

1 CHAPTER Introduction to Data Warehousing I

nformation these However, assets the assets must availability be are properly immensely of too stored much valuable and data readily to makes any accessible enterprise, the extraction when and of because they the are most

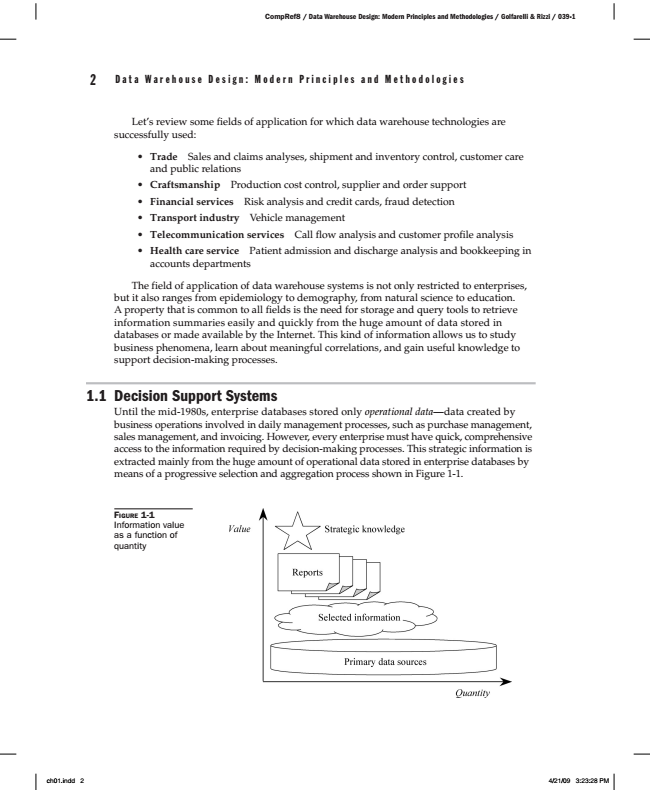
of needed. this,

important information difficult, if not impossible. View results from any Google search, and you’ll see that the data = information equation is not always correct—that is, too much data is simply too much.

Data warehousing is a phenomenon that grew from the huge amount of electronic data stored in recent years and from the urgent need to use that data to accomplish goals that go beyond the routine tasks linked to daily processing. In a typical scenario, a large corporation has many branches, and senior managers need to quantify and evaluate how each branch contributes to the global business performance. The corporate database stores detailed data on the tasks performed by branches. To meet the managers’ needs, tailor-made queries can be issued to retrieve the required data. In order for this process to work, database administrators must first formulate the desired query (typically an aggregate SQL query) after closely studying database catalogs. Then the query is processed. This can take a few hours because of the huge amount of data, the query complexity, and the concurrent effects of other regular workload queries on data. Finally, a report is generated and passed to senior managers in the form of a spreadsheet.

Many years ago, database designers realized that such an approach is hardly feasible, because it is very demanding in terms of time and resources, and it does not always achieve the desired results. Moreover, a mix of analytical queries with transactional routine queries inevitably slows down the system, and this does not meet the needs of users of either type of query. Today’s advanced data warehousing processes separate online analytical processing (OLAP) from online transactional processing (OLTP) by creating a new information repository that integrates basic data from various sources, properly arranges data formats, and then makes data available for analysis and evaluation aimed at planning and decision-making processes (Lechtenbörger, 2001).

1



2 D a t a W a r e h o u s e D e s i g n : M o d e r n P r i n c i p l e s a n d M e t h o d o l o g i e s

Let’s review some fields of application for which data warehouse technologies are successfully used:

• Trade Sales and claims analyses, shipment and inventory control, customer care and public relations

• Craftsmanship Production cost control, supplier and order support

• Financial services Risk analysis and credit cards, fraud detection

• Transport industry Vehicle management

• Telecommunication services Call flow analysis and customer profile analysis

• Health care service Patient admission and discharge analysis and bookkeeping in accounts departments

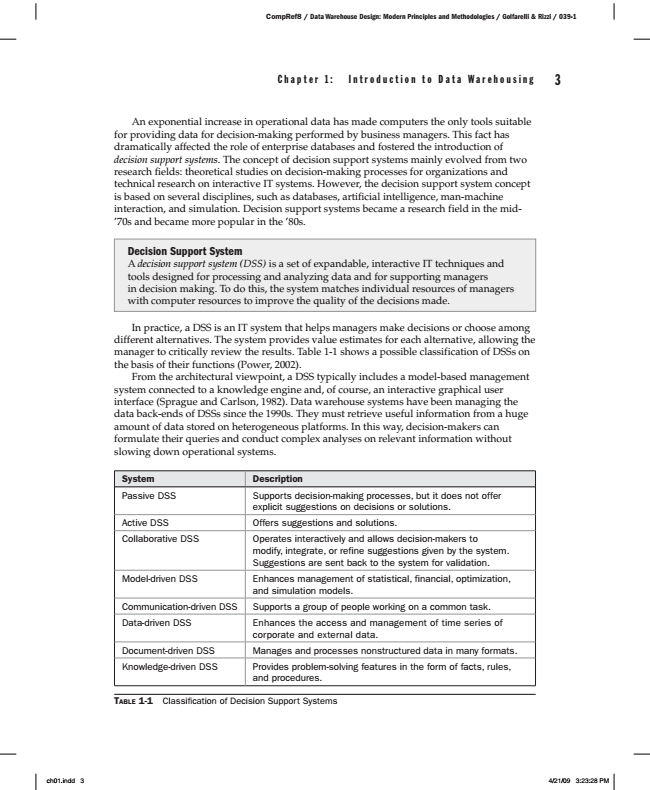
The field of application of data warehouse systems is not only restricted to enterprises, but it also ranges from epidemiology to demography, from natural science to education. A property that is common to all fields is the need for storage and query tools to retrieve information summaries easily and quickly from the huge amount of data stored in databases or made available by the Internet. This kind of information allows us to study business phenomena, learn about meaningful correlations, and gain useful knowledge to support decision-making processes.

1.1 Decision Support Systems

Until the mid-1980s, enterprise databases stored only operational data—data created by business operations involved in daily management processes, such as purchase management, sales management, and invoicing. However, every enterprise must have quick, comprehensive access to the information required by decision-making processes. This strategic information is extracted mainly from the huge amount of operational data stored in enterprise databases by means of a progressive selection and aggregation process shown in Figure 1-1.

1-1 F

IGURE Information value as a function of quantity



C h a p t e r 1 : I n t r o d u c t i o n t o D a t a W a r e h o u s i n g 3

An exponential increase in operational data has made computers the only tools suitable for providing data for decision-making performed by business managers. This fact has dramatically affected the role of enterprise databases and fostered the introduction of decision support systems. The concept of decision support systems mainly evolved from two research fields: theoretical studies on decision-making processes for organizations and technical research on interactive IT systems. However, the decision support system concept is based on several disciplines, such as databases, artificial intelligence, man-machine interaction, and simulation. Decision support systems became a research field in the mid- ’70s and became more popular in the ’80s.

Decision Support System A decision support system (DSS) is a set of expandable, interactive IT techniques and tools designed for processing and analyzing data and for supporting managers in decision making. To do this, the system matches individual resources of managers with computer resources to improve the quality of the decisions made.

In practice, a DSS is an IT system that helps managers make decisions or choose among different alternatives. The system provides value estimates for each alternative, allowing the manager to critically review the results. Table 1-1 shows a possible classification of DSSs on the basis of their functions (Power, 2002).

From the architectural viewpoint, a DSS typically includes a model-based management system connected to a knowledge engine and, of course, an interactive graphical user interface (Sprague and Carlson, 1982). Data warehouse systems have been managing the data back-ends of DSSs since the 1990s. They must retrieve useful information from a huge amount of data stored on heterogeneous platforms. In this way, decision-makers can formulate their queries and conduct complex analyses on relevant information without slowing down operational systems.

System Description Passive DSS Supports decision-making processes, but it does not offer

explicit suggestions on decisions or solutions. Active DSS Offers suggestions and solutions. Collaborative DSS Operates interactively and allows decision-makers to

modify, integrate, or refine suggestions given by the system. Suggestions are sent back to the system for validation. Model-driven DSS Enhances management of statistical, financial, optimization,

and simulation models. Communication-driven DSS Supports a group of people working on a common task. Data-driven DSS Enhances the access and management of time series of

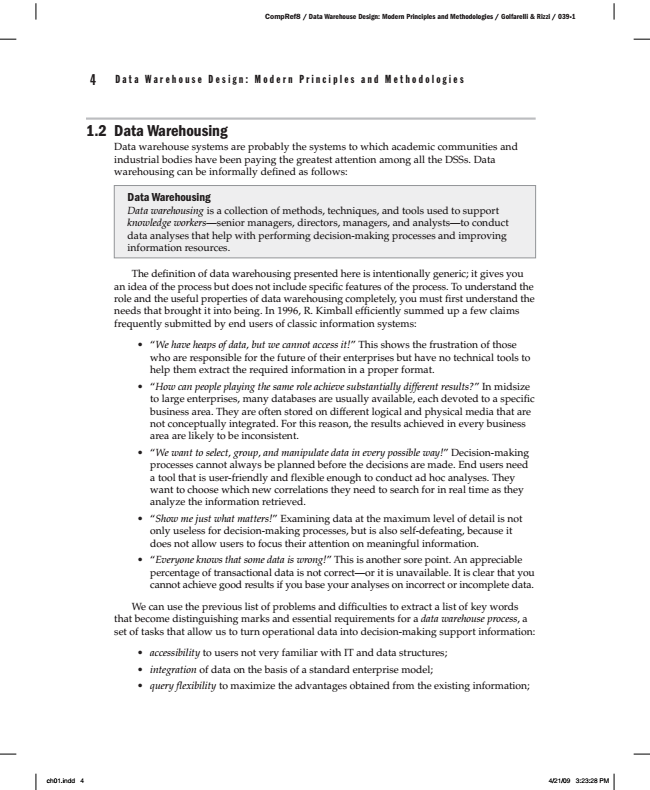
corporate and external data. Document-driven DSS Manages and processes nonstructured data in many formats. Knowledge-driven DSS Provides problem-solving features in the form of facts, rules,

and procedures.

T

ABLE

1-1 Classification of Decision Support Systems



4 D a t a W a r e h o u s e D e s i g n : M o d e r n P r i n c i p l e s a n d M e t h o d o l o g i e s

1.2 Data Warehousing

Data warehouse systems are probably the systems to which academic communities and industrial bodies have been paying the greatest attention among all the DSSs. Data warehousing can be informally defined as follows:

Data Warehousing Data warehousing is a collection of methods, techniques, and tools used to support knowledge workers—senior managers, directors, managers, and analysts—to conduct data analyses that help with performing decision-making processes and improving information resources.

The definition of data warehousing presented here is intentionally generic; it gives you an idea of the process but does not include specific features of the process. To understand the role and the useful properties of data warehousing completely, you must first understand the needs that brought it into being. In 1996, R. Kimball efficiently summed up a few claims frequently submitted by end users of classic information systems:

• “We have heaps of data, but we cannot access it!” This shows the frustration of those who are responsible for the future of their enterprises but have no technical tools to help them extract the required information in a proper format.

• “How can people playing the same role achieve substantially different results?” In midsize to large enterprises, many databases are usually available, each devoted to a specific business area. They are often stored on different logical and physical media that are not conceptually integrated. For this reason, the results achieved in every business area are likely to be inconsistent.

• “We want to select, group, and manipulate data in every possible way!” Decision-making processes cannot always be planned before the decisions are made. End users need a tool that is user-friendly and flexible enough to conduct ad hoc analyses. They want to choose which new correlations they need to search for in real time as they analyze the information retrieved.

• “Show me just what matters!” Examining data at the maximum level of detail is not only useless for decision-making processes, but is also self-defeating, because it does not allow users to focus their attention on meaningful information.

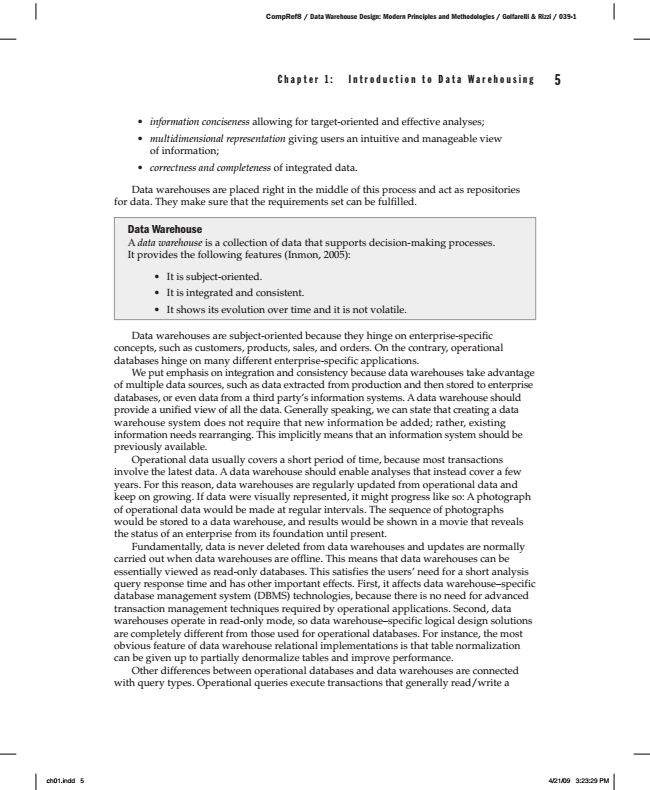
• “Everyone knows that some data is wrong!” This is another sore point. An appreciable percentage of transactional data is not correct—or it is unavailable. It is clear that you cannot achieve good results if you base your analyses on incorrect or incomplete data.

We can use the previous list of problems and difficulties to extract a list of key words that become distinguishing marks and essential requirements for a data warehouse process, a set of tasks that allow us to turn operational data into decision-making support information:

• accessibility to users not very familiar with IT and data structures;

• integration of data on the basis of a standard enterprise model;

• query flexibility to maximize the advantages obtained from the existing information;



C h a p t e r 1 : I n t r o d u c t i o n t o D a t a W a r e h o u s i n g 5

• information conciseness allowing for target-oriented and effective analyses;

• multidimensional representation giving users an intuitive and manageable view of information;

*• correctness and completeness of integrated data.*

Data warehouses are placed right in the middle of this process and act as repositories for data. They make sure that the requirements set can be fulfilled.

Data Warehouse A data warehouse is a collection of data that supports decision-making processes. It provides the following features (Inmon, 2005):

• It is subject-oriented.

• It is integrated and consistent.

• It shows its evolution over time and it is not volatile.

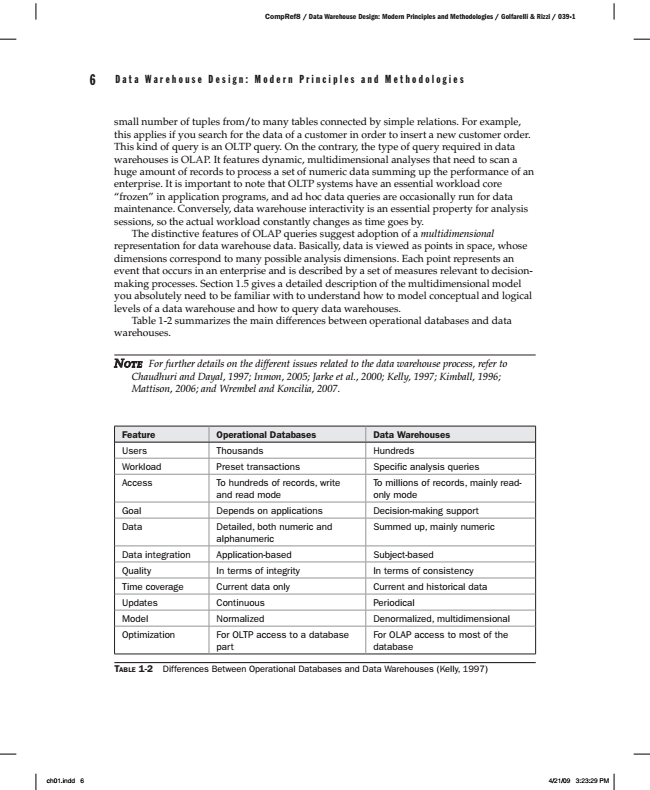
Data warehouses are subject-oriented because they hinge on enterprise-specific concepts, such as customers, products, sales, and orders. On the contrary, operational databases hinge on many different enterprise-specific applications.

We put emphasis on integration and consistency because data warehouses take advantage of multiple data sources, such as data extracted from production and then stored to enterprise databases, or even data from a third party’s information systems. A data warehouse should provide a unified view of all the data. Generally speaking, we can state that creating a data warehouse system does not require that new information be added; rather, existing information needs rearranging. This implicitly means that an information system should be previously available.

Operational data usually covers a short period of time, because most transactions involve the latest data. A data warehouse should enable analyses that instead cover a few years. For this reason, data warehouses are regularly updated from operational data and keep on growing. If data were visually represented, it might progress like so: A photograph of operational data would be made at regular intervals. The sequence of photographs would be stored to a data warehouse, and results would be shown in a movie that reveals the status of an enterprise from its foundation until present.

Fundamentally, data is never deleted from data warehouses and updates are normally carried out when data warehouses are offline. This means that data warehouses can be essentially viewed as read-only databases. This satisfies the users’ need for a short analysis query response time and has other important effects. First, it affects data warehouse–specific database management system (DBMS) technologies, because there is no need for advanced transaction management techniques required by operational applications. Second, data warehouses operate in read-only mode, so data warehouse–specific logical design solutions are completely different from those used for operational databases. For instance, the most obvious feature of data warehouse relational implementations is that table normalization can be given up to partially denormalize tables and improve performance.

Other differences between operational databases and data warehouses are connected with query types. Operational queries execute transactions that generally read/write a



6 D a t a W a r e h o u s e D e s i g n : M o d e r n P r i n c i p l e s a n d M e t h o d o l o g i e s

small number of tuples from/to many tables connected by simple relations. For example, this applies if you search for the data of a customer in order to insert a new customer order. This kind of query is an OLTP query. On the contrary, the type of query required in data warehouses is OLAP. It features dynamic, multidimensional analyses that need to scan a huge amount of records to process a set of numeric data summing up the performance of an enterprise. It is important to note that OLTP systems have an essential workload core “frozen” in application programs, and ad hoc data queries are occasionally run for data maintenance. Conversely, data warehouse interactivity is an essential property for analysis sessions, so the actual workload constantly changes as time goes by.

The distinctive features of OLAP queries suggest adoption of a multidimensional representation for data warehouse data. Basically, data is viewed as points in space, whose dimensions correspond to many possible analysis dimensions. Each point represents an event that occurs in an enterprise and is described by a set of measures relevant to decision- making processes. Section 1.5 gives a detailed description of the multidimensional model you absolutely need to be familiar with to understand how to model conceptual and logical levels of a data warehouse and how to query data warehouses.

Table 1-2 summarizes the main differences between operational databases and data warehouses.

***N***

*OTE For further details on the different issues related to the data warehouse process, refer to Chaudhuri and Dayal, 1997; Inmon, 2005; Jarke et al., 2000; Kelly, 1997; Kimball, 1996; Mattison, 2006; and Wrembel and Koncilia, 2007.*

Feature Operational Databases Data Warehouses Users Thousands Hundreds Workload Preset transactions Specific analysis queries Access To hundreds of records, write

and read mode

T

ABLE

To millions of records, mainly read- only mode Goal Depends on applications Decision-making support Data Detailed, both numeric and

alphanumeric

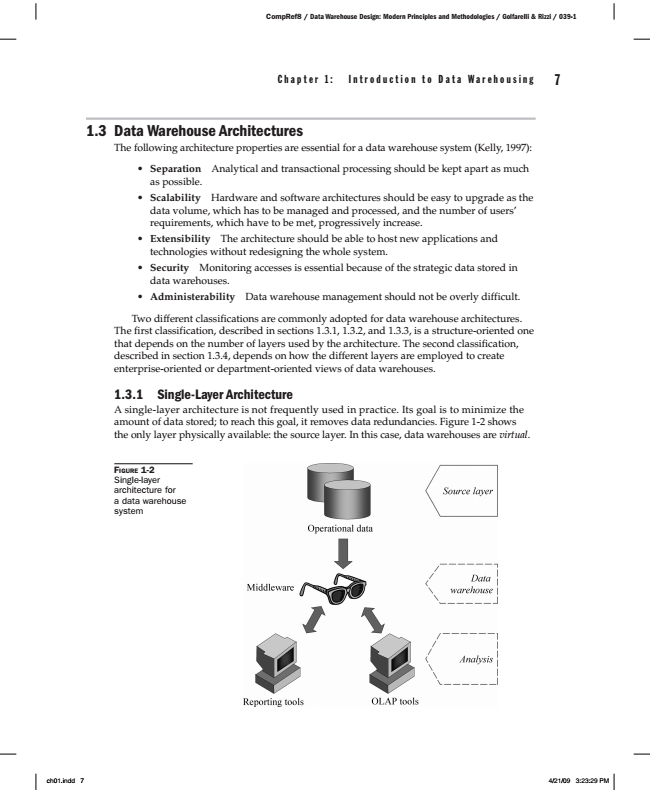
Summed up, mainly numeric

Data integration Application-based Subject-based Quality In terms of integrity In terms of consistency Time coverage Current data only Current and historical data Updates Continuous Periodical Model Normalized Denormalized, multidimensional Optimization For OLTP access to a database

part

For OLAP access to most of the database

1-2 Differences Between Operational Databases and Data Warehouses (Kelly, 1997)



C h a p t e r 1 : I n t r o d u c t i o n t o D a t a W a r e h o u s i n g 7

1.3 Data Warehouse Architectures

The following architecture properties are essential for a data warehouse system (Kelly, 1997):

• Separation Analytical and transactional processing should be kept apart as much as possible.

• Scalability Hardware and software architectures should be easy to upgrade as the data volume, which has to be managed and processed, and the number of users’ requirements, which have to be met, progressively increase.

• Extensibility The architecture should be able to host new applications and technologies without redesigning the whole system.

• Security Monitoring accesses is essential because of the strategic data stored in data warehouses.

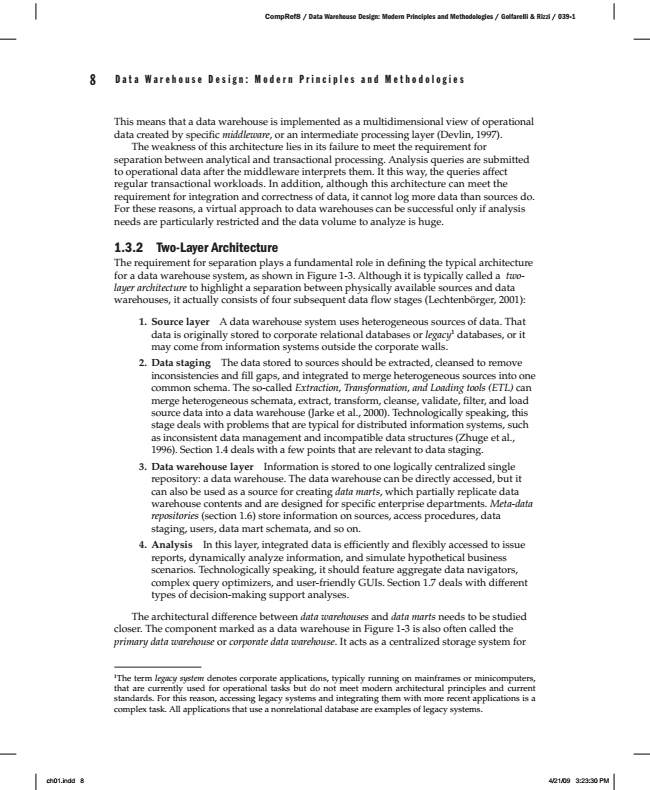
• Administerability Data warehouse management should not be overly difficult.

Two different classifications are commonly adopted for data warehouse architectures. The first classification, described in sections 1.3.1, 1.3.2, and 1.3.3, is a structure-oriented one that depends on the number of layers used by the architecture. The second classification, described in section 1.3.4, depends on how the different layers are employed to create enterprise-oriented or department-oriented views of data warehouses.

1.3.1 Single-Layer Architecture A single-layer architecture is not frequently used in practice. Its goal is to minimize the amount of data stored; to reach this goal, it removes data redundancies. Figure 1-2 shows the only layer physically available: the source layer. In this case, data warehouses are virtual.

1-2 F

IGURE Single-layer architecture for a data warehouse system



8 D a t a W a r e h o u s e D e s i g n : M o d e r n P r i n c i p l e s a n d M e t h o d o l o g i e s

This means that a data warehouse is implemented as a multidimensional view of operational data created by specific middleware, or an intermediate processing layer (Devlin, 1997). The weakness of this architecture lies in its failure to meet the requirement for separation between analytical and transactional processing. Analysis queries are submitted to operational data after the middleware interprets them. It this way, the queries affect regular transactional workloads. In addition, although this architecture can meet the requirement for integration and correctness of data, it cannot log more data than sources do. For these reasons, a virtual approach to data warehouses can be successful only if analysis needs are particularly restricted and the data volume to analyze is huge.

1.3.2 Two-Layer Architecture The requirement for separation plays a fundamental role in defining the typical architecture for a data warehouse system, as shown in Figure 1-3. Although it is typically called a two- layer architecture to highlight a separation between physically available sources and data warehouses, it actually consists of four subsequent data flow stages (Lechtenbörger, 2001):

1. Source layer A data warehouse system uses heterogeneous sources of data. That data is originally stored to corporate relational databases or legacy1 databases, or it may come from information systems outside the corporate walls. 2. Data staging The data stored to sources should be extracted, cleansed to remove

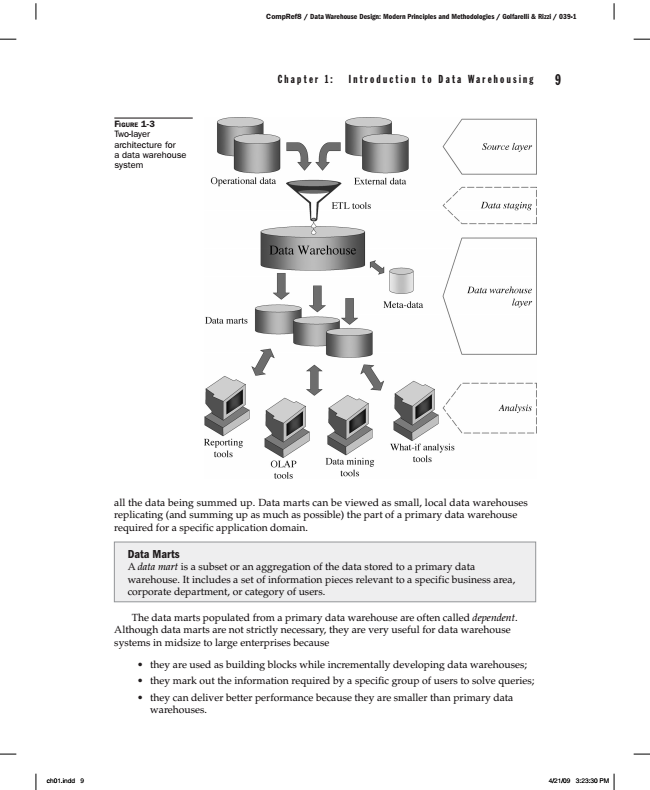
inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one common schema. The so-called Extraction, Transformation, and Loading tools (ETL) can merge heterogeneous schemata, extract, transform, cleanse, validate, filter, and load source data into a data warehouse (Jarke et al., 2000). Technologically speaking, this stage deals with problems that are typical for distributed information systems, such as inconsistent data management and incompatible data structures (Zhuge et al., 1996). Section 1.4 deals with a few points that are relevant to data staging. 3. Data warehouse layer Information is stored to one logically centralized single

repository: a data warehouse. The data warehouse can be directly accessed, but it can also be used as a source for creating data marts, which partially replicate data warehouse contents and are designed for specific enterprise departments. Meta-data repositories (section 1.6) store information on sources, access procedures, data staging, users, data mart schemata, and so on. 4. Analysis In this layer, integrated data is efficiently and flexibly accessed to issue

reports, dynamically analyze information, and simulate hypothetical business scenarios. Technologically speaking, it should feature aggregate data navigators, complex query optimizers, and user-friendly GUIs. Section 1.7 deals with different types of decision-making support analyses.

The architectural difference between data warehouses and data marts needs to be studied closer. The component marked as a data warehouse in Figure 1-3 is also often called the primary data warehouse or corporate data warehouse. It acts as a centralized storage system for

1The term legacy system denotes corporate applications, typically running on mainframes or minicomputers, that are currently used for operational tasks but do not meet modern architectural principles and current standards. For this reason, accessing legacy systems and integrating them with more recent applications is a complex task. All applications that use a nonrelational database are examples of legacy systems.



C h a p t e r 1 : I n t r o d u c t i o n t o D a t a W a r e h o u s i n g 9

F

IGURE

1-3 Two-layer architecture for a data warehouse system

all the data being summed up. Data marts can be viewed as small, local data warehouses replicating (and summing up as much as possible) the part of a primary data warehouse required for a specific application domain.

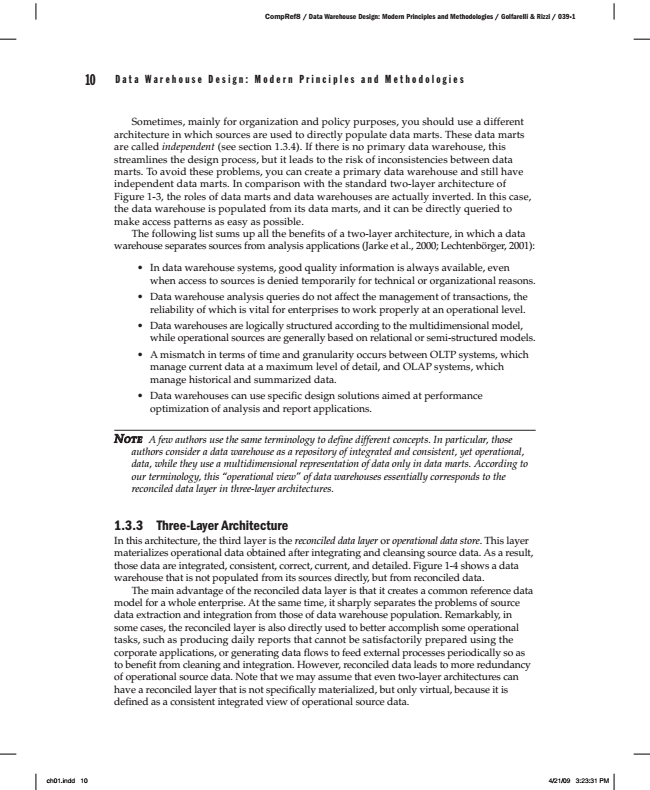
Data Marts A data mart is a subset or an aggregation of the data stored to a primary data warehouse. It includes a set of information pieces relevant to a specific business area, corporate department, or category of users.

The data marts populated from a primary data warehouse are often called dependent. Although data marts are not strictly necessary, they are very useful for data warehouse systems in midsize to large enterprises because

• they are used as building blocks while incrementally developing data warehouses;

• they mark out the information required by a specific group of users to solve queries;

• they can deliver better performance because they are smaller than primary data warehouses.



10 D a t a W a r e h o u s e D e s i g n : M o d e r n P r i n c i p l e s a n d M e t h o d o l o g i e s

Sometimes, mainly for organization and policy purposes, you should use a different architecture in which sources are used to directly populate data marts. These data marts are called independent (see section 1.3.4). If there is no primary data warehouse, this streamlines the design process, but it leads to the risk of inconsistencies between data marts. To avoid these problems, you can create a primary data warehouse and still have independent data marts. In comparison with the standard two-layer architecture of Figure 1-3, the roles of data marts and data warehouses are actually inverted. In this case, the data warehouse is populated from its data marts, and it can be directly queried to make access patterns as easy as possible.

The following list sums up all the benefits of a two-layer architecture, in which a data warehouse separates sources from analysis applications (Jarke et al., 2000; Lechtenbörger, 2001):

• In data warehouse systems, good quality information is always available, even when access to sources is denied temporarily for technical or organizational reasons.

• Data warehouse analysis queries do not affect the management of transactions, the reliability of which is vital for enterprises to work properly at an operational level.

• Data warehouses are logically structured according to the multidimensional model, while operational sources are generally based on relational or semi-structured models.

• A mismatch in terms of time and granularity occurs between OLTP systems, which manage current data at a maximum level of detail, and OLAP systems, which manage historical and summarized data.

• Data warehouses can use specific design solutions aimed at performance optimization of analysis and report applications.

***N***

*OTE A few authors use the same terminology to define different concepts. In particular, those*

*authors consider a data warehouse as a repository of integrated and consistent, yet operational, data, while they use a multidimensional representation of data only in data marts. According to our terminology, this “operational view” of data warehouses essentially corresponds to the reconciled data layer in three-layer architectures.*

1.3.3 Three-Layer Architecture In this architecture, the third layer is the reconciled data layer or operational data store. This layer materializes operational data obtained after integrating and cleansing source data. As a result, those data are integrated, consistent, correct, current, and detailed. Figure 1-4 shows a data warehouse that is not populated from its sources directly, but from reconciled data.

The main advantage of the reconciled data layer is that it creates a common reference data model for a whole enterprise. At the same time, it sharply separates the problems of source data extraction and integration from those of data warehouse population. Remarkably, in some cases, the reconciled layer is also directly used to better accomplish some operational tasks, such as producing daily reports that cannot be satisfactorily prepared using the corporate applications, or generating data flows to feed external processes periodically so as to benefit from cleaning and integration. However, reconciled data leads to more redundancy of operational source data. Note that we may assume that even two-layer architectures can have a reconciled layer that is not specifically materialized, but only virtual, because it is defined as a consistent integrated view of operational source data.