are extracted with the data, and the dimensions have to be redesigned in Analysis Services. Any changes made to InfoCubes also have to be taken into account in Analysis Services.

1.8 Summary

- MDX is the de facto standard for queries and calculations against multidimensional (OLAP) databases, such as SAP NetWeaver BW InfoCubes and SAP BEx InfoQueries.
- MDX consists of more than 150 expressions that cover mathematical, statistical, logical, conditional, member, set, time series, and navigation functions.
- MDX supports complex business analytics and self-service reporting.
- SAP was an early adopter of the OLE DB for OLAP standard and continues to enhance its implementation of MDX. The MDX-based interfaces to SAP NetWeaver BW provide open connectivity to any third-party BI tool that adheres to the ODBO or XML/A standards.
- To be most effective, MDX requires virtual time hierarchies and external hierarchies to navigate multilevel hierarchies.
- Third-party BI tools can be an effective alternative to the SAP BEx or SAP BusinessObjects frontend BI tools.
- The SAP BusinessObjects Enterprise XI server connects to SAP NetWeaver BW through the OLAP BAPI interface.

Index

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X