Introduction to OLAP

1. Multidimensional Model

The data warehouse layer is a vitally important part of this book. Here, we introduce a data warehouse key word: multidimensional. You need to become familiar with the concepts and terminology used here to understand the information presented throughout this book, particularly information regarding conceptual and logical modeling and designing.

Over the last few years, multidimensional databases have generated much research and market interest because they are fundamental for many decision-making support applications, such as data warehouse systems. The reason why the multidimensional model is used as a paradigm of data warehouse data representation is fundamentally connected to its ease of use and intuitiveness even for IT newbies. The multidimensional model's success is also linked to the widespread use of productivity tools, such as spreadsheets, that adopt the multidimensional model as a visualization paradigm.

Perhaps the best starting point to approach the multidimensional model effectively is a definition of the types of queries for which this model is best suited. Section 1.7 offers more details on typical decision-making queries such as those listed here:

- "*What is the total amount of receipts recorded last year per state and per product category?*"
- "*What is the relationship between the trend of PC manufacturers' shares and quarter gains over the last five years?*"
- "*Which orders maximize receipts?*"
- "*Which one of two new treatments will result in a decrease in the average period of admission?*"
- "*What is the relationship between profit gained by the shipments consisting of less than 10 items and the profit gained by the shipments of more than 10 items?*"

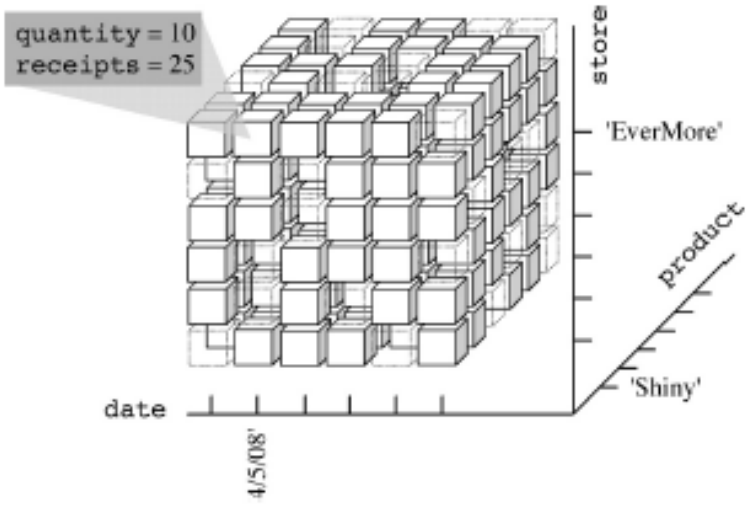
It is clear that using traditional languages, such as SQL, to express these types of queries can be a very difficult task for inexperienced users. It is also clear that running these types of queries against operational databases would result in an unacceptably long response time.

The multidimensional model begins with the observation that the factors affecting decision-making processes are enterprise-specific *facts*, such as sales, shipments, hospital admissions, surgeries, and so on. Instances of a fact correspond to *events* that occurred. For example, every single sale or shipment carried out is an event. Each fact is described by the values of a set of relevant *measures* that provide a quantitative description of events. For example, sales receipts, amounts shipped, hospital admission costs, and surgery time are measures.

Obviously, a huge number of events occur in typical enterprises—too many to analyze one by one. Imagine placing them all into an *n*-dimensional space to help us quickly select and sort them out. The *n*-dimensional space axes are called *analysis dimensions,* and they define different perspectives to single out events. For example, the sales in a store chain can be represented in a three-dimensional space whose dimensions are products, stores, and dates. As far as shipments are concerned, products, shipment dates, orders, destinations, and terms & conditions can be used as dimensions. Hospital admissions can be defined by the department-date-patient combination, and you would need to add the type of operation to classify surgery operations.

The concept of dimension gave life to the broadly used metaphor of *cube*s to represent multidimensional data. According to this metaphor, events are associated with cube cells and cube edges stand for analysis dimensions. If more than three dimensions exist, the cube is called a *hypercube*. Each cube cell is given a value for each measure. Figure 1 shows an intuitive representation of a cube in which the fact is a sale in a store chain. Its analysis dimensions are *store*, *product* and *date*. An event stands for a specific item sold in a specific store on a specific date, and it is described by two measures: the *quantity* sold and the *receipts*. This figure highlights that the cube is *sparse*—this means that many events did not actually take place. Of course, you cannot sell every item every day in every store.

FIGURE 1
The three-dimensional cube modeling sales in a store chain:10 packs of Shiny were sold on 4/5/2008 in the EverMore store, totaling $25



If you want to use the relational model to represent this cube, you could use the following relational schema:

SALES(store, product, date, quantity, receipts)

Here, the underlined attributes make up the primary key and events are associated with tuples, such as <'EverMore', 'Shiny', '04/05/08', 10, 25>. The constraint expressed by this primary key specifies that two events cannot be associated with an individual store, product, and date value combination, and that every value combination *functionally determines* a unique value for quantity and a unique value for receipts. This means that the following functional dependency2 holds:

store, product, date → quantity, receipts

To avoid any misunderstanding of the term *event*, you should realize that the group of dimensions selected for a fact representation singles out a unique event in the multidimensional model, but the group does not necessarily single out a unique event in the application domain. To make this statement clearer, consider once again the sales example. In the application domain, one single *sales* event is supposed to be a customer's purchase of a set of products from a store on a specific date. In practice, this corresponds to a sales receipt. From the viewpoint of the multidimensional model, if the sales fact has the product, store, and date dimensions, an event will be the daily total amount of an item sold in a store. It is clear that the difference between both interpretations depends on sales receipts that generally include various items, and on individual items that are generally sold many times every day in a store. In the following sections, we use the terms *event* and *fact* to make reference to the granularity taken by events and facts in the multidimensional model.

Normally, each dimension is associated with a *hierarch*y of aggregation levels, often called *roll-up hierarchy*. Roll-up hierarchies group aggregation level values in different ways. Hierarchies consist of levels called *dimensional attributes*. Figure 2 shows a simple example of hierarchies built on the product and store dimensions: products are classified into types, and are then further classified into categories. Stores are located in cities belonging to states. On top of each hierarchy is a fake level that includes all the dimension-related values. From the viewpoint of relational theory, you can use a set of functional dependencies between dimensional attributes to express a hierarchy:

product → type → category
store → city → state

In summary, a *multidimensional* cube hinges on a *fact* relevant to decision-making. It shows a set of *events* for which numeric *measures* provide a quantitative description. Each cube axis shows a possible analysis *dimension*. Each dimension can be analyzed at different detail levels specified by hierarchically structured *attributes*.

The scientific literature shows many formal expressions of the multidimensional model, which can be more or less complex and comprehensive. We'll briefly mention alternative terms used for the multidimensional model in the scientific literature and in commercial tools.

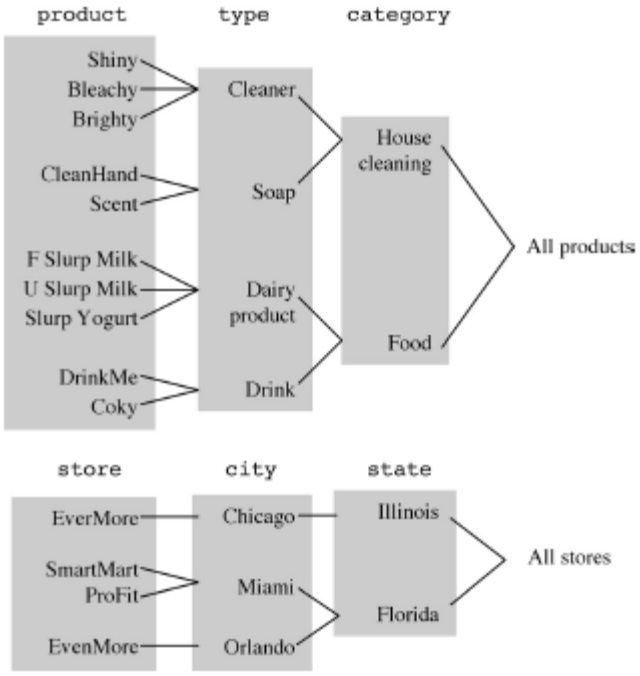


FIGURE 2
Aggregation hierarchies built on the product and store dimensions

The *fact* and *cube* terms are often interchangeably used. Essentially, everyone agrees on the use of the term *dimensions* to specify the coordinates that classify and identify fact occurrences. However, entire hierarchies are sometimes called *dimensions*. For example, the term *time dimension* can be used for the entire hierarchy built on the date attribute. Measures are sometimes called *variables, metrics, properties, attributes,* or *indicators.* In some models, dimensional attributes of hierarchies are called *levels* or *parameters.*

The information in a multidimensional cube is very difficult for users to manage because of its quantity, even if it is a concise version of the information stored to operational databases. If, for example, a store chain includes 50 stores selling 1000 items, and a specific data warehouse covers three-year-long transactions (approximately 1000 days), the number of potential events totals 50 × 1000 × 1000 = 5 × 107. Assuming that each store can sell only 10 percent of all the available items per day, the number of events totals 5 × 106. This is still too much data to be analyzed by users without relying on automatic tools.

You have essentially two ways to reduce the quantity of data and obtain useful information: *restriction* and *aggregation*. The cube metaphor offers an easy-to-use and intuitive way to understand both of these methods, as we will discuss in the following paragraphs.

1.5.1 Restriction

Restricting data means separating part of the data from a cube to mark out an analysis field. In relational algebra terminology, this is called making *selections* and/or *projections*.

The simplest type of selection is *data slicing,* shown in Figure 3. When you slice data, you decrease cube dimensionality by setting one or more dimensions to a specific value. For example, if you set one of the sales cube dimensions to a value, such as store='EverMore', this results in the set of events associated with the items sold in the EverMore store. According to the cube metaphor, this is simply a plane of cells—that is, a data slice that can be easily displayed in spreadsheets. In the store chain example given earlier, approximately 105 events still appear in your result. If you set two dimensions to a value, such as store='EverMore' and date='4/5/2008', this will result in all the different items sold in the EverMore store on April 5 (approximately 100 events). Graphically speaking, this information is stored at the intersection of two perpendicular planes resulting in a line. If you set all the dimensions to a particular value, you will define just one event that corresponds to a point in the threedimensional space of sales.

Dicing is a generalization of slicing. It poses some constraints on dimensional attributes to scale down the size of a cube. For example, you can select only the daily sales of the food items in April 2008 in Florida (Figure 3). In this way, if five stores are located in Florida and 50 food products are sold, the number of events to examine changes to 5 × 50 × 30 = 7500.

Finally, a *projection* can be referred to as a choice to keep just one subgroup of measures for every event and reject other measures.

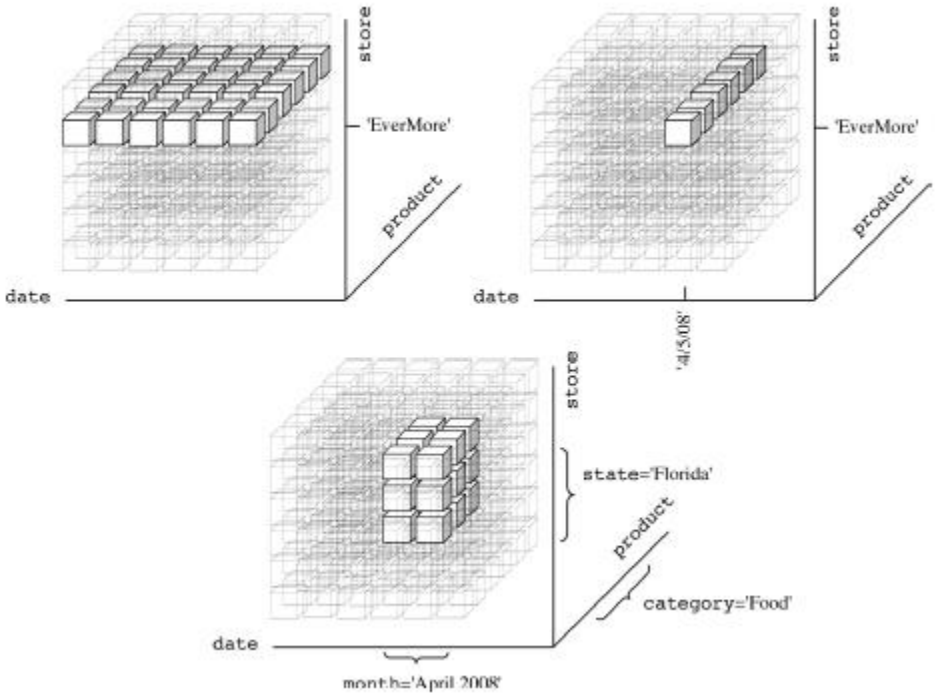


FIGURE 3 Slicing and dicing a three-dimensional cube

1.5.2 Aggregation

Aggregation plays a fundamental role in multidimensional databases. Assume, for example, that you want to analyze the items sold monthly for a three year period. According to the cube metaphor, this means that you need to sort all the cells related to the days of each month by product and store, and then merge them into one single macrocell. In the aggregate cube obtained in this way, the total number of events (that is, the number of macrocells) is 50 × 1000 × 36. This is because the granularity of the time dimensions does not depend on days any longer, but now depends on months, and 36 is the number of months in three years. Every aggregate event will then sum up the data available in the events it aggregates. In this example, the total amount of items sold per month and the total receipts are calculated by summing every single value of their measures (Figure 4). If you further aggregate along time, you can achieve just three events for every storeproduct combination: one for every year. When you completely aggregate along the time dimension, each store-product combination corresponds to one single event, which shows the total amount of items sold in a store over three years and the total amount of receipts.

FIGURE 4
Time hierarchy aggregation of the quantity of items sold per product in three stores. A dash shows that an event did not occur because no item was sold.

	EverMore	EvenMore	SmartMart
1/1/2007	–	–	–
1/2/2007	10	15	5
1/3/2007	20	–	5
.....	.....	.....	.....
1/1/2008	–	–	–
1/2/2008	15	10	20
1/3/2008	20	20	25
.....	.....	.....	.....
1/1/2009	–	–	–
1/2/2009	20	8	25
1/3/2009	20	12	20
.....	.....	.....	.....

	EverMore	EvenMore	SmartMart
January 2007	200	180	150
February 2007	180	150	120
March 2007	220	180	160
.....	.....	.....	.....
January 2008	350	220	200
February 2008	300	200	250
March 2008	310	180	300
.....	.....	.....	.....
January 2009	380	200	220
February 2009	310	200	250
March 2009	300	160	280
.....	.....	.....	.....

	EverMore	EvenMore	SmartMart
2007	2,400	2,000	1,600
2008	3,200	2,300	3,000
2009	3,400	2,200	3,200

	EverMore	EvenMore	SmartMart
Total	9,000	6,500	7,800

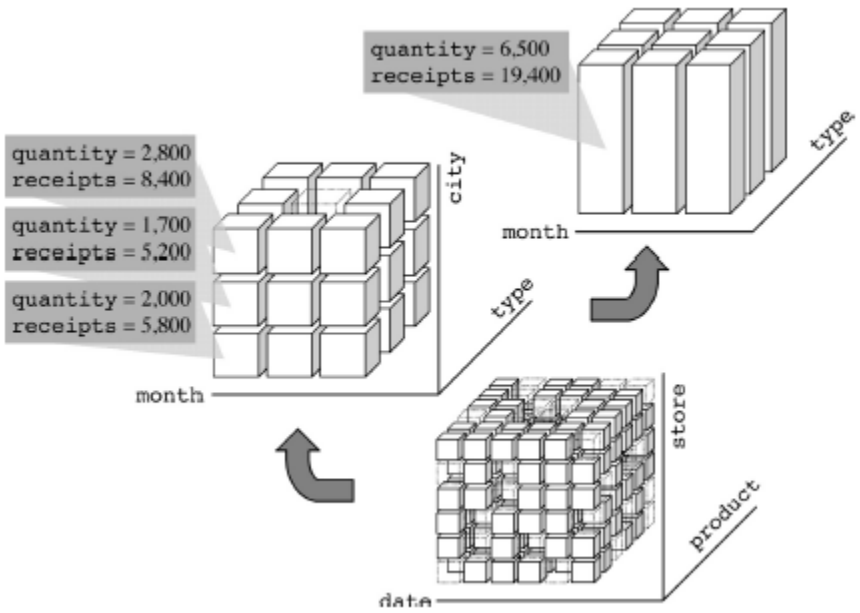


FIGURE 5 Two cube aggregation levels. Every macro-event measure value is a sum of its component event values.

You can aggregate along various dimensions at the same time. For example, Figure 5 shows that you can group sales by month, product type, and store city, and by month and product type. Moreover, selections and aggregations can be combined to carry out an analysis process targeted exactly to users' needs.

2. Accessing Data Warehouses

Analysis is the last level common to all data warehouse architecture types. After cleansing, integrating, and transforming data, you should determine how to get the best out of it in terms of information. The following sections show the best approaches for end users to query data warehouses: *reports*, *OLAP,* and *dashboards*. End users often use the information stored to a data warehouse as a starting point for additional business intelligence applications, such as what-if analyses and data mining. See Chapter 15 for more details on these advanced applications.

2.1 Reports

This approach is oriented to those users who need to have regular access to the information in an almost static way. For example, suppose a local health authority must send to its state offices monthly reports summing up information on patient admission costs. The layout of those reports has been predetermined and may vary only if changes are applied to current laws and regulations. Designers issue the queries to create reports with the desired layout and "freeze" all those in an application. In this way, end users can query current data whenever they need to.

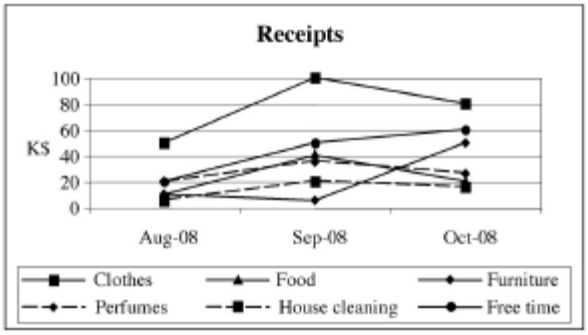
A *report* is defined by a query and a layout. A query generally implies a restriction and an aggregation of multidimensional data. For example, you can look for the monthly receipts during the last quarter for every product category. A layout can look like a table or a chart (diagrams, histograms, pies, and so on). Figure 7 shows a few examples of layouts for the receipts query.

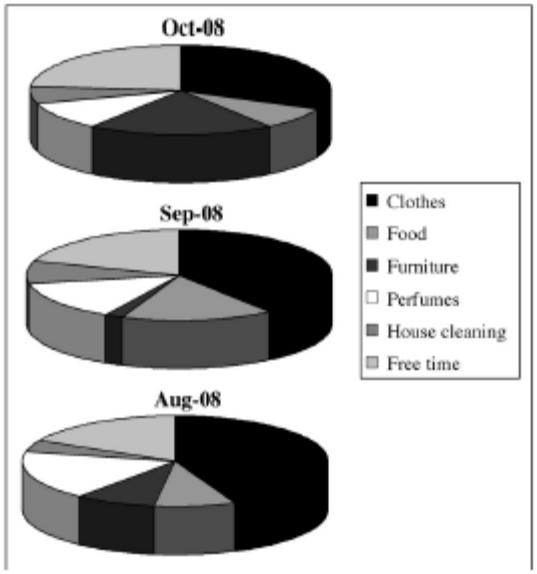
A reporting tool should be evaluated not only on the basis of comprehensive report layouts, but also on the basis of flexible report delivery systems. A report can be explicitly run by users or automatically and regularly sent to registered end users. For example, it can be sent via e-mail.

Keep in mind that reports existed long before data warehouse systems came to be. Reports have always been the main tool used by managers for evaluating and planning tasks since the invention of databases. However, adding data warehouses to the mix is beneficial to reports for two main reasons: First, they take advantage of reliable and correct results because the data summed up in reports is consistent and integrated. In addition, data warehouses expedite the reporting process because the architectural separation between transaction processing and analyses significantly improves performance.

FIGURE 7
Report layouts: table (top), line graph (middle), 3-D pie graphs (bottom)

Receipts (K$)	Oct. 2008	Sep. 2008	Aug. 2008
Clothes	80	100	50
Food	20	40	10
Furniture	50	5	10
Perfumes	25	35	20
House cleaning	15	20	5
Free time	60	50	20





2.2 OLAP

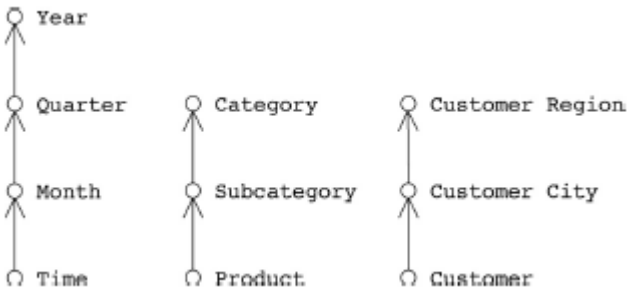
OLAP might be the main way to exploit information in a data warehouse. Surely it is the most popular one, and it gives end users, whose analysis needs are not easy to define beforehand, the opportunity to analyze and explore data interactively on the basis of the multidimensional model. While users of reporting tools essentially play a passive role, OLAP users are able to start a complex analysis session actively, where each step is the result of the outcome of preceding steps. Real-time properties of OLAP sessions, required in-depth knowledge of data, complex queries that can be issued, and design for users not familiar with IT make the tools in use play a crucial role. The GUI of these tools must be flexible, easy-to-use, and effective.

An OLAP session consists of a *navigation path* that corresponds to an analysis process for facts according to different viewpoints and at different detail levels. This path is turned into a sequence of queries, which are often not issued directly, but differentially expressed with reference to the previous query. The results of queries are multidimensional. Because we humans have a difficult time deciphering diagrams of more than three dimensions, OLAP tools typically use tables to display data, with multiple headers, colors, and other features to highlight data dimensions.

Every step of an analysis session is characterized by an *OLAP operator* that turns the latest query into a new one. The most common operators are roll-up, drill-down, slice-and-dice, pivot, drill-across, and drill-through. The figures included here show different operators, and were generated using the MicroStrategy Desktop front-end application in the MicroStrategy 8 tool suite. They are based on the V-Mall example, in which a large virtual mall sells items from its catalog via phone and the Internet. Figure 8 shows the attribute hierarchies relevant to the sales fact in V-Mall.

The *roll-up* operator causes an increase in data aggregation and removes a detail level from a hierarchy. For example, Figure 9 shows a query posed by a user that displays monthly revenues in 2005 and 2006 for every customer region. If you "roll it up," you remove the month detail to display quarterly total revenues per region. Rolling-up can also reduce the number of dimensions in your results if you remove all the hierarchy details. If you apply this principle to Figure 10, you can remove information on customers and display yearly total revenues per product category as you turn the three-dimensional table into a two-dimensional one. Figure 21 uses the cube metaphor to sketch a roll-up operation with and without a decrease in dimensions.

FIGURE 18 Attribute hierarchies in V-Mall; arrows show functional dependencies



Month	Metrics / Customer Region	Revenue: Northeast	Mid-Atlantic	Southeast	Central	South	Northwest	Southwest
Jan 2005		$160,155	$518,405	$81,381	$322,294	$98,001	$103,368	$298,730
Feb 2005		$170,777	$491,629	$90,399	$314,466	$91,222	$114,341	$373,645
Mar 2005		$200,434	$611,424	$129,102	$382,946	$123,038	$147,472	$351,602
Apr 2005		$194,811	$502,241	$93,654	$357,188	$90,663	$124,824	$304,416
May 2005		$169,998	$462,364	$117,780	$300,389	$79,999	$117,506	$311,123
Jun 2005		$202,477	$559,466	$109,979	$309,683	$115,318	$117,008	$373,526
Jul 2005		$194,490	$577,515	$105,099	$332,300	$92,730	$103,494	$369,380
Aug 2005		$203,085	$599,761	$118,805	$410,885	$119,178	$131,148	$384,555
Sep 2005		$241,992	$625,517	$122,261	$415,763	$75,655	$124,974	$364,651
Oct 2005		$217,477	$641,340	$137,025	$382,321	$89,670	$124,276	$337,489
Nov 2005		$238,004	$708,036	$156,525	$457,105	$116,478	$156,466	$386,399
Dec 2005		$273,721	$774,372	$154,130	$479,729	$110,113	$143,758	$414,983
Jan 2006		$215,786	$662,632	$125,236	$392,922	$91,791	$122,235	$343,027
Feb 2006		$253,128	$711,937	$123,725	$415,742	$97,309	$137,589	$391,277
Mar 2006		$253,564	$704,652	$135,180	$430,143	$112,459	$144,659	$406,956
Apr 2006		$255,382	$710,402	$126,717	$426,423	$113,233	$140,976	$395,924
May 2006		$231,766	$676,205	$130,981	$440,813	$107,277	$136,043	$377,349
Jun 2006		$290,534	$769,788	$123,743	$507,166	$125,631	$131,549	$439,321
Jul 2006		$247,683	$811,060	$145,955	$448,939	$113,683	$128,113	$415,251
Aug 2006		$252,313	$719,509	$125,944	$427,188	$108,987	$153,966	$421,310
Sep 2006		$288,772	$801,819	$148,023	$539,406	$112,784	$149,236	$419,878
Oct 2006		$307,610	$710,458	$163,254	$450,006	$105,218	$144,906	$440,856
Nov 2006		$284,671	$800,941	$157,117	$505,952	$118,552	$163,560	$470,591
Dec 2006		$310,775	$891,543	$170,207	$575,086	$120,228	$155,409	$534,680

Quarter	Metrics / Customer Region	Revenue: Northeast	Mid-Atlantic	Southeast	Central	South	Northwest	Southwest
2005 Q1		$531,366	$1,621,457	$290,882	$1,019,706	$312,261	$365,181	$1,023,977
2005 Q2		$567,286	$1,524,070	$321,413	$967,260	$285,981	$359,339	$989,065
2005 Q3		$639,567	$1,802,793	$346,165	$1,156,948	$287,563	$359,616	$1,118,587
2005 Q4		$729,202	$2,123,749	$448,589	$1,319,154	$325,269	$424,495	$1,138,871
2006 Q1		$722,478	$2,079,221	$384,143	$1,239,807	$301,559	$404,493	$1,141,260
2006 Q2		$777,651	$2,156,394	$381,441	$1,374,402	$346,141	$408,567	$1,212,593
2006 Q3		$788,768	$2,332,388	$419,923	$1,415,533	$335,455	$431,315	$1,256,439
2006 Q4		$903,056	$2,402,942	$490,578	$1,531,044	$343,998	$463,874	$1,446,127

FIGURE 19 Time hierarchy roll-up

Category	Year	Metrics / Customer Region	Revenue: Northeast	Mid-Atlantic	Southeast	Central	South	Northwest	Southwest
Books	2005		$416,183	$316,104	$36,517	$207,850	$137,502	$19,062	$187,368
	2006		$534,932	$401,908	$42,027	$239,806	$138,683	$22,655	$183,275
Electronics	2005		$1,860,172	$6,517,723	$1,226,825	$3,719,752	$915,633	$1,434,575	$3,625,191
	2006		$2,403,311	$8,253,620	$1,451,397	$4,631,259	$999,611	$1,615,848	$4,298,985
Movies	2005		$112,560	$138,611	$118,179	$153,556	$119,566	$27,060	$362,858
	2006		$148,785	$188,567	$147,445	$203,547	$145,434	$35,878	$463,470
Music	2005		$78,507	$99,631	$25,528	$383,911	$38,373	$27,933	$95,083
	2006		$104,025	$126,851	$35,215	$485,174	$43,424	$33,860	$110,680

Category	Year	Metrics	Revenue
Books	2005		$1,320,585
	2006		$1,563,287
Electronics	2005		$19,299,870
	2006		$23,654,030
Movies	2005		$1,032,391
	2006		$1,333,126
Music	2005		$748,966
	2006		$940,136

FIGURE 20 Roll-up removing customer hierarchy

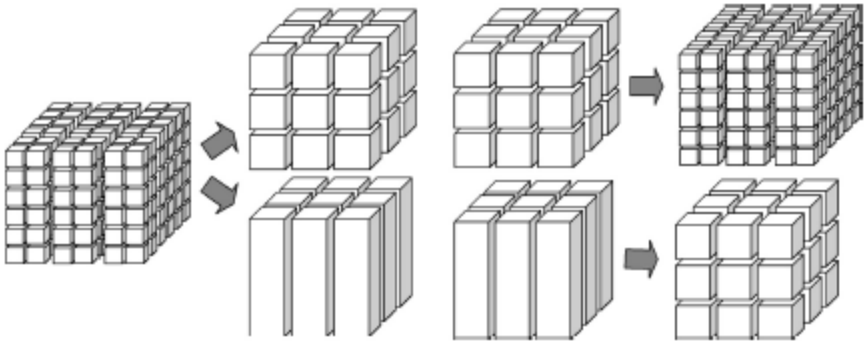


FIGURE 21 Rolling-up (left) and drilling-down (right) a cube

The *drill-down* operator is the complement to the roll-up operator. Figure 21 shows that it reduces data aggregation and adds a new detail level to a hierarchy. Figure 22 shows an example based on a bidimensional table. This table shows that the aggregation based on customer regions shifts to a new fine-grained aggregation based on customer cities.

Quarter	Metrics / Customer Region	Revenue: Northeast	Mid-Atlantic	Southeast	Central	South	Northwest	Southwest
2005 Q1		$531,366	$1,621,457	$290,882	$1,019,706	$312,261	$365,181	$1,023,977
2005 Q2		$567,286	$1,524,070	$321,413	$967,260	$285,981	$359,339	$989,065
2005 Q3		$639,567	$1,802,793	$346,165	$1,158,948	$287,563	$390,616	$1,118,587
2005 Q4		$720,202	$2,123,749	$448,580	$1,319,154	$325,269	$424,495	$1,138,871
2006 Q1		$722,478	$2,079,221	$384,143	$1,238,807	$301,559	$404,483	$1,141,260
2006 Q2		$777,651	$2,156,394	$381,441	$1,374,402	$346,141	$408,567	$1,212,593
2006 Q3		$788,768	$2,332,388	$419,923	$1,415,533	$335,455	$431,315	$1,256,439
2006 Q4		$903,056	$2,402,942	$490,578	$1,531,044	$343,998	$463,874	$1,446,127

Quarter	Metrics / Customer City	Revenue: Addison	Akron	Albany	Albert City	Alexandria	Allentown	Anderson	Annapolis	Arden	Arlington Heights	Arlington	Artesia	Ar
2005 Q1		$7,713	$30,140	$4,626	$6,686	$29,042	$4,579	$1,948	$40,066	$23,341	$10,481		$10,922	$1
2005 Q2		$15,903	$48,029	$8,959	$2,088	$17,590	$7,268	$2,416	$42,764	$21,026	$7,514	$1,695	$2,984	$
2005 Q3		$10,091	$30,510	$5,763	$11,380	$26,389	$10,195	$568	$51,650	$25,132	$10,784	$3,796	$8,701	$
2005 Q4		$12,425	$48,588	$10,939	$10,463	$28,016	$10,426	$3,412	$67,515	$28,398	$14,692		$9,975	$1
2006 Q1		$7,256	$26,183	$7,998	$5,603	$35,959	$10,273	$2,732	$50,121	$26,351	$7,276	$1,593	$6,208	$1
2006 Q2		$10,411	$49,540	$6,065	$5,670	$25,166	$6,992	$1,377	$68,198	$27,556	$16,755	$3,420	$6,656	$1
2006 Q3		$10,325	$38,414	$9,108	$7,760	$33,170	$16,978	$821	$69,858	$33,579	$10,235	$4,319	$7,636	$1
2006 Q4		$16,613	$49,133	$13,137	$10,637	$30,344	$13,875	$747	$48,305	$32,842	$13,311	$6,176	$9,191	$1

FIGURE 22 Drilling-down customer hierarchy

In Figure 23, the drill-down operator causes an increase in the number of table dimensions after adding customer region details.

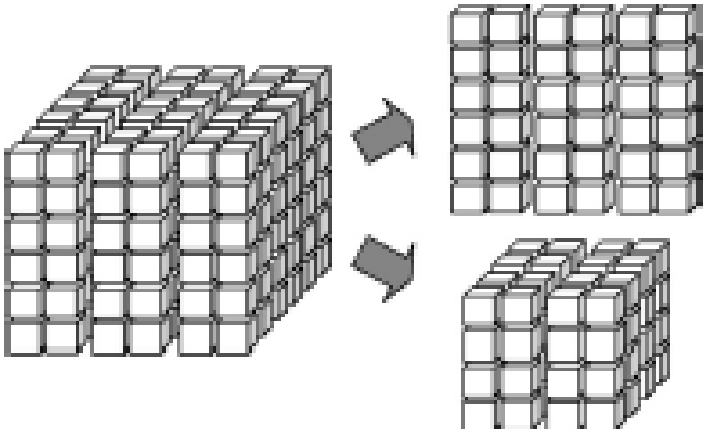
Category	Metrics / Year	Revenue 2005	2006
Books		$1,320,585	$1,563,287
Electronics		$19,299,870	$23,654,030
Movies		$1,032,391	$1,333,126
Music		$748,966	$940,136

Category	Metrics / Customer Region / Year	Revenue Northeast 2005	Northeast 2006	Mid-Atlantic 2005	Mid-Atlantic 2006	Southeast 2005	Southeast 2006	Central 2005	Central 2006	South 2005	South 2006	Nort 2
Books		$416,183	$534,932	$316,104	$401,908	$36,517	$42,027	$207,850	$239,806	$137,502	$138,683	$
Electronics		$1,860,172	$2,403,311	$6,517,723	$8,253,620	$1,226,825	$1,451,397	$3,719,752	$4,631,259	$915,633	$999,611	$1,4
Movies		$112,560	$148,785	$138,611	$188,567	$118,179	$147,445	$153,556	$203,547	$119,566	$145,434	$
Music		$78,507	$104,925	$99,631	$126,851	$25,528	$35,215	$383,911	$485,174	$38,373	$43,424	$

FIGURE 23 Drilling-down and adding a dimension

Slice-and-dice is one of the most abused terms in data warehouse literature because it can have many different meanings. A few authors use it generally to define the whole OLAP navigation process. Other authors use it to define selection and projection operations based on data. In compliance with section 1.5.1, we define *slicing* as an operation that reduces the number of cube dimensions after setting one of the dimensions to a specific value. *Dicing* is an operation that reduces the set of data being analyzed by a selection criterion (Figure 24). Figures 1-24 and 1-25 show a few examples of slicing and dicing.

FIGURE 24
Slicing (above) and dicing (below) a cube



Category	Year	Metrics / Customer Region	Revenue Northeast	Mid-Atlantic	Southeast	Central	South	Northwest	Southwest
Books	2005		$416,183	$316,104	$36,517	$207,850	$137,502	$19,062	$187,368
	2006		$534,932	$401,908	$42,027	$239,806	$138,683	$22,655	$183,275
Electronics	2005		$1,860,172	$6,517,723	$1,226,825	$3,719,752	$915,633	$1,434,575	$3,625,191
	2006		$2,403,311	$8,253,620	$1,451,397	$4,631,259	$999,611	$1,615,848	$4,298,985
Movies	2005		$112,560	$138,611	$118,179	$153,556	$119,566	$27,060	$362,858
	2006		$148,785	$188,567	$147,445	$203,547	$145,434	$35,878	$463,470
Music	2005		$78,507	$99,631	$25,528	$383,911	$38,373	$27,933	$95,083
	2006		$104,925	$126,851	$35,215	$485,174	$43,424	$33,860	$110,689

Report Filter (Local Filter)
Year = 2006

Category	Metrics / Customer Region	Revenue Northeast	Mid-Atlantic	Southeast	Central	South	Northwest	Sout
Books		$534,932	$401,908	$42,027	$239,806	$138,683	$22,655	$1
Electronics		$2,403,311	$8,253,620	$1,451,397	$4,631,259	$999,611	$1,615,848	$4,2
Movies		$148,785	$188,567	$147,445	$203,547	$145,434	$35,878	$4
Music		$104,925	$126,851	$35,215	$485,174	$43,424	$33,860	$1

FIGURE 25 Slicing based on the Year='2006' predicate

Subcategory \ Metrics / Customer City	Revenue: Addison	Akron	Albany	Albert City	Alexandria	Allentown	Anderson	Annapolis	Arden	Adir Heig
Art & Architecture	$253	$365	$1,506	$279	$268	$1,960		$407	$282	
Business	$198	$357	$719	$75	$134	$1,225	$8	$304	$184	
Literature	$92	$116	$277	$54	$66	$503		$137	$128	
Books - Miscellaneous	$216	$95	$830	$128	$73	$605	$4	$233	$71	
Science & Technology	$363	$943	$3,271	$578	$491	$2,547		$834	$366	
Sports & Health	$220	$153	$416	$165	$128	$1,476		$409	$204	
Audio Equipment	$4,782	$32,192	$2,982	$4,318	$36,458	$7,303	$4,430	$77,591	$17,397	$
Cameras	$7,671	$39,381	$5,810	$3,898	$12,045	$8,124	$405	$33,744	$11,840	$
Computers	$1,531	$14,810	$1,935	$3,022	$7,967	$3,354		$5,097	$7,087	$
Electronics - Miscellaneous	$4,607	$25,861	$5,289	$5,979	$25,484	$4,821	$130	$9,533	$11,923	$
TV's	$9,027	$34,635	$5,013	$3,500	$33,432	$4,930		$46,933	$32,527	$
Video Equipment	$7,422	$10,078	$5,540	$2,479	$5,500	$7,470	$540	$55,552	$35,723	$
Action	$108	$302	$204	$134	$256	$259	$25	$659	$117	
Comedy	$308	$520	$355	$195	$204	$261		$361	$216	
Drama	$424	$548	$225	$243	$313	$390		$511	$415	
Horror	$339	$196	$141	$195	$188	$372		$261	$184	
Kids / Family	$361	$277	$236	$213	$235	$460	$11	$744	$323	
Special Interests	$700	$670	$353	$289	$179	$695		$916	$298	
Alternative	$1,077	$236	$114	$484	$234	$139		$365	$173	$
Country	$1,169	$336	$310	$576	$206	$123	$43	$289	$298	
Music - Miscellaneous	$1,193	$364	$167	$494	$254	$281	$18	$410	$132	$
Pop	$541	$286	$231	$392	$139	$178	$26	$350	$173	
Rock	$1,184	$324	$202	$490	$161	$341	$14	$285	$127	
Soul / R&B	$760	$225	$182	$1,496	$223	$293	$23	$463	$159	$

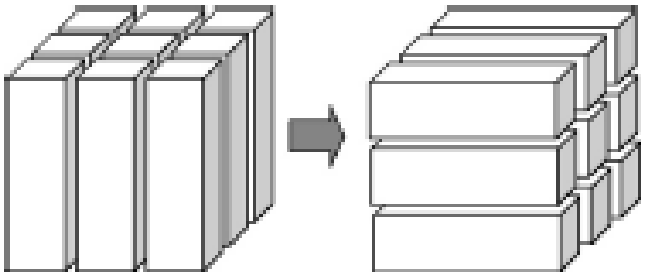
Report Filter (Local Filter):
[Year = 2006] And [Category = Electronics] And [Revenue > 80] And [[Customer Region] = Northwest]

Subcategory \ Metrics / Customer City	Revenue: Bellevue	Bothell	Caldwell	Cheyenne	Coulee City	E
Audio Equipment	$9,945	$23,211	$2,531	$2,347	$30,857	$
Cameras	$31,245	$5,613	$5,592	$1,240	$11,250	$
Computers	$12,751	$4,771	$1,443	$713	$8,181	
Electronics - Miscellaneous	$13,940	$29,072	$3,722	$2,444	$16,254	$
TV's	$40,652	$9,798	$3,990	$1,871	$10,846	$
Video Equipment	$36,423	$26,363	$3,543	$3,947	$5,480	$

FIGURE 26 Selection based on a complex predicate

The *pivot* operator implies a change in layouts. It aims at analyzing an individual group of information from a different viewpoint. According to the multidimensional metaphor, if you pivot data, you rotate your cube so that you can rearrange cells on the basis of a new perspective. In practice, you can highlight a different combination of dimensions (Figure 27). Figures 28 and 29 show a few examples of pivoted two-dimensional and three-dimensional tables.

FIGURE 27
Pivoting a cube



Category	Year \ Metrics	Revenue
Books	2005	$1,320,585
	2006	$1,563,287
Electronics	2005	$19,299,870
	2006	$23,654,030
Movies	2005	$1,032,301
	2006	$1,333,126
Music	2005	$748,966
	2006	$940,136

Category \ Metrics / Year	Revenue: 2005	2006
Books	$1,320,585	$1,563,287
Electronics	$19,299,870	$23,654,030
Movies	$1,032,301	$1,333,126
Music	$748,966	$940,136

FIGURE 28 Pivoting a two-dimensional table

Category	Year	Metrics / Customer Region	Revenue: Northeast	Mid-Atlantic	Southeast	Central	South	Northwest	Southwest
Books	2005		$416,183	$316,104	$36,517	$207,850	$137,502	$19,062	$187,368
	2006		$534,932	$401,908	$42,027	$239,806	$138,683	$22,655	$183,275
Electronics	2005		$1,860,172	$6,517,723	$1,226,825	$3,719,752	$915,633	$1,434,575	$3,625,191
	2006		$2,403,311	$8,253,620	$1,451,397	$4,631,259	$999,611	$1,615,848	$4,298,985
Movies	2005		$112,560	$138,611	$118,179	$153,556	$119,566	$27,060	$362,858
	2006		$148,785	$188,567	$147,445	$203,547	$145,434	$35,878	$463,470
Music	2005		$78,507	$99,631	$25,528	$383,911	$38,373	$27,933	$96,083
	2006		$104,925	$126,851	$35,215	$485,174	$43,424	$33,860	$110,689

Metrics / Customer Region / Year / Category	Revenue: Northeast 2005	Northeast 2006	Mid-Atlantic 2005	Mid-Atlantic 2006	Southeast 2005	Southeast 2006	Central 2005	Central 2006	South 2005	South 2006	Nort 2
Books	$416,183	$534,932	$316,104	$401,908	$36,517	$42,027	$207,850	$239,806	$137,502	$138,683	$
Electronics	$1,860,172	$2,403,311	$6,517,723	$8,253,620	$1,226,825	$1,451,397	$3,719,752	$4,631,259	$915,633	$999,611	$1,4
Movies	$112,560	$148,785	$138,611	$188,567	$118,179	$147,445	$153,556	$203,547	$119,566	$145,434	$
Music	$78,507	$104,925	$99,631	$126,851	$25,528	$35,215	$383,911	$485,174	$38,373	$43,424	$

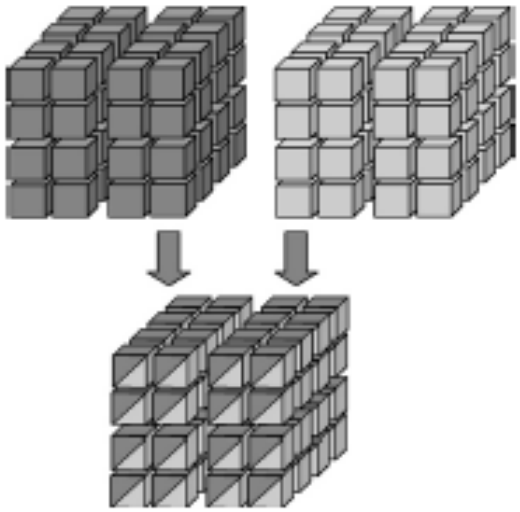
FIGURE 29 Pivoting a three-dimensional table

The term *drill-across* stands for the opportunity to create a link between two or more interrelated cubes in order to compare their data. For example, this applies if you calculate an expression involving measures from two cubes (Figure 30). Figure 31 shows an example in which a sales cube is drilled-across a promotions cube in order to compare revenues and discounts per quarter and product category.

Most OLAP tools can perform *drill-through* operations, though with varying effectiveness. This operation switches from multidimensional aggregate data in data marts to operational data in sources or in the reconciled layer.

In many applications, an intermediate approach between static reporting and OLAP is broadly used. This intermediate approach is called *semi-static reporting*. Even if a semi-static report focuses on a group of information previously set, it gives users some margin of freedom. Thanks to this margin, users can follow a limited set of navigation paths. For example, this applies when you can roll up just to a few hierarchy attributes. This solution is common, because it provides some unquestionable advantages. First, users need less skill to use data models and analysis tools than they need for OLAP. Second, this avoids the risk that occurs in OLAP of achieving inconsistent analysis results or incorrect ones because of any misuse of aggregation operators. Third, if you pose constraints on the analyses allowed, you will prevent users from unwillingly slowing down your system whenever they formulate demanding queries.

FIGURE 30
Drilling across two cubes



Category	Metrics / Quarter	Revenue 2005 Q1	2005 Q2	2005 Q3	2005 Q4	2006 Q1	2006 Q2	2006 Q3	2006 Q4
Books		$319,767	$313,339	$336,862	$350,617	$348,493	$397,849	$407,392	$419,563
Electronics		$4,448,112	$4,299,411	$4,918,673	$5,633,676	$5,411,499	$5,714,783	$5,999,174	$6,528,576
Movies		$228,108	$232,201	$264,471	$307,611	$299,531	$326,270	$334,143	$373,182
Music		$168,843	$169,462	$193,234	$217,427	$212,438	$228,289	$239,112	$260,298

Category	Quarter / Metrics	2005 Q1 Discount	2005 Q1 Revenue	2005 Q2 Discount	2005 Q2 Revenue	2005 Q3 Discount	2005 Q3 Revenue	2005 Q4 Discount	2005 Q4 Revenue	2006 Q1 Discount	Re
Books		$ 0	$319,767	$ 10,645	$313,339	$ 9,497	$336,862	$ 18,279	$350,617	$ 0	
Electronics		$ 0	$4,448,112	$ 150,366	$4,299,410	$ 143,905	$4,918,673	$ 302,884	$5,633,675	$ 0	$5
Movies		$ 0	$228,108	$ 9,025	$232,201	$ 7,948	$264,471	$ 16,649	$307,611	$ 0	
Music		$ 0	$168,843	$ 6,143	$169,462	$ 5,563	$193,234	$ 11,047	$217,427	$ 0	

FIGURE 31 Drilling across the sales cube (Revenue measure) and the promotions cube (Discount measure)

1.7.3 Dashboards

Dashboards are another method used for displaying information stored to a data warehouse. The term *dashboard* refers to a GUI that displays a limited amount of relevant data in a brief and easy-to-read format. Dashboards can provide a real-time overview of the trends for a specific phenomenon or for many phenomena that are strictly connected with each other. The term is a visual metaphor: the group of indicators in the GUI are displayed like a car dashboard. Dashboards are often used by senior managers who need a quick way to view information. However, to conduct and display very complex analyses of phenomena, dashboards must be matched with analysis tools.

Today, most software vendors offer dashboards for report creation and display. Figure 32 shows a dashboard created with MicroStrategy Dynamic Enterprise. The literature related to dashboard graphic design has also proven to be very rich, in particular in the scope of enterprises (Few, 2006).

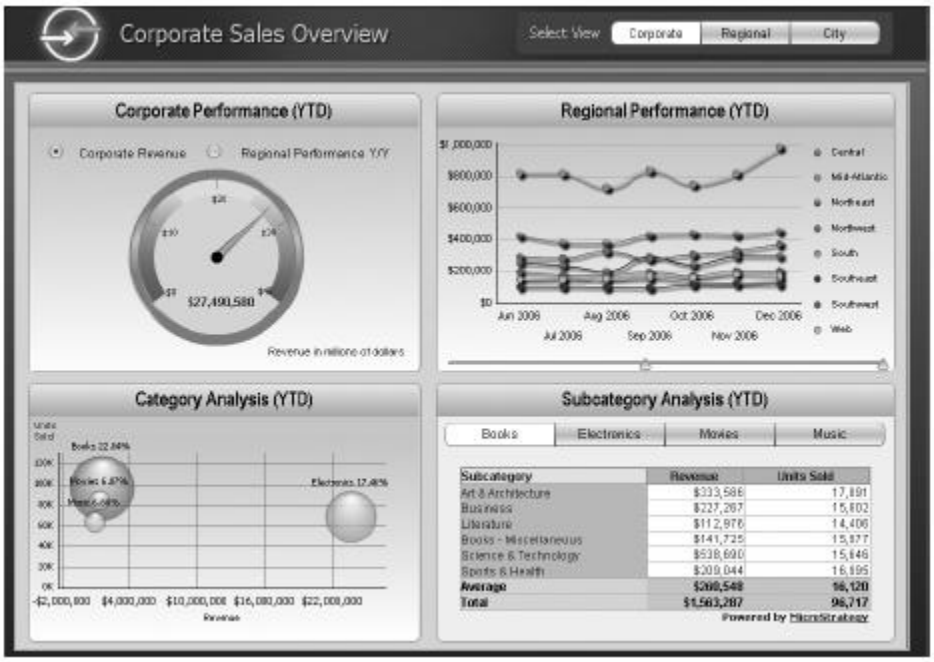


FIGURE 32 An example of dashboards

Keep in mind, however, that dashboards are nothing but performance indicators behind GUIs. Their effectiveness is due to a careful selection of the relevant measures, while using data warehouse information quality standards. For this reason, dashboards should be viewed as a sophisticated effective add-on to data warehouse systems, but not as the primary goal of data warehouse systems. In fact, the primary goal of data warehouse systems should always be to properly define a process to transform data into information.

1.8 ROLAP, MOLAP, and HOLAP

These three acronyms conceal three major approaches to implementing data warehouses, and they are related to the logical model used to represent data:

- *ROLAP* stands for *Relational OLAP*, an implementation based on relational DBMSs.
- *MOLAP* stands for *Multidimensional OLAP*, an implementation based on multidimensional DBMSs.
- *HOLAP* stands for *Hybrid OLAP*, an implementation using both relational and multidimensional techniques.

The idea of adopting the relational technology to store data to a data warehouse has a solid foundation if you consider the huge amount of literature written about the relational model, the broadly available corporate experience with relational database usage and management, and the top performance and flexibility standards of relational DBMSs (RDBMSs). The expressive power of the relational model, however, does not include the concepts of dimension, measure, and hierarchy, so you must create specific types of schemata so that you can represent the multidimensional model in terms of basic relational elements such as attributes, relations, and integrity constraints. This task is mainly performed by the well-known *star schema*. See Chapter 8 for more details on star schemata and star schema variants.

The main problem with ROLAP implementations results from the performance hit caused by costly join operations between large tables. To reduce the number of joins, one of the key concepts of ROLAP is *denormalization*—a conscious breach in the third normal form oriented to performance maximization. To minimize execution costs, the other key word is *redundancy,* which is the result of the materialization of some derived tables (*views*) that store aggregate data used for typical OLAP queries.