Chapter 10 – Multidimensional Databases

Matthew Moccaro

Summary

Multidimensional databases are those which are optimized for the retrieval of data by using multidimensional storage structures such as multidimensional arrays and data cubes. This emerging technology helps organizations to make strategic decisions and gives them a new way of thinking about large quantities of information [1]. They also give everyone a new way to more efficiently and effectively organize our data. Finally, even in relational and other types of databases, multidimensional concepts can still carry over to be an effective solution to a problem.

Data warehouses are the main setting for multidimensional database applications. Data warehouses are a server or group of servers which store a great magnitude of data [2]. This data is normally infrequently updated but still useful to conduct large-scale analytical queries. Preparing a data warehouse can be a difficult but worthwhile process when data needs preparation for the complex queries which will be run. The results of these queries help make decisions. Multidimensional databases are the key in which these applications become more efficient. Where a relational database may take minutes to complete a query, a multidimensional database can take only seconds to retrieve the requested dataset [3].

OLAP, or Online Analytical Processing, is a type of application which is concerned with obtaining specific information to make strategic decisions [2]. This type of application is becoming increasingly critical for businesses to stay current with trends and their competition. They also help to make customizations in marketing and other business aspects. OLAP queries are run on large amounts of data, normally stored in data warehouses which may or may not use multidimensional databases. Multiple versions of OLAP, including Relational OLAP, or ROLAP, and Multidimensional OLAP, or MOLAP, can also be explored. They can be a vital part of any business plan.

When learning any subject, a hands-on example is always helpful. Essbase is a multidimensional database system currently being maintained and developed by Oracle [4]. It was originally developed by Hyperion which has since been acquired by Oracle. Gaining actual experience with this tool can help to better outline certain aspects and to illustrate multidimensional aspects in clearer way. We can also see how to use a multidimensional database for actual analytical applications. Essbase is highly regarded, being on several innovative technology lists [5].

All of these components create a scenario where multidimensional databases can be an integral part of data analytics. Businesses can use these databases to visualize their data in the most organized way. Multidimensional databases can help to gather information quickly so that decisions can be made quicker, which can make a huge difference in the success of a business. Data warehouses store the data needed to make these decisions. OLAP applications can help to deliver this data in the most efficient way possible. Finally, Essbase is an actual example of all of these concepts in practice. Overall, we can see how multidimensional databases and their related components can help businesses everywhere be as efficient as possible.
Multidimensional Databases

Introduction

Databases have become an indispensable part of many technology related industries. They help us to store data and keep track of data. Databases are useful to store small sets of data as well as large ones, ranging from only a few sets of data for a small office to many terabytes of data for the biggest corporations. There are many different types of databases which have been in development since the mid-twentieth century. These different databases help to achieve the most efficient data storage structure for the data in use. Multidimensional databases have come into light recently compared to other database platforms, but are proving their worth in being very efficient in the field of data analysis [2].

The origins of multidimensional databases come from IRI Software and Comshare, two companies that, in the 1960s, began developing the initial traces of multidimensional data storage. IRI Express was the main application which allowed analytical processing. Comshare developed System W, which was another popular application for analysis. Finally, Hyperion Solutions released the Essbase system in 1991. This system was later bought by Oracle and has become one of the most popular multidimensional database systems to date [6].

Two other technologies were also developed in the 1990s which helped in the development of multidimensional databases. The concept of OLAP was brought forth by E.F. Codd, and this name has become synonymous with multidimensional databases and data analysis today. Also, data warehouses had begun to develop in many places. These warehouses held large amounts of data which were normally queried and analyzed using the popular relational model. However, with the development of multidimensional models, database administrators and application developers now had a new, more efficient tool to analyze their data.

The mass market has also seen an increase in the availability in this technology for smaller applications. Microsoft has released a multidimensional database system called MS OLAP Server which was first available in 1998. IBM also has a version of Essbase integrated with their popular DB2 server. Both of these services allow users to do several things. First, it allows users to get hands-on experience with this technology. This is a great advantage to both users and businesses alike as users of all experience levels will be able to use this technology in small-scale applications such as a small business financing analysis application.

Multidimensional Databases: Example by Comparison

To illustrate a multidimensional database, we will first explain and demonstrate the most common type of database, the relational database [2]. The relational database is one which uses the relational model. The relational model stores data within tables, where data can be easily organized, viewed, and updated. This is very similar to a desktop spreadsheet application. Each column in the table defines a field of data, describing all of the data below the heading. For example, if the column was entitled “Car Model,” all of the data in this column should represent a specific car model. Each row in the table defines a different “tuple” or “record.” This row is a set of related data that goes together in the database. This type of database is extremely common and used in small applications as well as extremely large applications, holding data for millions of transactions for customers and other types of data. An example of a typical relational database table is shown in Table 1.
Table 1. A typical relation database table [1].

<table>
<thead>
<tr>
<th>MODEL</th>
<th>COLOR</th>
<th>SALES</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINIVAN</td>
<td>BLUE</td>
<td>6</td>
</tr>
<tr>
<td>MINIVAN</td>
<td>RED</td>
<td>5</td>
</tr>
<tr>
<td>MINIVAN</td>
<td>WHITE</td>
<td>4</td>
</tr>
<tr>
<td>TRUCK</td>
<td>BLUE</td>
<td>3</td>
</tr>
<tr>
<td>TRUCK</td>
<td>RED</td>
<td>5</td>
</tr>
<tr>
<td>TRUCK</td>
<td>WHITE</td>
<td>5</td>
</tr>
<tr>
<td>SEDAN</td>
<td>BLUE</td>
<td>4</td>
</tr>
<tr>
<td>SEDAN</td>
<td>RED</td>
<td>3</td>
</tr>
<tr>
<td>SEDAN</td>
<td>WHITE</td>
<td>2</td>
</tr>
</tbody>
</table>

We can see that this table demonstrates an excellent way to organize data. In this table, we can see the sales for an automobile merchant over a certain period of time. The three fields are the model of the car, the color of the car, and the sales of each type of car. We can see that this information is organized and useful to make decisions. In relational databases, SQL, or structured query language, is used to run queries against the database which return the corresponding datasets. This is useful for data analysis.

When looking at this data, we can see that there is one inefficient feature that is very apparent throughout. Redundancy of several pieces of information is included in several tuples. This is demonstrated by each of the types of models: minivan, truck, and sedan, which are entered into the table multiple times. Also, each of the colors is entered multiple times, including blue, white, and red. As you can see, this table is not the most efficient form of storage for this particular set of data. Let’s take a look at another data structure which we can use for this problem.

In Figure 1, we can see the same data as shown in Table 1, only now it is stored in a multidimensional array [1]. This array is two dimensional, and two of the fields in our previous table are now each a dimension. The model field has now become the model dimension running along the left side of the figure, and the color dimension runs along the bottom. As one can see, this structure allows for very efficient data storage, and consequently, allows for efficient data retrieval as well. The setup comes with a much more obvious result to the end user, who can now retrieve their data more easily.

Figure 1 demonstrates a two dimensional array which models three fields which are model, numbers of sales, and color. Let’s take a look at the differences in data organization...
when we add yet another field to the example at hand. If we add the dealership field, which will tell us how many of each type of car is sold at each dealership, which can deduce further information to make better business decisions, such as which car sells best in which locations. To illustrate this concept in a multidimensional format, Figure 2 is shown with a three dimensional data cube.

![SALES
MODEL
MINIVAN
TRUCK
SEDAN
MODELCOLOR
BLUE RED WHITE

Figure 1. A two dimensional array representation of data [1].

![MODEL
MINIVAN
TRUCK
SEDAN
COLOR
BLUE RED WHITE

Figure 2. A three dimensional data cube [1].

In Figure 2, we can now see how multidimensional databases and concepts can truly come to light. In the table representation of this data, we would have had another field with repetition and redundancy throughout. In our data cube however, this third dimension can be added easily, and still keep our data organized and easily accessible. One can see how as the dimensions grow, the organization of the data is still kept intact.

Finally, to demonstrate a further use of multidimensional data, let us observe a fourth dimensional model, which includes time. As one can see in Figure 3, many data cubes are now...
spread out over a fourth dimension of time. So, if one were to query the database for a certain sales figure for a certain color, model, and dealership, the database would be able to easily return this data by selecting the correct data cube from the time dimension, and then gathering the necessary data. This is a perfect example of a multidimensional model for this data and also demonstrates the clear advantages over a relational model which would put this data into a table.

Figure 3. A four dimensional model of the sales data, including the time dimension [1].

Now that we have demonstrated conceptually how data can be stored in a multidimensional model, let us list some actual advantages that multidimensional models have over the relational model. These include ease of data presentation and navigation, as the output of a multidimensional database can be hard to match exactly with SQL queries. Another advantage included would be that of ease of maintenance. Multidimensional databases store data the same way that it is viewed. Relational databases must use joins and indexing to keep data intuitive, and these things require heavy amounts of maintenance. Finally, performance is important for OLAP applications and multidimensional databases excel in this area. Relational databases can use database tuning to improve their performance, and yet these levels are not as easily obtainable as a multidimensional database.

Finally, it is important to note that not all applications work well with multidimensional databases. Some datasets work much better with the standard relational model. A dataset with values that have a one-to-one mapping, such as an employee and their ID number, do not work well with multidimensional models. These models work best with data that has a high amount of interrelationships, such as the car sales data in the first part of this section.

**Multidimensional Database Query Concepts**

A multidimensional database has several types of queries and concepts which are used when manipulating this type of data. Most of the query concepts described here are used with both ROLAP and MOLAP [2]. This means that there are SQL equivalents to these queries, and they can be thought of in the normal relational model as well, only it may be difficult to visualize them. This also means that many of these queries are in use with star schemas and fact tables. Before discussing the specific types of queries, let’s discuss a few concepts and terms that are associated with multidimensional databases.
Fact tables can be an important part of multidimensional databases. A fact table is simply a table which stores all of the data regarding what aspects we would like to analyze [6]. So, if we would like to know all the types of cars sold in one month, we would have a fact table to tell us that information. A fact table can be described simply as a relational table in a relational database. For our purposes, this data can be thought of as being a part of a data cube as described in earlier sections. Each piece of data that we are concerned with, which in the case of our car example is sales, can make up one cell of the cube. Dimension tables can also be a part of this model, and give detailed information about each dimension itself. For example, for a car dealership dimension, this table would list things such as employees, location, and so on.

Another term which is associated with multidimensional databases is “Star Schemas” [2]. A star schema is a schema with the fact table in the center. Recall that a fact table holds the data we want to analyze, and in this example, that data represents the car sales. A typical star schema is shown in Figure 4. Notice how the fact table is labeled “Sales” and located in the center of the diagram. Then, the three dimensions represented in the cube example are branched off of this main central part. These three dimensions each represent a dimension table. If one normalizes a star schema, it becomes a snowflake schema. However, this is not normally done due the small size of the dimension tables, and that the schema is rarely updated.

Figure 4. An example star schema [2].
Another form described here is the “Constellation Schema”. A typical constellation schema is demonstrated in Figure 5. A constellation schema involves multiple fact tables which may or may not share a dimension table. As we can see in the example, both the sales fact table and the inventory fact table share the model dimension table and the dealership dimension table. For example, the sales data will be related to how many cars were sold in each dealership and how many of each model was sold as well. Besides this, the inventory may need to keep track of how many of each model is in inventory and how much inventory resides at each dealership location.

After clearing key terms and terminology, we can now begin to discuss the types of queries involved with multidimensional databases. The first query concepts which will be discussed are drilling down and rolling up. Pivoting, or rotation, will also be discussed. Finally, we will describe slicing and dicing concepts, as well as ranging. All of these query concepts are essential to OLAP [6][2].

Drilling down and rolling up are a concept which deals with hierarchies within dimension tables. Let us focus on the dealership aspect of our example. Dealerships can be located within a certain city, within a certain state, and within a certain country. Each of these locations deals with a range that is less and less specific. Obviously to run a query on this data, for the locations that are the most specific, such as “town”, we need to access more specific data. Calling this query will aggregate the data in the fact table as not all is necessary. When we move from less specific domains to more specific domains within a query, this technique is called “drilling down”. When we move in the opposite direction, such as moving from a very specific domain to a less specific one, this is called “rolling up.” These techniques can use previously computed results to obtain the desired results of the query much faster. “ROLLUP” has been also added as an SQL keyword in some implementations.

Another query concept is that of pivoting, or rotating. When visualizing the concept of pivoting, we imagine rotating a data cube so that different dimensions are more prominent. In SQL for ROLAP databases, this query can normally be performed through use of the “GROUP
BY” clause in a query. These different queries produce different views of not only the data, but the data cube itself. This adds great flexibility to users who are analyzing their data.

Slicing and dicing are two more important concepts to the idea of OLAP and multidimensional databases. These two terms deal with taking the main data cube and breaking it up into small “subcubes” of data. Dicing involves pivoting, where the pivot or rotation allows only some of the cells in the cube to be viewed. Slicing involves breaking the main cube up into subcubes by specifying a constant for a certain dimension and “slicing” the cube into pieces in this way. Both of these techniques are normally combined and done in tandem resulting in the common phrase of “slicing and dicing.”

Finally, ranging is another concept which breaks the main data cube up into a subcube to be analyzed. To perform this query, the user takes certain ranges of each dimension, and then uses those to construct a subcube. In our 3-dimensionals data cube example, each dimension has three options in the cube. Ranging would be to take a subset of those options on one or more dimensions and to create a smaller subcube of data.

To conclude this section, let us exemplify a typical multidimensional database query. Unlike relational databases, multidimensional databases do not have one standard query language. Instead this is mostly vendor or application specific. Many even support GUI interaction with queries represented by cubes that are able to be simply clicked on instead of writing out a typed query in text. The following query coincides with our car sales example. The query can be done as “PRINT TOTAL.(SALES_VOLUME KEEP MODEL DEALERSHIP)” [1]. This query will simply print a table of the total number of sales. The headings of each column will be the model of the cars, and the dealership where each amount of sales has taken place. The same query done in a relational database would be much longer and with several clauses including a “GROUP BY” clause and an “ORDER BY” clause. These simple queries save time and effort when viewing results and making time critical decisions.

Data Warehouses

Introduction

The term “data warehouse” was first heard in the 1970s. A data warehouse can be defined as being a large database whose main purpose is to support the decision-making process [2]. This is in contrast to an OLTP, or Online Transaction Processing system. An OLTP system is designed in light of real-time operations such as processing incoming orders and payments. It is important that these transactions are handled within a certain period of time as these operations are usually time critical. Because of this, these OLTP systems normally contain the most current data within a system and data is updated frequently. On the other hand, data warehouses normally contain a long history of data that is not frequently updated. These databases are geared more towards analysis rather than processing and maintaining a record of current transactions. This is where multidimensional databases can excel.

Preparing and Utilizing a Data Warehouse:

The end goal of a data warehouse is to assemble a large repository of data so the results of queries can be analyzed for decisions [3]. However, setting up this data warehouse is no simple task. Many steps must be taken to arrange all of the pieces of a data warehouse so that information can be retrieved as quickly and efficiently as possible. The process begins by
retrieving data from “source systems,” which normally include different applications in use by the organization decisions are being made for. This can include OLTP databases including data such as order processing and employee payroll. This data must be extracted from the source systems using whatever methods may be specific to that system. After the data is properly extracted, it must be transformed. This transformation can include many processes, including cleansing, validating, being integrated, and organizing. After this process, the data must then be loaded in a way which is efficient and effective for analysis. This loaded data must also be accessible to the people who must make the decisions, and these people may not always have an intense technical background. Therefore, it is essential that this data be prepared in a way that is also easy to understand. This entire process is known as the “Extract, Transform, and Load” process. It is abbreviated ETL.

After the data is properly brought through the ETL process, the data now resides in what is termed a “data mart.” A data mart is a database which contains organized data ready for analysis. These databases can be specialized for certain groups including the finance division of a company. Also, this data must be analyzed and then presented in a manner where the decision-makers can easily observe and understand the data and its trends. This can be done in the form of reports on data in the database. Groups of these reports and the analysis that goes with them are called “Business Intelligence.” The entire group of processes, including the ETL process and the presentation of the data to the people who make decisions, constitutes what is normally termed a data warehouse.

![Figure 6. The data warehouse preparation process [5].](image-url)
The data warehouse preparation process is illustrated in Figure 6. It begins with the user asking a strategic question which needs answering for the benefit of the organization. The ETL process is then initiated. It begins by extracting data from the source system. Then, once the data is extracted, it can be transformed. This is including all of the processes explained for the data transformation stage previously. Next, the data is loaded into the data mart and ready to be analyzed. Once the data is analyzed and the queries are run on the database, business intelligence is produced in the form of any preferred reporting materials. Finally, these materials are sent back to the original asked of the question, so that they may be analyzed and a decision can be made.

**Multidimensional Database Involvement**

Since the main focus of this chapter is multidimensional databases, it is important not to forget where these databases fit in with the picture of data warehouses. As one can see after reading through this section, data warehouses are the platform in which analytical queries are run. A data warehouse, and a multidimensional database for this example, must have two main goals in order to be used effectively. The first is the ability to be understood. The query results, as well as the data sitting in the data mart, are useless if the data cannot be interpreted and used for decision making. Secondly, the goal of producing timely results is also a top priority. Since multidimensional databases are implemented with query performance regarding analysis in mind, they prove their usefulness in this manner. As with the data warehouse preparation process, the process of setting up a dimensional model which will be effective for your organization is also no easy task. However, it is one that is worthwhile and able to keep your business thriving.

**OLAP**

**Introduction**

OLAP stands for Online Analytical Processing. Its main definition can be described as obtaining specific information to guide strategic decisions [2]. We have already presented two different types of OLAP in this chapter, including ROLAP and MOLAP. ROLAP stands for Relational OLAP, which uses the relational model for implementing the database and multidimensional concepts are simply used to visualize the data. MOLAP stands for Multidimensional OLAP, which is where the database is actually implemented and stored in the form of data cubes. In this section, we will describe OLAP in more detail, so that the role of multidimensional databases within OLAP can be better understood.

**A History**

In 1993, E.F. Codd coined the term OLAP [6]. In the report where this term first appeared, Codd and his co-authors defined twelve rules for an application to actually be considered as an OLAP application [7]. However, this proposal did not succeed, yet the term was carried on and used everywhere. Further on in time, it was suggested that all OLAP tools must pass the FASMI test, which is an abbreviation for Fast Analysis of Shared Multidimensional Information. This test ensured several characteristics to make the database efficient for analysis via complex queries. This test, as well as the characteristic tests which come with it, do not truly define a standard measure for each of their attributes. This means that
instead of this test, the overall consensus is that for an application to be considered an OLAP application, it must be able to present data in a multidimensional view.

There was, at one point, an OLAP council. This council formed in the mid 1990s. However, no major players entered into the council, and it eventually was forgotten. Because of this, no standard has ever been set for the OLAP model. There is no one set query language or data model for OLAP at the time of this writing, only vendor specific languages for each multidimensional database.

**Comparison to OLTP and Other Topics**

OLAP is almost always compared with OLTP. Recall that OLTP deals with maintaining and storing current data, whilst OLAP is concerned with analysis and storage of a multitude of data for analysis. This leads to the notion that OLTP’s workload is foreseeable, while OLAP’s workload is unforeseeable. This is because an OLTP system is designed to deal with the same queries frequently, including updating, inserting, and deleting data. OLAP systems are designed to determine different trends and to constantly search for data that is useful to solve a problem. Therefore, these problems can be new each time the data is queried. Another difference between OLTP and OLAP is that the data with an OLTP system is read and write while the data within an OLAP system is read only. This is because OLTP systems will need to constantly update their data for things such as transactions and user accounts. OLAP systems are made to have queries run against data for the past several years, and if this data were to be changed it would compromise the integrity of the entire system. Therefore, OLAP systems should only be read from. The query structure for an OLTP system is normally also simple. Simple update and insert queries should cover most of the queries arranged in an OLTP system. They also do not cover large volumes of data. OLAP queries however, normally deal with huge amounts of data, with complex clauses in the query statements. Finally, another important difference between OLAP and OLTP systems is the number of users. An OLTP database, such as one for a retail store chain, may have millions of users each day. However, an OLAP database may have only a few users, as only the organization’s decision-makers need to be concerned with the data.

Even though there is no standard query language for OLAP systems and multidimensional databases, there have been several research proposals towards this technology. In industry, the accepted standard is MDX, which stands for Multidimensional Expressions. This language was developed by Microsoft and released in 1997. An example query template of this type of language would be:

```
[ WITH <MeasureDefinition>+ ]
SELECT <DimensionSpecification>+
FROM <CubeName>
[WHERE <SlicerClause> ] [7]
```

As one can see, this query has syntax similar to that of SQL, which makes it very easy for a database user to make the transition from relational models to multidimensional models. One can also see in this example how a specific cube can be selected along with a specific dimension along that cube. Slicing can be performed in the WHERE clause of this query. The WITH clause of this query also allow complex calculations. This makes multidimensional queries more accessible. The following is an actual query for a multidimensional database:
WITH MEMBER [Measures].[pending] AS '[Measures].[Units Ordered]-[Measures].[Units Shipped]'

SELECT
{[Time].[2006].children} ON COLUMNS,
{[Warehouse].[Warehouse Name].members} ON ROWS
FROM Inventory
WHERE ([Measures].[pending],[Trademark].[Acme]); [7]

Because of the frequent complexities of these queries however, OLAP queries are normally not written out by hand. The schemas of these databases are normally extremely detailed and complicated. Because of this, a graphical user interface and the use of the mouse are employed to give the user an easier way to obtain the results they seek instead. The improvement of user interaction is a subject of researchers trying to improve the usability of multidimensional databases. An application called Essbase will be described later in this chapter giving an example of such a program.

Finally, security is also a large problem with OLAP applications and databases. Earlier in this model’s history, security was not as big of a concern as less people normally had access to OLAP databases. However, as time and the technology progressed, more and more users are normally connected with OLAP databases. The need for security of this data is greater than ever as many years of personal data can be found in these databases. Therefore, the privacy of many users, and more likely customers, is at stake. This is a great area for research as these databases and applications will only grow in size and complexity and the need for security will only increase.

**Benefits and Advantages**

With all of this discussion of the benefits of multidimensional databases, data warehouses, and OLAP applications, it is important to list the specific advantages of using these tools for data analysis to make all of this discussion seem more worthwhile. One of the most important benefits of this system is understanding customers and their behavior [3]. This is one of the most important things organizations are looking to understand. Certain reports which help identify purchase trends of users can help organizations keep their inventories stocked with the right items to sell. Another benefit is developing new products by viewing certain research testing results which can be used to optimize products with customer needs. Finally, another benefit would be to understand financial results of a company. Hidden trends in financial data can be invaluable for saving the company money in the long and short term. These are only a few of the many benefits of using OLAP applications with multidimensional databases and data warehouses.
**Essbase – An Example**

**Introduction**

Essbase is one of the most popular multidimensional databases in today’s market. This product began development with a company named Arbor Software. Arbor merged with Hyperion in 1998 and finally Hyperion was bought by Oracle in 2007 [6]. After these major changes, the product is currently labeled “Oracle Essbase.” The name “Essbase” stands for “extended spreadsheet database.” Theoretically, multidimensional databases are very complex and interesting. However, theoretical knowledge is not complete without actual experience using a system. Therefore, in this section we will explore how to use a multidimensional database and discover the process of storing and querying data with this popular product.

**Managing An Essbase Database**

Managing an Essbase database is similar to many other relational database systems and is simple to do. There are two ways in which an Essbase database can store data [4]. One way is with block storage. The other way is with aggregate storage. Block storage is preferred for data which is dense, and aggregate storage is geared towards data which is sparse. To manage the database itself, a database administrator can use the Administration Services Console. This console is a simple GUI program used to manage the database. An example of this GUI can be seen in Figure 7. It is organized into three frames which include the navigation frame, for providing a graphical view of the Essbase environment being worked on, the object window frame, which is where objects are displayed after being opened, and the message frame, which gives the administrator messages from the system. In the figure, the navigation frame is seen on the left, the object window frame is seen on the right, and the messages frame is seen running along the bottom of the figure.

There is a specific process which must be completed to create a database. All of the processes described in this section can be performed in the Administration Services Console unless otherwise mentioned. The first step is to create a database outline. Creating a database outline involves determining how your database will be structured, including defining the dimensions of data cubes, as well as the hierarchies associated with them. Creating a database outline is done in the Outline Editor of the Administration Services Console. The next step is to load the data using rule files. Rule files simply help to make the process of loading data easier by allowing administrators to clean data as it is loaded. For example, a rule file could scale data values as they are loaded so that they immediately work well with the format of the database. Another example would be that a rule file could help to keep member names within a certain format, or to ignore specific incoming fields in a data source. Thirdly, calculation scripts can be used to determine how an administrator would like to calculate a database if it is not already defined within the database outline. Finally, Smart View is an application interface for Microsoft Office and Essbase. It allows users to perform reporting and other tasks right from within Microsoft Office. Smart View works with Microsoft Excel, Word, and PowerPoint. Microsoft Office is a platform with which many users all across the world are familiar with. This becomes a great and important feature from which business owners and decision makers, who are not always the most technically oriented, can still make decisions and find multidimensional databases useful.
Another important component of the Essbase System is the Essbase Studio. This graphical tool allows users to develop and deploy Essbase analytical applications. The most interesting feature of the Essbase studio is that it allows users to build Essbase data cubes from snowflake and star schemas, which were described in an earlier part of this chapter. It also allows the user to create data cubes from relational databases as well. This tool is new to the Essbase platform, yet becomes an important part of developing applications.

**Creating A Database Model**

Creating an actual database model for use in a multidimensional database is an easy task with the Administration Services Console. We will now review this process so that one can see exactly how a multidimensional database is created. To begin, one will create a database, and open the Outline Editor. The Outline Editor shows the database outline in tree form. To add a new dimension to the database, one can right click on the highest part of the outline, which, in Figure 8 is the outline name, click on “Add Child”, and enter the name of the new dimension into the provided text box. This process also creates a hierarchy for your database. As one can see, this process is not difficult, and allows any users familiar with basic computer operations to create a database outline and database model. Remember that after the outline is set, rule files will be made to ensure that data is imported in the correct format. Then, the data from the data sources will be extracted and imported into the database, following the given database outline.
Using Smartview

Now that we have created a database model, we will now review how to view our data in the Smart View application, so that we can make business decisions in the future using these concepts. In this example, we will see how to use the Smart View extension to view data in excel. It is a very useful tool, and one which coincides with software many users have worked with before.

Figure 8. Entering a new dimension into a database. [4]

Figure 9. Using Smart View in Microsoft Excel [4].
To begin, one will open up a running instance of Microsoft Excel. Then, open the Smart View Data Source Manager. We will now create an ad hoc report and initiate a query against the Essbase database. Then, we will view the data within Microsoft Excel. An example of this is shown in Figure 9, where we can see the data for IBM shown in the spreadsheet. For the next step, we will open up and connect to a server instance by finding the specific database in the Data Source Manager and then logging in. Then we will right click on the database and select “Ad-hoc Analysis.” If one wishes to drill down into the hierarchy of the database, one only needs to double-click on that particular cell in the database to view the data. As can be seen in Figure 9, this format is not only familiar, but allows non-technical decision makers to view the data in their data warehouses in the simple format of an excel spreadsheet.

Review and Conclusion

In this chapter, we have reviewed the many aspects of multidimensional databases. We have seen that, most importantly, multidimensional databases help to more quickly and efficiently analyze data. The first concepts which were reviewed were the differences between multidimensional databases and relational databases. We saw how multidimensional arrays can help to better organize and display data. We also saw how this is now always the best format for certain types of data, such as individual mappings. Next we saw conceptually how a three dimensional database and a four dimensional database for car sales over a period of time would be set up. This chapter also discusses star schemas and constellation schemas. Next, the types of query concepts, such as drilling down as slicing and dicing, that were associated with multidimensional databases were explained.

In the following section, we discussed data warehouses. Recall that data warehouses are large databases filled with data to be analyzed. This data does not need to be updated frequently but is normally found in very large amounts. The process to prepare and load data into a data warehouse was explained in several stages. OLAP was discussed in the next section. OLAP stands for Online Analytical Processing and it can be defined as the process of obtaining specific information to make strategic decisions for an organization. With OLAP, no standard query language has been defined for OLAP applications. There are many differences between OLAP and OLTP, and security is a true concern for today’s OLAP applications and databases. Finally, Essbase is an example of a multidimensional OLAP system in use today. It is currently developed by Oracle and uses several GUIs to accomplish certain tasks. The processes involved in setting up a database with the Administration Services Console are discussed. Also, creating a database model and viewing data in the Smart View Microsoft Office extension are explained. Overall, this chapter discusses the most important parts of multidimensional databases, and gives the reader a starting point for using them when timely data analysis is required. It can help all organizations gain a great advantage in today’s competitive business world.

References


