



Dipartimento di Scienze Economiche, Matematiche e Statistiche

Università degli Studi di Foggia

**Data Warehouse Design and Management:
Theory and Practice**

**Crescenzo Gallo, Michelangelo De Bonis e
Michele Perilli**

Quaderno n. 07/2010

“Esemplare fuori commercio per il deposito legale agli effetti della legge 15 aprile 2004 n. 106”
Quaderno riprodotto dal Dipartimento di Scienze Economiche, Matematiche e Statistiche
nel mese di ottobre 2010 e depositato ai sensi di legge.

Authors only are responsible for the content of this reprint.

Data Warehouse Design and Management: Theory and Practice

Crescenzo Gallo, Michelangelo De Bonis, Michele Perilli

IEEE MEMBERS

c.gallo@ieee.org, m.debonis@ieee.org, m.perilli@unifg.it

DIPARTIMENTO DI SCIENZE ECONOMICHE, MATEMATICHE E STATISTICHE

UNIVERSITÀ DI FOGGIA

Largo Papa Giovanni Paolo II, 1 - 71121 Foggia, Italy

Abstract—The need to store data and information permanently, for their reuse in later stages, is a very relevant problem in the modern world and now affects a large number of people and economic agents. The storage and subsequent use of data can indeed be a valuable source for decision making or to increase commercial activity. The next step to data storage is the efficient and effective use of information, particularly through the Business Intelligence, at whose base is just the implementation of a Data Warehouse. In the present paper we will analyze Data Warehouses with their theoretical models, and illustrate a practical implementation in a specific case study on a pharmaceutical distribution company.

Index Terms—Data warehouse, database, data model.

MSC (2000)— 68P15; 68P20, 68P05.

I. INTRODUCTION

Data storage and information retrieval is a very important topic nowadays and affects a large number of people and economic agents, being a valuable source for decision making or increasing business. Besides to data storage, the efficient and effective use of information is particularly important. This is typically done with Business Intelligence (BI) [1], at whose base is the implementation of a Data Warehouse or Data Mart.

BI provides a significant economic value to companies, helping them in practice to gain a better understanding of themselves. More generally, it refers to the competence, technology and applications involved in bringing this understanding to light. It can be defined as the ability to grasp the interrelationship of facts in order to direct action toward a desired goal. BI is not only concentrated in technological terms. It is also important in understanding the relationships between different aspects of the company, so that you can drive towards specific objectives such as increasing the market share and improving customer satisfaction. What is utmost important is that BI is crucial in supporting decisions.

This type of solution is due to the fact that companies are drowning in data that record in operational databases: payroll data, financial data, customer data, vendor data, and so on.

These databases are typically tuned for each operation, such as retrieving a single customer order, or for specific batch jobs, such as processing payroll at the end of each month. These databases are not designed to communicate with one another, allowing users to explore data in an unusual way, or to provide high level summary data at once.

BI pulls out all the data together and puts them in the report. Data may seem unrelated, but all the activity is in some way almost always related. BI does not generate new data, it simply makes it easier to explore the relationships between the data that would be overlooked by decision makers. Effectively and efficiently storing data in a data warehouse is so a “sine qua non” condition for supporting decisions. Without this technology, it would not be possible to think of a BI system and all its benefits.

In the following sections we will analyze in more detail, first from a theoretical and then from a practical point of view, the design and management of a Data Warehouse. We will also create a Data Warehouse populated with a decade’s sales data from a pharmaceutical products distribution company, with a typical response time of any query on the traditional database of several hours. By creating a specific Data Warehouse, we want to overcome the inefficiency due to response times and give management an effective tool to perform queries useful for decision support.

II. BACKGROUND

A. Files and databases

The limits of traditional data files derive from:

Redundancy. The same data appear two or more times.

Incongruence. When information is updated in a store and not in another, or there are different values for the same data.

Inconsistency. Available data is no longer reliable, because no one knows with certainty which of several values is correct.

Databases organize data in an integrated manner through modeling techniques. Data are managed on a storage system through dedicated software, with the aim of achieving high

efficiency in processing and retrieval of data. While files are anchored to the physical media, databases are independent of the location and the physical structure of the data.

A database is managed by the Data Base Management System (DBMS), a software providing:

Consistency. The data contained in the records must be meaningful and be usable in user applications;

Safety. Prevent data from being damaged by accidental and unauthorized operations on database;

Integrity. Ensure that database transactions performed by authorized users do not cause a loss of data consistency.

Data management performed by DBMSes has these key features:

- independence from the physical structure of data;
- independence from the logical structure of data;
- multiple concurrent users;
- elimination of redundancy;
- elimination of inconsistency;
- ease of access;
- data integrity;
- data security;
- use of languages for database management and query.

B. Database models

The easiest way of structuring data consists in a flat file, suitable only for very simple databases. This is for example the case of a spreadsheet in office automation software. From 1960 onwards four main types of models for databases were developed:

Hierarchical. Suited for representing associations of type one-to-many. It has some limitations, especially in the rigidness of the data structure created, which sometimes can not avoid data redundancy.

Network. The nodes represent entities and associations represent the links of a directed graph, an extension of the hierarchical tree model. The main difference with this model is that a child record can have any number of parents, so avoiding data redundancy.

Relational. It represents the database as a set of tables. It is the easiest and most effective data model [2], based on few mathematical concepts with great importance to the rigorous use of a mathematical language (*relational algebra*) for manipulating and querying data [4].

Object-Oriented. Based on the concepts of *object* and *class* [3]. An object has two characteristics: properties (or attributes), and methods (or actions). A class is an abstract type of similar objects.

III. DATA WAREHOUSES

Data managed and stored within typical transactional systems do not allow the type of analysis users demand, hence the need for companies to implement a corporate information system based on a Data Warehouse (DW).

A DW is a computer file containing the data of an organization, designed to help on easily producing reports and analysis. This approach provides an excellent way to move the border from operational processing to decision-making. It serves the continuously rising needs to analyze business information and to do this easily, quickly and correctly. The DW consists of a copy of data from transactional systems, stored in such a way as to facilitate access to those (users and/or applications) who have to make decisions based on them.

Data warehousing techniques resulted from the need to overcome the limitations of transactional systems. A DW collects data from various transactional systems, integrates them into logical sets relevant for end users, stores data in an accessible and easy to understand way, providing direct access to data by users through powerful graphical tools for querying and reporting.

Data warehousing techniques allow access to data without interfering with every-day system operation, which often prove critical for business development. The result is the possibility for the end user to take better quality decisions faster, easier and with fewer errors.

A. Definition

If we want to give a precise definition of Data Warehouse we can see it as “a collection of integrated, subject oriented, time variant and nonvolatile support for decision making” [6].

Integrated. Fundamental requirement for a DW is the integration of the data collected. It merges data from multiple transactional systems and external sources. The objective of integration can be achieved along different paths: through the use of uniform coding methods, through the pursuit of a semantic homogeneity of all variables, using the same units.

Subject Oriented. The DW is focused on business issues rather than specific applications or functions. In a DW data is stored so as to be easily read or processed by users. The goal, therefore, is no longer to minimize redundancy through normalization, but to provide data organized in a manner that enhances the production of information. We move from functional design to a data modeling which allows a multidimensional view of themselves.

Time variant. Data stored within a DW cover a much wider horizon than those stored in an operating system. The DW contains a wealth of information on areas of interest that capture the situation of a particular phenomenon in a given rather extended time period. This implies that data contained in it is updated until a certain date which, in most cases, is earlier than when the user queries the system. This differs from what occurs in a transactional system, in which data are always up-to-date, usually not allowing to provide a historical overview of the phenomenon.

Nonvolatile. This feature indicates that the data contained in the DW is not modifiable, so allowing only read-only access. This involves a simple database

design than that of transactional applications. In this context you do not consider the possible problems due to updates, nor do resort to complex tools to manage referential integrity or to lock records being accessed by other users.

B. Data management

The data flows in a DW are:

Inflow: ingoing data flow;

Upflow: ingoing flow of aggregated or summarized data;

Outflow: outgoing flow of data to users or applications;

Downflow: internal flow of further aggregation of data;

Down & Out: flow with which data are removed.

The process of data flow for a DW begins with the transfer of data to the same. Data are collected inside the transactional systems and are sent to the DW. This process is called *Inflow*.

Data in a transactional system are normally at a detail level. Part of it is aggregated and summarized to give users a faster response time. Data thus processed is transferred into the DW, and this process is called *Upflow*.

Data can be made available from a DW to end users which, through query and analysis tools, can receive an output data stream from the DW. This process is called *Outflow*.

Data stored in a DW can be re-stored in it with a format of further gathering before being removed. This happens when data are old and are no longer used with sufficient frequency to justify their presence. This process is called *Downflow*.

The original data of a Downflow process are removed from the DW and transferred to external storage media. Data are not simply deleted and then lost forever. If that were true, we would lose the historical memory of the company in contravention of the basic principles of data warehousing. This process is called *Down & Out*.

C. Applications

Applications are categorized according to the business requirements the system must consider. Data warehousing applications can be grouped in the following categories.

1) *Individual productivity:* The applications used to process and present data on a user's PC are usually developed in an independent environment and access and manipulate limited volumes of data.

2) *Data query and reporting:* This type of application is used for limited complexity queries and reports on historical or recent enough data.

3) *Planning and analysis:* This type is used for complex analysis of historical data and for planning and forecasting the future based on those same historical data. The result is the planning and forecasting of future events, simulations, assessments of processes and opportunities. These applications are known as OLAP (On Line Analytical Processing) [5].

These applications are front-end tools to access, extract and analyze data in a DW.

D. Databases

A database is a collection of (in some way related) data. In a DW they must contain a large volume of data, both historical and current. We must therefore rely on a powerful technology that can handle large volumes, but at the same time offering flexibility of access and retrieval of data. There are two types of databases that can be used to contain and manage data in DW:

- Relational
- Multi-dimensional

1) *Relational databases:* A database is called this way when the data elements and relationships that exist between them are recorded in tabular form. It is used both for transactional systems and DWs, but data are optimized differently because of different requirements that characterize the two types of systems. Transactional systems are engineered to manage daily operations, then the database is optimized to enable efficient updating of individual records. The data are normalized, i.e. table data are "splitted" into smaller tables to avoid redundancy that could be hazardous to data integrity and the speed of update operations.

A DW is built to allow instant access to data and ensure the flexibility of queries and analysis. If data were normalized access and query would present difficulties because they must be collected from several tables, a slowdown in response time. Then in a DW data should be de-normalized, i.e. stored in a smaller number of large tables, to improve performance of query tools.

2) *Multidimensional databases:* Are especially designed to meet the needs of managers, executives and analysts who want to see data in a particular way, do a large number of specialized queries, and analyze results using special techniques. A multi-dimensional database represents data in it as dimensions and not tables. It is easier to use and faster than a relational database. In addition to providing an overview of data in more dimensions, this type of database supports data storing at various levels of aggregation, because dimensions are hierarchically structured, also supporting the ability to data drill down and roll up.

Multi-dimensional databases are particularly appropriate when there are many interrelationships between database dimensions. Otherwise, a relational database is more efficient.

E. Objectives

The objective of the approach to data warehousing is in effect the user: the DW is designed to allow flexible access to data in a business environment. Data in a DW (as already stated) are:

- targeted towards a defined subject (*subject oriented*);
- *integrated*;
- permanent or static (*non-volatile*);
- with a temporal depthness (*time-variant*).

Data in a transactional system are targeted to an application, aiming to provide support to an application process. Data in a DW are targeted to a particular subject, that is being intended

to provide support to a decision. Data in a DW are the result of a consolidation of data from various transactional systems. While a transactional system continually replaces old data with new, a DW absorbs new data and integrates them with those already present. In fact, data in a DW is a mix of current and historical data.

IV. DATA WAREHOUSE COMPONENTS

A DW does not have a predefined structure, but it is determined by the components used. You can decide to bring in a DW only those components you need according to the needs, and develop only those that are not available. To implement a DW you must first understand the needs of users. It is necessary to understand whether and how the various technology products on the market meet or not the needs of designers, developers, administrators, and all users of the company where the DW will be implemented.

A DW has the following logical components:

- *Data definition*
- *Data collection*
- *Data management*
- *Metadata*
- *Analysis*

A. Data definition

This component is aimed at the design and definition of the environment of the data warehouse and then to design and define its structure, identify data sources, define the data cleansing and transformation rules that affect the data in a format usable by decision-making processes. It is used to design and establish the database of the DW, that is to create:

- entities (tables);
- attributes (columns);
- identifiers (keys).

This component is also used to identify the various sources of data both inside and outside the company, namely:

- information contained in transactional systems;
- data from the industry.

B. Data collection

The objective of this component is to acquire the necessary data, and then adjusting the data flow. Designers and developers of the DW use this component to extract data from various sources, clean and process them and do their mapping on structures created and load them into the database. The extraction of data from sources is made through analysing them using the appropriate selection criteria applied to generic or specific programs.

After extraction, this component takes care of data cleaning, namely:

- remove inconsistencies;
- add missing data;
- ensure that data integrity is maintained.

In addition, a data transformation is also carried out to:

- add time fields (e.g. date of extraction);

- aggregation of detailed data;
- derivation of new fields.

The mapping of data structures and their loading into the database are also aspects of this component. Mapping can be done both through generic tools and specialized programs. Loading may be accomplished either by utility programs specific to the selected database or through ad hoc programs.

C. Data management

This is a component that provides services to other components and manages all databases within the DW. The services offered by this component are:

- derivation of new data summarized from detailed data;
- data distribution to users' workstations;
- application of security policies;
- recovery operations in the event of data loss;
- data archiving;
- continuous data check.

Data management is responsible to create, access, extract, maintain data across the DW. So that these services are effective, the DBMS must be capable of processing large volumes of data efficiently and, in particular, must support parallel accesses and sophisticated indexing criteria.

D. Metadata

They represent all the information regarding the mass of data in a DW, and can be compared to a library catalog, which helps the reader to know if there is a book shelf and what it is. Metadata provides information on the description of data, their structure and where they are registered.

E. Analysis

It allows obtaining the benefits of implementing a DW, supporting the achievement of data by users and their analysis. This support consists of providing direct access to DW's data, inspecting them through multi-dimensional views and allowing ad hoc or predefined queries. This component contains OLAP tools aimed to analyzing the data contained in the DW. These tools are designed to speed data obtaining, summarizing and analysis, and present a multidimensional view, using an engine of the same type.

V. OLAP

Acronym for the term On Line Analytical Processing. It designates a set of software techniques for interactive and fast analysis of large amounts of data that can be examined in rather complex ways. This is the technological basis of the DW and, for example, is used by businesses to analyze sales results, trends in costs of buying goods, by marketing to measure the success of an advertising campaign, by a university to analyze survey data and other similar cases. OLAP tools are different from OLTP because the former goal query performance and width, while the latter are aimed at ensuring transaction integrity and security.

A. Functionalities

Creating an OLAP system consists in taking a photograph of information at a given moment and turning it into multidimensional data. By subsequent queries on these structured data you can get answers more quickly than similar operations on other types of databases, because for example an OLTP system is not designed to allow articulated analyses.

An OLAP structure created for this purpose is called *multidimensional cube*. There are several ways to create a cube, but the best known is that making use of a “star” schema where the center is the *fact* table which lists the main elements on which the query will be built, and connected to this table are several *dimension* tables that specify how data are aggregated.

For example, a customers’ data set can be grouped by city, state, region, and these customers can be related to the products and each product can be grouped by category. Computing the possible combinations of these groupings form an OLAP structure that potentially could contain all the answers for each combination. Only a predetermined number of combinations is actually stored, while the remaining are recalculated only when the request is really effected.

An OLAP system allows to:

- study a large amount of data;
- see data from different perspectives;
- support decision making.

B. Types of OLAP systems

There are three types of OLAP systems: MOLAP (Multi-dimensional OLAP), ROLAP (Relational OLAP) and HOLAP (Hybrid OLAP).

MOLAP is the most widely used and it is simply called OLAP. It uses a summary database with an engine specifically designed for multidimensional analysis and creates dimensions with a mixture of detail and aggregation. It appears the best choice for small amounts of data, because it is fast in calculating aggregations and returning results, but it creates huge amounts of intermediate data.

ROLAP works directly with relational databases. Data and dimension tables are stored as relational tables and new tables are created to store information aggregation. It is considered more scalable and requires less disk space and less RAM, but is slow in setting up tables and producing query results.

HOLAP uses relational tables to store data and multidimensional tables for “speculative” aggregations. It stands in the middle, can be created faster than ROLAP and is more scalable than MOLAP.

The difficulty in implementing an OLAP database starts from the hypotheses of the possible user queries. Choosing the type of OLAP, its schema and creating a complete and consistent database is very complicated for a large and diverse user base. To meet users’ needs, many modern products provide a wealth of pre-set diagrams and queries.

C. OLAP characteristics

The basic functions of an OLAP tool are:

Slicing. The operation of rotating analysis dimensions.

It is a fundamental operation for analyzing totals obtained on different dimensions or if you want to analyze cross aggregations.

Dicing is the process of extracting a subset of information from the aggregate that is being analyzed. The dicing operation is performed when the analysis is focused on a piece of the cube of particular interest to the analyst. In some cases, the dicing operation may be “physical” in that it does not only consist in filtering the information of interest but also in extracting them from the general aggregate to deliver its content.

Drill-down is the process of “explosion” of data into its determinants. The drill-down operation can be performed along two different paths: the *hierarchy* built on the dimension of analysis (e.g. the transition from the product family to all products in it) or the *mathematical relationship* that links a calculated data to its determinants (e.g. the transition from the margin to the income and expense that generate it). You understand the importance of this operation for analytical purposes in terms of understanding data determinants.

Drill-across. The process by which you navigate through the same level in a hierarchy. As noted above, the transition from product family to product list is a drill-down operation, while the transition from one family to another is an operation of drill-across.

Drill-through. Conceptually similar to drill-down, is the process with which we pass from an aggregate level to the detail level belonging to the normalized database. Many vendors proclaim that their products have the ability, through the operation of drill-through, to go from the DW to the transactional systems that feed it. This operation, although technically feasible in a number of quite substantial conditions, is not very feasible because of the security and performance problems induced in transactional systems themselves.

D. Weaknesses

The weaknesses of OLAP tools are:

- inaccessibility/difficulty in accessing the data at the atomic level: OLAP tools work very well on synthetic data, it is not convenient to use them on analytical data;
- backup/restore/security/rollback systems not very sophisticated or non-existent: while in many cases they are database engines, OLAP tools have not yet reached the level of completeness of relational databases, mainly because, unlike the latter, they do not have a conceptual paradigm of reference, but are subject to different interpretations of software vendors;
- require a de-normalized structure to function efficiently. OLAP engines generate large amounts of data for the simple fact that they are forced to store redundant keys and summaries to improve access performance;

- possible proliferation of SQL code. If the database on which OLAP analyses are carried out is not multidimensional (MOLAP) but is relational (ROLAP), each of the above operations (slicing, dicing, drilling) causes the generation and execution of extremely complex SQL queries that require a lot of computing resources.

VI. DATA WAREHOUSE MODELS AND DATA STRUCTURES

A DW is typically based on a data model different from that used in transactional systems. Data include:

- the typical structures of databases (tables, attributes and key fields);
- the representation of the relationships between these different structures;

The data model mainly used for database designing is the Entity/Relationship model (ERM). This type of model, however, presents problems: as in reality, entities have different characteristics, contain a different quantity of data, etc. Therefore it is necessary to adopt a multi-dimensional view.

To allow a multi-dimensional visualization of data, techniques have been developed known as "schemas" and are:

- star schema;
- snowflake schema;
- mixed schema.

A. Star schema

A model that uses the star technique reflects the way a user sees the data. For example, data contained in an invoice through the dimensions customer, product, supplier, geography and time.

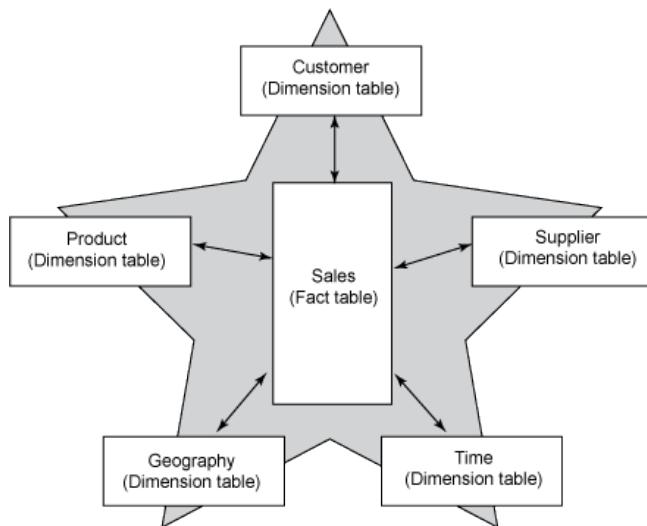


Fig. 1. A diagram of the star schema (source: ibm.com)

The fact table (invoice) contains only attributes that measure the business, in addition to identifiers (foreign keys). The dimension tables (customer, product, supplier, period, geography, time) contain attributes describing the dimension, in addition to identifiers (keys) that index and organize the data in the fact table.

B. Snowflake schema

It is an extension of the star schema in which one or more star vertices spread. All dimension tables are normalized. For example, the table of the *time* dimension is normalized to quarter and month.

Advantages compared to the star technique are a query performance improvement through better storage occupation due to elimination of redundant data and use of small normalized tables instead of large not normalized tables.

The disadvantages are due to the fact that it presents a more complex structure. There is a greater number of tables and therefore greater difficulty in deciding which table to use in a query.

C. Mixed schema

In some situations there may be some dimension tables with substantial differences in the number of attributes and volume. In this case it is not possible to use the star or snowflake model for the whole structure: you must therefore use a combination of both, called mixed model.

The decision on which data model to use depends on data characteristics and requirements of the organization that will use the DW.

VII. DATA MANAGEMENT AND PERFORMANCE IMPROVEMENT

A DW contains a large amount of data. Its success also depends on the efficiency with which they are managed. There are techniques for performance optimization that are used to allow the DW to meet the needs of users. These techniques help to reduce the disk space needed to hold data and improve response times.

Granularity is defined as the level of detail present in a unit of data. This level of detail corresponds to the level at which data is recorded in the DW. The granularity has impacts on the volume of data maintained and the type of queries that could be achieved. There are three levels of granularity:

High. Data are aggregated and summarized, and therefore preserved in a less detailed format. It reduces the space needed and improves response time, but restricts the types of queries because data are less detailed.

Low. Data are kept in great detail.

Double. Certain data is stored in detail and others are summarized and aggregated.

A DW can be optimized with specific techniques:

Partitioning. Dividing a unit of data into smaller units according to criteria defined a priori. The breakdown of data into smaller groups also facilitates management and access.

Summaries. Creating a high level of granularity. It reduces the level of detail and then assists users in decision making especially with regard to forecasting. There are no fixed rules for summaries: they depend on the specific needs of required analyses. The advantages are in terms of need for less disk space, faster response times and lower operating costs. The only drawback is the loss of detail.

VIII. DESIGN AND MANAGEMENT OF A DATA WAREHOUSE

A. Identifying needs

When building a DW it is necessary to identify and address the needs of end user, designers and developers. Designers and developers will be able to build a DW based on the requirements expressed by end users. An end user uses the DW to support decision-making. The requirements can be classified into three categories:

Query. The needs of an end user can be determined by collecting examples of possible queries from various sectors of the company.

Analysis. All types of analysis that an end user might do are to be identified.

Reporting. The requirements for reporting can be determined by collecting examples of all reports produced by various sectors of the company.

B. The designer

Is responsible for defining DW's topology according to the needs expressed by users. Based on these requirements, the designer decides on the DW's scope, the amount of data that must be preserved, both historical and current, and the technology to be used, both hardware and software.

C. The developer

Is responsible for converting the conceptual (logical) data model into a logical (physical) data model. He must also provide for:

- where databases and tables are physically to be defined;
- applications that, within each component, execute processes necessary to meet users' requests;
- the programming language most appropriate for applications;
- the communication protocols;
- a method and tools for efficient data access;
- methods for optimization and data security.

D. Data Warehouse design

Designing a DW means selecting the sources of data a company needs and define the data structures that constitute its DW. After identifying the needs, the designer must select the sources from which to extract the data. One of the most important tasks is to ensure that sources contain accurate, complete, and up-to-date data.

Sources of DW data are classified into three categories:

Internal. Internal data is data generated inside the company and stored within the transactional systems used by the company for its management.

Archived. Archived data are data not used by the company, moved on remote magnetic supports. These data constitute an important historical basis on which to base the trend analyses during decision support.

External. These data are generated outside the company and are commercial data. They provide information on competitors and market trend.

After choosing data sources for the DW, designers define the data structures and then the entities that are mapped to database tables, and represent the relationships between the entities using the star, snowflake or mixed model. Designers convert the logical definition of the structures in physical definition through the creation of databases. During this process entities are mapped to tables, and their attributes and key fields are also defined together with additional attributes indicating temporality (e.g. date and time when the record was written) and others containing aggregated or summarized data. This conversion process is sometimes referred to as the *mapping from the conceptual (logical) model to the logical (physical) model*.

Strategies for data management are:

- *de-normalization* (the technique of keeping data on a smaller number of larger tables);
- *indexing* (data classification to provide quick access);
- *partitioning* (splitting a single unit of data in a number of smaller units);
- *aggregation* (the summarization of data).

The path to implement a Data Warehouse includes the following steps.

- Creating the interfaces between the DW and its data sources (this allows the DW to be populated).
- Refining the data through the use of cleaning and processing tools (includes cleaning and processing data):
 - *cleaning* (is carried by utility programs that can be purchased on the market or developed internally);
 - *transformation* (adding a "time" element to data; data aggregation).
- Loading data into the DW (populating the Data Warehouse).
- Making the DW available to users (this means to complete installation and testing). It is delivered to users in sections and not in one unit. This approach is safer because one can check the feedbacks of a section and then load the subsequent sections based on them.

After the DW has been made available to users it is possible that the need arises to make changes. From time to time with predetermined frequency data in it must be updated (refreshed). A crucial phase is also that of maintenance, where you manage the size of the DW, ensure the security of its data and monitor its performance (measured by response times to queries). Access to data by its nature involves risk. A full and indiscriminate access to all corporate data is not conceivable precisely because of the risks involved. It is then necessary to introduce the concept of *user access rights*.

IX. DATA WAREHOUSE IMPLEMENTATION

A. Enterprise Data Warehouse and Data Marts

A crucial task for a company is implementing a Data Warehouse (DW) or a Data Mart (DM). These are essentially a kind of specialized database that is designed to support

business analysis and include data from one or several sources. The difference between a DM and a DW is their scope. A Data Mart seeks only to satisfy the needs of a company's sector, such as marketing or finance department. Can be thought of as a small warehouse. A DW seeks to serve the "whole" company. So it is defined of business interest when responding to the needs of the entire company, i.e. includes business data that are used by all sectors inside the company to make decisions on the future of the same.

The creation of multiple sectoral Data Marts (*bottom-up approach*) as an alternative to a single enterprise DW involves fragmentation in the overview of a company's data. Data Marts will be built for every user department within the company, and their combination come to the Data Warehouse. It will therefore be difficult to reconcile these views back to get the whole enterprise overview. In essence, the various DMs will have gaps and will never offer an enterprise-wide view. Choosing a DM instead of a DW may be dictated by the need to implement a decision support system in a short time and with reduced resources.

A centralized Data Warehouse (*top-down approach*) is an approach aiming to build the data repository in the first place, considering the needs of the company already during the design phase. It is used to keep the data organization in a single container, as the majority of decision-making is done in one place. It is obviously a very complex task: you must consider and study how the entire organization works. Once built the DW, you can create DMs by extracting a portion of the DW to meet the needs of specific sectors, such as finance or marketing.

It is recommended when:

- the data are integrated within the organization and headquarters takes a global view of them;
- a single database provides for storing whole company's data;
- data are not easily accessible if they are kept in different locations.

There is also an alternative called *Client/Server*. Several computers on which the DW's portions are stored are considered either as clients or as servers. Clients are computers that require a service from another computer. Servers are computers that provide a service to another computer. Client and server can sometimes reside in the same computer.

There are two alternatives:

- *Two-Tier* (pure client/server). There are two levels of handling. The client services are held on a platform (client), the services on another (server). Both a workstation and the mainframe can be used as server. Data access tools are kept on the client.
- *Three-Tier* (mainframe/server/workstation). Three levels of technical platforms. The mainframe is used for the extraction and processing of data from data sources. The server contains data from the DW and its related software components. The workstation contains user applications (front-end) and data taken from the DW (DM).

B. Data recording

A DW requires different units, logical and physical, which hold and handle the data. Due to the volume of data that are part of a DW, management must be efficient and effective regarding time and costs. The technology choice should be oriented towards hardware and software that allow the use of a variety of tools. Only one type of units available (e.g. Direct Access Storage Device - DASD) is not mature enough for a DW.

The main units to consider are:

- Main Memory (high cost, extremely fast access)
- Extended Memory (relatively high cost, extremely fast access)
- DASD (affordable, very fast access)
- Other magnetic media (low cost, access is quite slow)

C. Implementing the Company Data Warehouse

The DW is an information system where data are organized and structured for easy access by the user and for supporting decision making. The following systems are enabled:

- | | |
|-----|---|
| DSS | (Decision-Support System) used to solve specific problems. |
| EIS | (Executive / Enterprise Information System). Allows a continuous flow of data not dependent on specific problems. |

In banks and financial institutions generally the areas of use are varied, since all management areas of these organizations are characterized by large volumes of data on which strategic decisions must be taken. Because the DW can have a strategic value, inside such organizations it is fundamental to define a management strategy. The strategy for the DW is essentially an evolutionary path that carries the enterprise from DW applications not *mission critical* to a situation in which the DW is a key component of corporate information system.

A company's strategy for data warehousing can be classified according to two basic dimensions:

- 1) using existing data warehouse: level of maturity of users and DW's support functions in the use of what exists;
- 2) using the data warehouse in perspective: the use of the DW as a platform for decision support.

Companies then pass through four stages in the history of use of the DW:

- the first stage, called *support* (low use of existing DW, low prospective use of DW) is the stage where are companies that have failed one or more data warehousing projects and do not think to extend the prospective use. At this stage you can also find companies that do not have a DW and do not think to realize it;
- the second stage, called *opportunity* (low use of existing DW, high use of prospective DW) is the stage where are companies that, despite having failed one or more warehousing projects, or having simply explored the issue without any deepening, aim to develop the decision support activities through the DW;
- the third stage (high use of existing DW, high-usage of prospective DW) is that phase in which the DW becomes

- a *strategic* for business decision making. At this stage are those companies that have successfully undertaken a warehousing project and are exploiting its full potential;
- the fourth stage, called *factory* (high use of existing DW, low use of prospective DW) is the stage where are farms with a mature DW, an established implementation methodology and managed critical decision areas. At this stage the main requirement is efficiency and cost savings arising from the DW and its use. A process of sclerotization in using the DW may in some cases cause the company to return to the first phase.

Let us now identify the application areas most suitable for the DW in the financial sector.

1) Management Control: This may be the basic application area for a data warehousing system in any organization. In this case, the DW is used mainly as a platform for reporting and analysis of profitability. It is useless and dangerous to suggest to implement a DW just for management control. This initiative makes sense only if this is the first evolutionary step in the company's data warehousing strategy. In fact, building a DW for management control lets you quickly analyze and resolve highly relevant needs and whose benefits are immediately clear, addressing problems (in the structure, validation and data processing) well known in their structure.

2) Risk and Asset Management: Another interesting application area is identified in Risk and Asset Management activities in two main very specific topics: analysis and simulation of portfolios and related risks, and reporting. These application areas are of particular importance and strategic nature and the DW is the appropriate tool to address them, including the ability to integrate its internal data from company's external sources. In this case, the DW is equipped with advanced analysis tools and algorithms based on statistical analysis and simulation. Another sub-area of great interest may be the development of systems for fraud detecting. Again it is necessary the use of statistical instruments.

3) Sales Support: DW is not necessarily appropriate to address and resolve this type of requirement, unless there is a need to store and manage significant amounts of data. In many cases the marketing database is trivially a customers' file enriched of certain "non-administrative" information; in more advanced cases it becomes a key tool to support *one-to-one marketing*. In this case, the marketing database is an information base essential to correctly drive campaigns and promotions or to enable advanced *customer care* services. In this case, given the substantial amount of data to manage the DW can become the ideal technology platform.

In the banking sector one-to-one marketing is still at the embryonic stage, at least from the perspective of Central Marketing, and this is because very often one-to-one marketing is done by the branch, the only enterprise structure historically able to establish a trust relationship with the end customer, which identifies the company with the front office and its clerk.

4) Marketing Information System: It is about using the DW as a "backbone" to support a set of integrated applications oriented to business and market analysis. The fundamental aspects that characterize this type of architecture are essentially two:

- the ability to integrate different transactional databases into a single database and then produce analytical and integrated views of customers and market products;
- the possibility of analysis with different tools and a different logic on a single basis.

The idea of marketing information system is to develop an evolutionary path that starts from basic reporting to advanced analytics to achieve, through systems of product and customer portfolio analysis and budgeting and simulation procedures.

5) Call Center Support: Again the DW is a technological option, not the only viable and not necessarily the cheapest. Use of data warehousing architecture to support call center activity has certainly sense if the requests are not necessarily of structured type and then solved with the classic terminal inquiry. It is also clear that the type of user for this kind of system is more advanced than the normal call center operator.

6) Knowledge Base: Even in this case are valid the considerations already made for the marketing database: DW is not necessarily the most suitable technology for this kind of stuff, but it gets when the knowledge in question is mainly composed of structured information and preferably numerical. In this case, also from the technological point of view, a relational database is certainly the best solution, efficient and economical. Not so if the information is of unstructured type; in this case the best solution is a groupware platform.

7) Product Engineering: The DW can be a decision platform for analysis and conceptualization of new products to offer to customers and/or to attack new markets or market segments. This functionality is obviously supported if the DW is not only equipped with tools for result analysis, but also with simulation environments that allow the construction and "laboratory" testing for new solutions to offer to customers. In such environments it is possible to identify some important issues such as marginalization, the economic break-even point, the customer segment concerned, the cannibalization mechanisms, the elasticity of demand and the impact on the business' financial balance.

8) E-business: The spread of digital channel in the financial sector poses a number of problems and new opportunities. First, this type of channel implies a change and then reaction rate considerably higher. The DW may be the analytical tool that allows you to capture important dynamics within the masses of online transactions. Secondly, the information can be a support tool or the object of the transaction and in this case, the DW can be the platform used to cover that scope.

The Data Warehouse can then support online trading systems, both in terms of analysis and of data architecture point of view.

X. DATA WAREHOUSE'S DESIGN AND IMPLEMENTATION FOR A PHARMACEUTICAL DISTRIBUTION COMPANY

A. The transactional system

As mentioned in the introduction, we will now show the design and implementation of a DW populated with data relating to sales of a pharmaceutical product distribution company (located in South Italy), which already has a system based on relational databases. This system, implemented in 1998, allows the company to record yearly sales figures. The system allows to record sales' data as the dispatched and ordered quantities, the customer who placed the order, the items sold, the product category, the price of sale, the standard cost, last cost and retail price.

The reference data for the construction of the DW are relative to the years 1999–2009. The relational database is structured as follows.

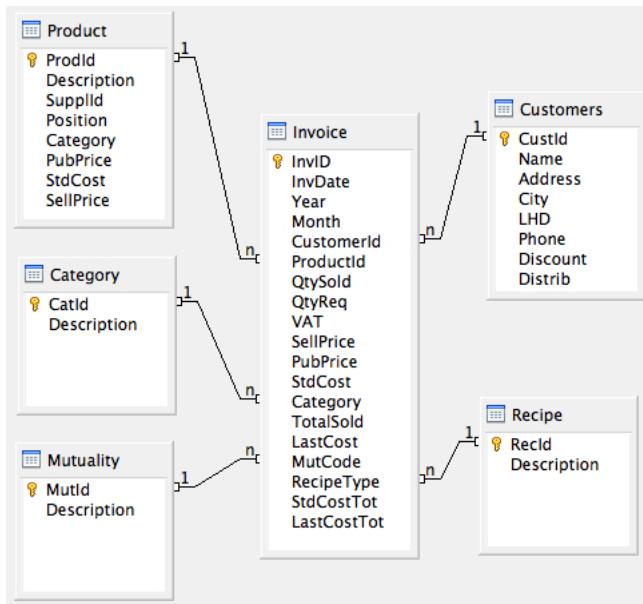


Fig. 2. Transactional system relational database

We have a table (Invoice) structured so as to accommodate all information relating to a sale, as the dispatched and ordered quantities, the sale price, the item sold, and so on. To this table are linked (with one-to-many relationships) other tables as: Category (product categories), Customer (who buy products), Product (items buyed by customers), Mutuality (health system products' distribution status), Recipe (type of recipe prescribed by the physician).

You can easily see that there is an apparent redundancy in the data due to duplication in some areas such as sales price, retail price, standard cost and last cost which are in fact present in both Customer and Product tables. The reason for this is because such measures are always subject to change, so it is advisable to historicize them. Same goes for the decision to link the Categories table to the Invoice instead of Product table.

Several queries have been implemented in the transactional system that the sales manager can use in order to analyze sales data. These queries are used to perform small aggregations of data, in order to study them from different points of view. For

example, reports are built on the Top 10 shareholders in terms of sales, a list of products sold by geographic area or category, etc.

The problem arises when you want to perform a bit more complex tests, such as listing all sales made by all customers in a wide time frame, or all sales divided by city, category and product. Before obtaining the results it will take several minutes, if the time horizon is limited to one month and we restrict the analysis to a single customer, category or location. These times would rise exponentially if we would extend our analysis to time intervals of whole years for all customers, cities or categories.

B. How to overcome the limitations of the transactional system

From what said it is evident that the existing system based on a relational database can not make more complex analisys, otherwise you run into time delays in the order of hours. It is also not possible to perform analysis of data in their various facets, in the sense that when you try to weave together multiple variables the relational engine breaks down.

These limits are overcome thanks to a tool (the DW) that allows for upstream data aggregation, so it is more effective to perform more complex analyses, both in terms of feasibility and in terms of response times. The analysis of data that can be performed with a DW are called OLAP. While with a query system the user may, for example, ask what is the best-selling product in a particular month of the year, in a particular geographical area, with an OLAP analysis you can ask the system to perform a report of all the products most sold in all months and in all geographic areas. It allows you to query across multiple dimensions, so it is seen as a hypercube because there is an interweaving of multiple dimensions such as geography, time, etc.

Also, depending on who is the decision maker, the cube is queried in a different perspective. E.g. the regional manager, responsible for sales in a particular region, will be interested only in selling of products in a given market, while the finance manager will consider the sale of products in all markets for the current and previous period. The product manager reviews the sale of a product at all times and in all markets, and finally the top manager will focus on product category, regional area and a medium time horizon. A DW-based system allows, therefore, for multiple-reporting, more complex and more useful for commercial purposes analyses.

C. Data Warehouse design and population

Having found the limits of a system based on relational databases and seen how a DW deals with them successfully, we turn to its design based on tables in the existing system. First we must consider that the transactional system has a different (relational) databases for each year. In the DW we will merge all data into a single structure. To do this, there are two possible ways:

- 1) Bring together all the information relating to various databases of individual years in a single database containing them all (see Fig. 3).

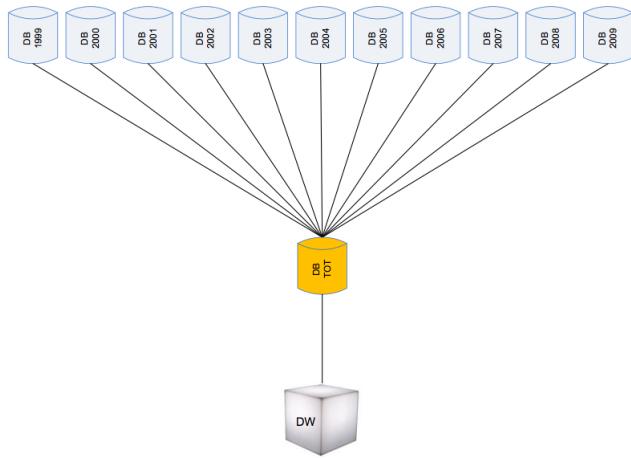


Fig. 3. Populating the Data Warehouse with a single cube

- 2) Create a data cube resulting from the sum of many cube partitions, each consisting of a year of sales data (see Fig. 4).

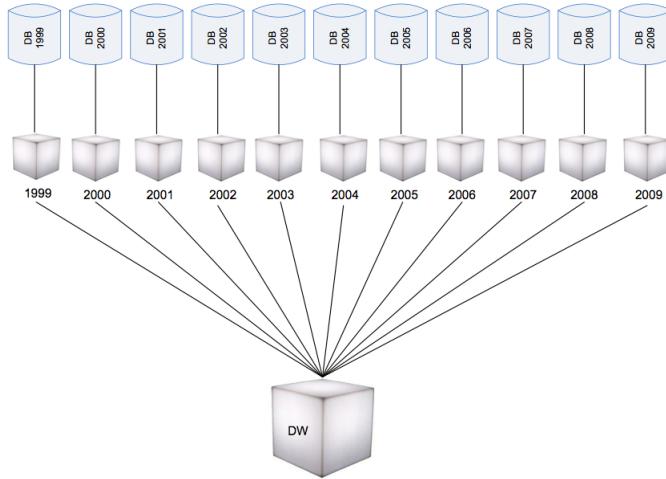


Fig. 4. Populating the Data Warehouse with cube partitions

The preferred choice is the first, to merge all data relating to several years of sales in a single database and then process it as a data cube.

Before we proceed with the migration of data into the overall database, we must clean data by deleting irrelevant fields and adding fields essential to the development of the cube, primarily the fields *year* and *month* that will be used for the time dimension.

Subsequently computed columns have been created to compute the following totals:

- Sales = *quantity dispatched* × *sell price*
- Standard cost = *quantity dispatched* × *standard cost*
- Last cost = *quantity dispatched* × *last cost*
- Retail price = *quantity dispatched* × *retail price*

Then the overall database can be populated with data from all yearly databases. and the design of the real cube can start using a star schema (see Fig. 5).

The Invoice table is established as the fact table. The measures of our cube are: Qty ordered, Qty dispatched, Total sales, Total standard cost, Total last cost, Total retail price (all measurable fields). Dimension tables are so arranged:

- Time dimension (divided by year and month)
- Customer dimension
- Geography (City) dimension
- Category dimension
- Distribution (customer grouping) dimension
- Supplier dimension
- Local Health District (LHD) dimension

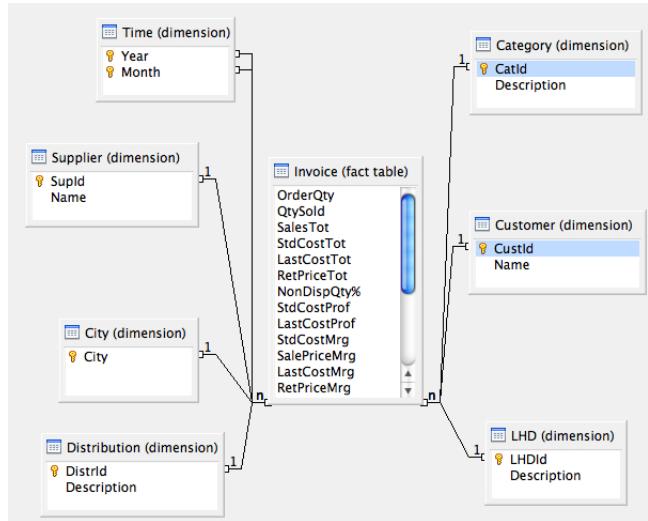


Fig. 5. The data warehouse cube

Finally, once the cube structure has been created, some computed members have been generated such as the percentage of not dispatched quantity, the return on standard and last cost, and the various margins.

D. Creating the data cube with Sql Server

After designing the DW we proceed with its creation in SQL Server 2000 – Analysis Services. We will look very briefly at this stage because it is a purely technical step which can change according to the tool that is used to create the DW. Of vital importance is the design of the DW, because if well done it can be implemented with any tool and without problems.

The first step involves the construction of the data source, i.e. to instruct SQL Server to catch data from the operational database. To do this, open the Analysis Manager tool included in the installation package of SQL Server 2000 and right-click the mouse on the server name on which we are working and then choose “New Database”, entering the name of the database to work with. Then click again with the right mouse button on the data source and then on “New Data Source”. This way we can select the data source from which our DW will populate once processed.

The next step is to create the cube data. With the same procedure shown for the data source, click on “Cubes” and

then click “New Cube”: this will activate a wizard to create a cube, where you can set the measures and dimensions as it was set at design time.

Choose the “Invoice” table as the fact table and continue with the wizard.

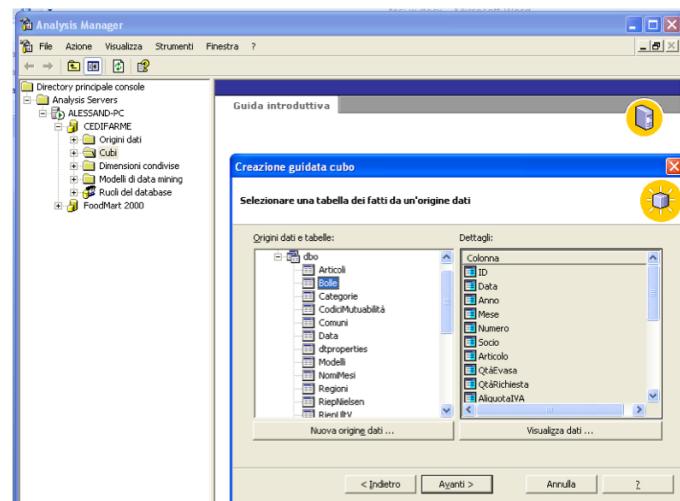


Fig. 6. *Cube wizard: choosing the fact table*

Then the cube measures will be selected. In our case:

- the ordered quantity;
- the dispatched quantity;
- total sales;
- total standard cost;
- total last cost;
- total retail price.

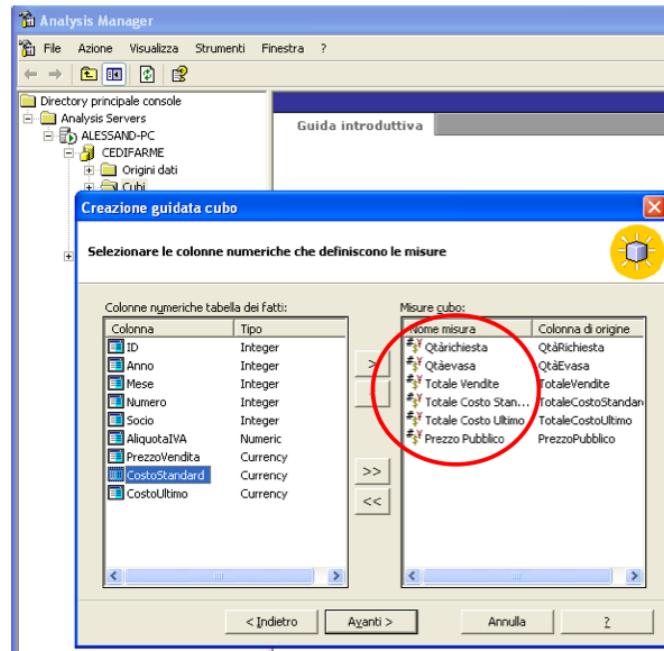


Fig. 7. *Cube wizard: choosing measures*

Then we go to creating dimensions, which are those decided at the design stage.

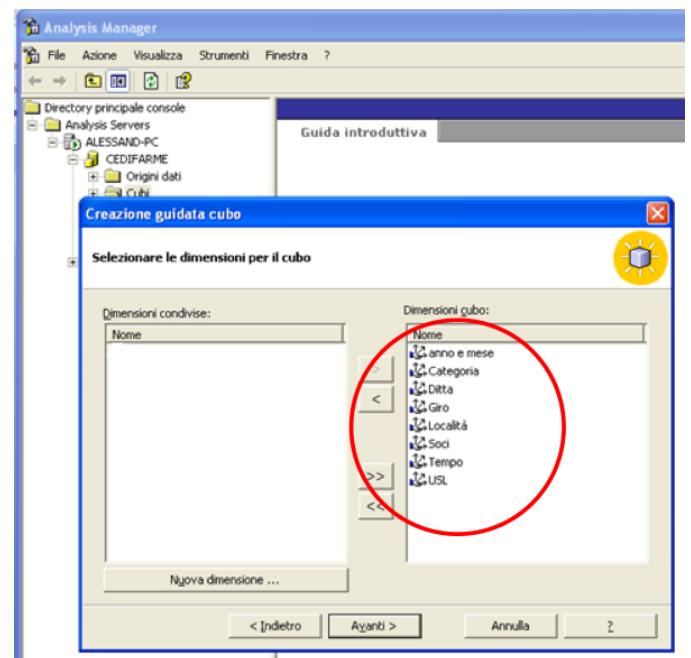


Fig. 8. *Cube wizard: choosing dimensions*

Once these operations are completed, our cube is created and can be viewed in its structure through the “Cube Editor”.

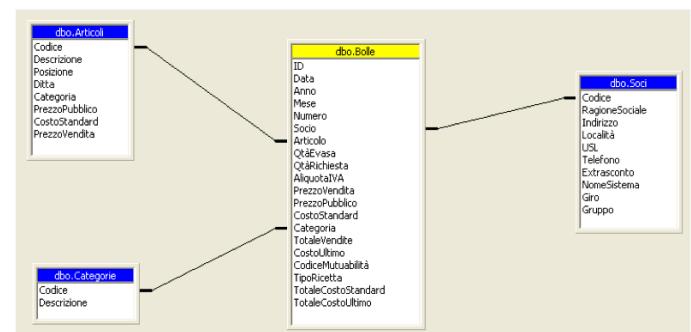


Fig. 9. *The Cube Editor*

With the Cube Editor you can edit or add cube dimensions and measures.

E. Displaying cube data

The first important tool for exploring DW data is the cube explorer already present in SQL Server 2000 Analysis Services. From here you can drill up (when you want to consider a higher level of data aggregation) and drill down (to explode the data into larger levels of detail).

Suppose we want to examine the total sales in 2009 for each product category. We put in the cube “faces” the time and category dimensions.

	+ Anno	
Categoria	- Totale Tempo	+ 2009
- Totale Categoria	€ 53.869.731,66	€ 53.869.731,66
2	€ 1.451.885,63	€ 1.451.885,63
3	€ 297.137,19	€ 297.137,19
4	€ 3.967.407,21	€ 3.967.407,21
5	€ 2.440.044,06	€ 2.440.044,06
6	€ 2.582.937,26	€ 2.582.937,26
7	€ 400.856,92	€ 400.856,92
A	€ 41.693.515,95	€ 41.693.515,95
B	€ 1.035.947,44	€ 1.035.947,44

Fig. 10. The Cube Explorer

The total over time (“Totale Tempo”) has the same results as the total of 2009 because, for simplicity, in the DW analysis we included only data for 2009. From this analysis we receive the total sales in 2009 for each product category.

If we want to explode our data at the month level we just make a simple drill down on the time dimension.

We can see here the simplicity and speed compared to a relational-based system. Response time and complexity of queries are far more efficient and effective.

In the previous example total sales were used as a measure. Of course you can set any other dimension that were included in the data cube, and you can also set more than one, which is very useful to make comparisons. Also the dimensions may vary among those set during cube creation, and more than one can be used in viewing data.

F. Using Excel as a reporting tool

Once the cube has been implemented in Microsoft SQL Server we have to make possible the analysis of data in it. The tool used to do this is Microsoft Excel (version 2007) which enables a *data link* useful for exploring and studying the DW data.

Once you connect with the DW you can choose the dimensions to be displayed in the PivotTable, the measures and the filters. Response times are pretty insignificant compared to queries on relational DB system. Indeed only about a few seconds are needed to make even very complex explorations, such as intersections with multiple dimensions and measures considering the total amount of data without using any filter. Basically, whatever the question the system does not take more than ten seconds before answering.

Moreover, an important feature of Microsoft Excel is the ability to perform many different operations on data obtained from the DW, using operators and functions already present in the application. You can also develop a macro when you want, for example, any degree of execution of certain procedures.

That said we now illustrate an overview of the queries to the DW that reproduce quite similarly the ones used by the sales manager on the transactional system.

1) *Report on quantity sales per customer:* As you can easily guess this report shows the quantities sold in time (in years and months), divided equally between customers and

classified by different categories of products sold. For privacy reasons we entered the Customer ID instead of its name.

To perform this analysis it is sufficient to choose the time, customer and category dimensions, using as measures the quantity sold. The system answered, as already mentioned, within a few seconds.

In this case we selected all customers, categories and months of the year 2009, but obviously for a more targeted analysis, only one or a group of customers and/or categories can be selected. The system also automatically gives the totals and, furthermore, may also return the minimum, maximum, average, or you could sort the data (in ascending or descending order), for example showing a top list of the top buyers in a period of time.

tempo-soci/categorie										Totale complessivo
	2	3	4	5	6	7	A	B		
2009	801.55 5	29.49 2	882.44 8	1.224.14 4	449.44 9	146.28 1	9.458.62 7	333.65 8		13.325.654
gennaio	84.999	2.033	77.522	110.130	38.097	14.354	874.530	40.940		1.242.605
100	420	3	271	443	61	3	3.641	187		5.029
1002	177	7	56	303	50	29	2.221	33		2.876
1004	331	5	185	260	90	26	2.241	84		3.222
1006	128	1	240	699	81	30	3.708	112		4.999
1008	114	9	496	67	134	45	5.871	237		6.973
1009	149	31	237	162	69	44	3.375	78		4.145
1010	230	9	107	116	64	25	2.012	77		2.640
[...]										
dicembre	56.913	2.001	70.862	84.023	31.888	10.126	774.690	23.698		1.054.201
100	403	2	316	447	61	7	4.570	182		5.988
1002	102	7	61	228	56	14	1.921	31		2.420
1004	258	2	210	275	62	35	2.154	66		3.062
1006	93	4	260	217	49	9	4.100	73		4.805
1007	12						12			24
1008	133	3	384	94	82	19	4.439	60		5.214
1009	49	16	175	49	60	33	2.419	40		2.841
1010	164	1	39	80	27	91	1.821	38		2.261
[...]										
Totale complessivo	801.55 5	29.49 2	882.44 8	1.224.14 44	449.44 9	146.28 1	9.458.62 27	333.65 8		13.325.654

Fig. 11. Report on qty sales per customer

2) *Report on quantity and value sales per customer:* The following table shows the results in terms not only of quantity sold, but also gives value indicators as total sales and costs (standard cost and last cost). In addition to this the “broken stock”, i.e. orders that have been requested but not fulfilled, is presented in percentage terms. Data are grouped by year, customer and category. This table would be very useful, for example to know the value created by each customer and which category emerges as the most important. Again, like the preceding and following, you can narrow the field to certain customers, groups or a given time period.

3) *Report on sales per location:* This further report shows the results in terms of quantity processed for each year and location (city). Then also the geographic data are considered are divided into categories. A very useful question could be where you make more product sales and which category you sell more in a given geographical location.

4) *Report on annual sales per category:* The following table shows a statistical analysis (qty sold, total sales,

tempo-categorie-soci/qtà- valore	Qtàrichiesta	Qtàevasa	non evasa percent.	Totale Vendite		Totale Costo Standard	Totale Costo Ultimo
2009	13.713.058	13.325.654	2,83%	€ 94.650.037,22	€ 87.426.493,80	€ 86.470.865,06	
100	61.260	59.808	2,37%	€ 385.195,41	€ 364.523,06	€ 360.143,83	
2	3.636	3.633	0,08%	€ 14.268,94	€ 12.476,98	€ 12.581,56	
3	9	9	0,00%	€ 89,54	€ 80,62	€ 81,86	
4	4.216	4.128	2,09%	€ 15.474,74	€ 13.739,97	€ 13.852,85	
5	5.108	5.139	-0,61%	€ 24.503,08	€ 21.751,41	€ 21.806,58	
6	969	969	0,00%	€ 5.252,87	€ 4.497,60	€ 4.557,04	
7	144	144	0,00%	€ 481,45	€ 289,27	€ 290,41	
A	44.533	43.555	2,20%	€ 316.533,66	€ 303.430,77	€ 298.579,48	
B	2.645	2.231	15,65%	€ 8.591,13	€ 8.256,44	€ 8.394,05	
1002	32.452	32.317	0,42%	€ 221.651,13	€ 199.282,16	€ 195.650,70	
2	1.231	1.231	0,00%	€ 5.407,11	€ 4.493,57	€ 4.640,95	
3	93	92	1,08%	€ 1.019,00	€ 850,49	€ 882,08	
4	1.003	1.000	0,30%	€ 5.365,88	€ 4.520,79	€ 4.561,47	
5	4.433	4.437	-0,09%	€ 21.197,51	€ 18.140,09	€ 18.122,92	
6	617	616	0,16%	€ 3.907,09	€ 3.192,80	€ 3.237,07	
7	417	409	1,92%	€ 973,24	€ 695,56	€ 740,14	
A	24.276	24.150	0,52%	€ 181.837,83	€ 165.631,15	€ 161.666,33	
B	382	382	0,00%	€ 1.943,47	€ 1.757,71	€ 1.799,74	
1004	38.525	37.875	1,69%	€ 239.206,07	€ 222.719,99	€ 220.053,56	
2	3.038	3.040	-0,07%	€ 13.134,63	€ 11.700,36	€ 11.945,73	
3	107	107	0,00%	€ 1.275,29	€ 1.141,57	€ 1.162,24	

Fig. 12. Report on qty/value sales per customer

tempo - località/categorie	2	3	4	5	6	7	A	B	Totale complessivo	
2009	801.555	29.492	882.448	1.224.144	449.449	146.281	9.458.627	333.658	13.325.654	
ADELFA	401	8	234	952	274	26	4.203	159		6.257
AIELLO DEL SABATO	988	55	951	1.458	643	122	10.199	342		14.758
ALTAMURA	2.327	27	2.091	5.136	1.397	349	18.972	556		30.855
ALTAVILLA IRPINA	2.593	210	1.925	2.699	990	256	23.406	1.105		33.184
ANDRETTA	42			107	6	2	84	9		250
ANDRIA	58.763	1.245	62.087	94.837	34.336	10.487	787.965	26.881		1.076.601
APICE	2.151	125	1.075	2.534	968	182	14.179	555		21.769
APRICENA	8.904	300	13.291	15.560	5.267	1.725	134.061	3.312		182.420
AQUILONIA	98		66	186	62	13	1.140	51		1.616
ARIANO IRPINO	3.988	207	6.448	4.776	2.045	844	42.356	1.430		62.094
ATELLA	1.986	188	1.283	2.698	684	136	37.373	1.334		45.682
ATRIPALDA				40			2			42
AVELLINO	4.555	178	3.341	8.674	2.549	555	62.871	2.168		84.891
BAGNOLI IRPINO	647	47	432	1.525	293	152	15.632	199		18.927
BARI	66.731	2.032	56.162	74.758	34.496	10.225	628.961	25.088		898.453
ZAPPONETA	2.876	97	4.880	7.217	2.127	1.127	51.733	1.474		71.531
ZUNGOLI	1.719	98	847	2.302	762	225	22.588	732		29.273
Totale complessivo	801.555	29.492	882.448	1.224.144	449.449	146.281	9.458.627	333.658	13.325.654	

Fig. 13. Report on sales per location (city)

total standard cost, return on standard cost, profit margins on the total final cost, return on costs and margins on the last income) relating to categories of goods sold divided by time in years. You can still make the drill-down on the time dimension to see these data even at a monthly detail.

5) *Report on company's profit per customer:* Here statistics (with the same measures that were used for the previous question) is not made for only categories over the years, but adds additional information (dimension) on customers. As previously mentioned, you can limit the query to certain customers or groups.

tempo-categorie/misure	Qtàevasa	Totale Vendite	Totale Costo Standard		Utile su costo standard	Margin e (utile su c.st.) su p.v	Margin e (utile su c.st.) su p.v
2009	13.325,65	€ 94.650.175,22	€ 87.426.618,80	€ 7.223.556,42	€ 8,26%	€ 7,63%	
2	801.555	€ 3.388.016,17	€ 3.089.325,47	€ 298.690,70	9,67%	8,82%	
3	29.492	€ 325.843,82	€ 285.024,92	€ 40.818,90	14,32%	12,53%	
4	882.448	€ 5.309.189,41	€ 4.626.472,06	€ 682.717,35	14,76%	12,86%	
5	1.224.144	€ 6.029.737,18	€ 5.250.257,22	€ 779.479,96	14,85%	12,93%	
6	449.449	€ 3.049.227,79	€ 2.571.832,38	€ 477.395,41	18,56%	15,66%	
7	146.282	€ 504.338,15	€ 326.703,40	€ 177.634,75	54,37%	35,22%	
A	9.458.627	€ 74.459.672,84	€ 69.797.413,82	€ 4.662.259,02	6,68%	6,26%	
B	333.658	€ 1.584.149,86	€ 1.479.589,53	€ 104.560,33	7,07%	6,60%	
Totale complessivo	13.325,65	€ 94.650.175,22	€ 87.426.618,80	€ 7.223.556,42	8,26%	7,63%	

tempo-categorie/misure	Totale Costo Ultimo	Utile su costo ultimo		Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.
2009	€ 86.470.990,06	€ 8.179.185,16	9,46%		8,64%
2	€ 3.170.835,99	€ 217.180,18	6,85%		6,41%
3	€ 291.066,33	€ 34.777,49	11,95%		10,67%
4	€ 4.664.205,39	€ 644.984,02	13,83%		12,15%
5	€ 5.270.372,77	€ 759.364,41	14,41%		12,59%
6	€ 2.613.007,63	€ 436.220,16	16,69%		14,31%
7	€ 337.945,78	€ 166.392,37	49,24%		32,99%
A	€ 68.609.110,71	€ 5.850.562,13	8,53%		7,86%
B	€ 1.514.445,46	€ 69.704,40	4,60%		4,40%
Totale complessivo	€ 86.470.990,06	€ 8.179.185,16	9,46%		8,64%

Fig. 14. Report on annual sales per category

6) *Report on sales by category and time:* In this case the list is not meant primarily to the time dimension, but by category. You can then study for example the trend of the measures under analysis over time. In this case we have only the year 2009 for ease of study, but in reality the DW presents data for the past 11 years and therefore this type of observation would be more significant.

7) *Report on company's profit by category and supplier:* This query is similar to the previous ones, with the only variation that – while before the “customers” dimension was selected – now we select suppliers. Then you can analyze the above statistics for each year (or certain periods of time entered by the user) for each category (or certain categories, just one) and finally for suppliers.

As we have seen, analysis carried out on data in the DW are very similar to those made by queries on relational databases. The disadvantages are that this system presents, as repeatedly stated, very long response times and it is almost impossible to perform more complex queries with it. While we noted that the analysis carried out on DW were all facing the entire population of data in it (in the sense that were selected all “members” of the dimensions set as cube “faces”). The problems of a system based on relational databases have been resolved without making a significant investment in more powerful (but expensive) computers, which might not solve them at all.

XI. CONCLUSIONS

Market analysis is a critical task for company's executives. One of the most significant improvements offered by the Data Warehouse is the speed with which you can understand

tempo - categoria - soci/misure	Qtà/versa	Totale Vendite	Totale Costo Standard	Utile su costo standard	Margine (utile su c.st.) su c.st.	Margine (utile su c.st.) su p.v
2009	13.325.655	€94650175,22	€87426618,8	€7223556,42	8,26%	7,63%
2	801.555	€3388016,17	€3089325,47	€298690,7	9,67%	8,82%
100	3.633	€14268,94	€12476,98	€1791,96	14,36%	12,56%
1002	1.231	€5407,11	€4493,57	€913,54	20,33%	16,90%
1004	3.040	€13134,63	€11700,36	€1434,27	12,26%	10,92%
1006	1.502	€6797,43	€6098,61	€698,82	11,46%	10,28%
1007	12	€24	€29,76	€-5,76	-19,35%	-24,00%
1008	1.167	€5322,86	€4504,02	€818,84	18,18%	15,38%
1009	1.137	€5329,99	€4749,91	€580,08	12,21%	10,88%
1010	2.285	€9858,06	€8300	€1558,06	18,77%	15,80%
[...]						
B	333.658	€1584149,86	€1479589,53	€104560,33	7,07%	6,60%
100	2.231	€8591,13	€8256,44	€334,69	4,05%	3,90%
1002	382	€1943,47	€1757,71	€185,76	10,57%	9,56%
1004	771	€3606	€3441,23	€164,77	4,79%	4,57%
1006	971	€4664,24	€4199,14	€465,1	11,08%	9,97%
1008	1.273	€6621,96	€6241,14	€380,82	6,10%	5,75%
1009	1.009	€5081,79	€4863,11	€218,68	4,50%	4,30%
1010	579	€2784,23	€2475,33	€308,9	12,48%	11,09%
tempo - categoria - soci/misure	Totale Costo Ultimo	Utile su costo ultimo	Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.		
2009	€86470990,06	€8179185,16		9,46%	8,64%	
2	€3170835,99	€217180,18		6,85%	6,41%	
100	€12581,56	€1687,38		13,41%	11,83%	
1002	€4640,95	€766,16		16,51%	14,17%	
1004	€11945,73	€1188,9		9,95%	9,05%	
1006	€6209,06	€588,37		9,48%	8,66%	
1007	€29,76	€-5,76		-19,35%	-24,00%	
1008	€4714,95	€607,91		12,89%	11,42%	
1009	€4782,54	€547,45		11,45%	10,27%	
1010	€8430,07	€1427,99		16,94%	14,49%	
[...]						
B	€1514445,46	€69704,4		4,60%	4,40%	
100	€8394,05	€197,08		2,35%	2,29%	
1002	€1799,74	€143,73		7,99%	7,40%	
1004	€3516,47	€89,53		2,55%	2,48%	
1006	€4350,94	€313,3		7,20%	6,72%	
1008	€6385,37	€236,59		3,71%	3,57%	
1009	€4946,09	€135,7		2,74%	2,67%	
1010	€2568,74	€215,49		8,39%	7,74%	

Fig. 15. Report on company's profit per customer

the current market trends and make predictions on its future performance, with response times much faster than other systems based on relational databases.

One of the main purposes of the Data Warehouse is also to improve the productivity of personnel involved in strategic company decisions. The areas where these improvements are most obvious are:

- forecast of the market performance;
- analysis of customer requirements;
- analysis of the competitive situation;
- workload of the technical staff.

Understanding and quickly satisfying customer needs is an essential ingredient of any company. Market competition becomes increasingly fierce, and it is therefore vital to understand what strategies are needed to cope with them and remain in a favorable position in the market. Through external data, the Data Warehouse allows you to analyze prices, offers, competitive attitude, and then to formulate your strategy to market approach.

categoria - data/misure	Qtà/versa	Totale Vendite	Totale Costo Standard	Utile su costo standard	Margine (utile su c.st.) su c.st.	Margine (utile su c.st.) su p.v
2	801.555	€3388016,17	€3089325,47	€298690,7	9,67%	8,82%
2009	801.555	€3388016,17	€3089325,47	€298690,7	9,67%	8,82%
3	29.492	€325843,82	€285024,92	€40818,9	14,32%	12,53%
2009	29.492	€325843,82	€285024,92	€40818,9	14,32%	12,53%
4	882.448	€5309189,41	€4626472,06	€682717,35	14,76%	12,86%
2009	882.448	€5309189,41	€4626472,06	€682717,35	14,76%	12,86%
5	1.224.144	€6029737,18	€5250257,22	€779479,96	14,85%	12,93%
2009	1.224.144	€6029737,18	€5250257,22	€779479,96	14,85%	12,93%
6	449.449	€3049227,79	€2571832,38	€477395,41	18,56%	15,66%
2009	449.449	€3049227,79	€2571832,38	€477395,41	18,56%	15,66%
7	146.281	€504200,15	€326578,4	€177621,75	54,39%	35,23%
2009	146.281	€504200,15	€326578,4	€177621,75	54,39%	35,23%
A	9.458.627	€74459672,84	€69797413,82	€4662259,02	6,68%	6,26%
2009	9.458.627	€74459672,84	€69797413,82	€4662259,02	6,68%	6,26%
B	333.658	€1584149,86	€1479589,53	€104560,33	7,07%	6,60%
2009	333.658	€1584149,86	€1479589,53	€104560,33	7,07%	6,60%
Totale complessivo	13.325.654	€94650037,22	€87426493,8	€7223543,42	8,26%	7,63%
categoria - data/misure	Totale Costo Ultimo	Utile su costo ultimo	Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.		
2	€3170835,99	€217180,18		6,85%	6,41%	
2009	€3170835,99	€217180,18		6,85%	6,41%	
3	€291066,33	€34777,49		11,95%	10,67%	
2009	€291066,33	€34777,49		11,95%	10,67%	
4	€4664205,39	€644984,02		13,83%	12,15%	
2009	€4664205,39	€644984,02		13,83%	12,15%	
5	€5270372,77	€759364,41		14,41%	12,59%	
2009	€5270372,77	€759364,41		14,41%	12,59%	
6	€2613007,63	€436220,16		16,69%	14,31%	
2009	€2613007,63	€436220,16		16,69%	14,31%	
7	€337820,78	€166379,37		49,25%	33,00%	
2009	€337820,78	€166379,37		49,25%	33,00%	
A	€68609110,7	€5850562,13		8,53%	7,86%	
2009	€68609110,7	€5850562,13		8,53%	7,86%	
B	€1514445,46	€69704,4		4,60%	4,40%	
2009	€1514445,46	€69704,4		4,60%	4,40%	
Totale complessivo	€86470865,1	€8179172,16		9,46%	8,64%	

Fig. 16. Report on sales by category and time

Data available in a Data Warehouse should be used and it is here that the BI systems come into play: it is not enough to have a lot of data stored in a Data Warehouse, you must have the intelligence to turn this data into useful reports, dashboards, scorecards (which provide an intuitive summary of the data), and other forms of information. Let us analyze them briefly.

Reports can address a wide variety of business needs, can be customized and targeted, and can be easily distributed through many means.

Dashboards translate complex information into information with high visual impact. They help identify problems more quickly. More specifically, a dashboard is a collection of analysis and KPIs (Key Performance Indicator) reporting that provides users a single data view that helps them to monitor information associated with a task, project or goal.

Scorecards also deliver information at a glance, but they show how well you are doing compared to specific targets. They can help to raise awareness of decision makers. A scorecard is a collection of information in line with the strategic objectives of the organization,

tempo-categoria-ditta/misure	Qtà evasa	Totale Vendite	Totale Costo Standard	Utile su costo standard	Margine (utile su c.st.) su c.st.	Margine (utile su c.st.) su p.v
2009	13.325.654	€ 94.650.037	€ 87.426.494	€ 7.223.543	8,26%	7,63%
2	801.555	€ 3.388.016	€ 3.089.325	€ 298.691	9,67%	8,82%
1002	1	€ 10	€ 7	€ 3	36,34%	26,65%
1070	184	€ 604	€ 357	€ 247	69,03%	40,84%
1071	612	€ 2.598	€ 2.106	€ 491	23,32%	18,91%
1211	2.865	€ 16.658	€ 14.315	€ 2.343	16,37%	14,07%
1269	318	€ 1.258	€ 1.049	€ 209	19,95%	16,63%
1598	73	€ 308	€ 175	€ 133	75,77%	43,11%
162	742	€ 5.161	€ 3.841	€ 1.320	34,37%	25,58%
1646	3.904	€ 22.536	€ 20.266	€ 2.270	11,20%	10,07%
[...]						
B	333.658	€ 1.584.150	€ 1.479.590	€ 104.560	7,07%	6,60%
1070	23	€ 70	€ 31	€ 39	126,05%	55,76%
1071	2.963	€ 19.616	€ 18.661	€ 956	5,12%	4,87%
1116	1.091	€ 6.537	€ 6.336	€ 201	3,17%	3,07%
1269	666	€ 3.902	€ 3.747	€ 155	4,14%	3,98%
1286	176	€ 1.201	€ 757	€ 444	58,69%	36,98%
1314	18.357	€ 139.398	€ 124.789	€ 14.609	11,71%	10,48%
1598	4.710	€ 28.811	€ 27.267	€ 1.544	5,66%	5,36%
1660	132	€ 1.192	€ 1.126	€ 66	5,84%	5,52%
1686	8.494	€ 78.288	€ 74.476	€ 3.812	5,12%	4,87%
[...]						
tempo-categoria-ditta/misure	Totale Costo Ultimo	Utile su costo ultimo	Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.		
2009	€ 86.470.865	€ 8.179.172		9,46%	8,64%	
2	€ 3.170.836	€ 217.180		6,85%	6,41%	
1002	€ 7	€ 3		36,34%	26,65%	
1070	€ 357	€ 247		69,03%	40,84%	
1071	€ 2.158	€ 440		20,39%	16,94%	
1211	€ 14.315	€ 2.343		16,37%	14,07%	
1269	€ 1.113	€ 145		13,00%	11,50%	
[...]						
B	€ 1.514.445	€ 69.704		4,60%	4,40%	
1070	€ 31	€ 39		126,05%	55,76%	
1071	€ 18.661	€ 956		5,12%	4,87%	
1116	€ 6.334	€ 202		3,19%	3,09%	
1269	€ 3.841	€ 62		1,61%	1,59%	
1286	€ 1.125	€ 76		6,78%	6,35%	
1314	€ 140.202	-€ 803		-0,57%	-0,58%	
[...]						

Fig. 17. Report on company's profit by category and supplier

allowing decision makers to know how their goals and their activities are related to business goals.

Analysis allows to explore the information coming from the entire company, helping to dig deeper to discover the “why” behind what you’re seeing in a report, dashboard or scorecard.

As we have seen, the Data Warehouse is not a point of arrival but of departure for meaningful analysis on huge amounts of data that are registered by companies. It is a necessary condition so that we can implement a business intelligence application to perform a more effective and efficient analysis than that of companies’ traditional data storage systems. What is vital to business success is not only the proper recording of data, but are fundamental the analyses you can extrapolate from them, too. The better this analysis, the more correct decisions are taken by a manager. The success and continuous improvement of a company go through a good implementation of a BI system to support decision makers.

REFERENCES

- [1] Don Jones, *The shortcut guide to achieving business intelligence in midsized companies* (sponsored by IBM), Real Time Publishers.
- [2] E.F. Codd (1990), *The relational model for database management: version 2*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, ISBN 0-201-14192-2.
- [3] R.G.G. Cattell (1994), *Object data management: Object-oriented and extended relational database systems*, Addison-Wesley.
- [4] G.M. Nijssen, and T.A. Halpin (1989), *Conceptual schema and relational database design: a fact oriented approach*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, ISBN 0-13-167263-0.
- [5] Chaudhuri, Surajit and Dayal, Umeshwar, (1997), An overview of data warehousing and OLAP technology, *SIGMOD Rec.* **26**:1(65–74), ISSN 0163-5808, ACM, New York, NY, USA.
- [6] W.H. Inmon, (1992), *Building the Data Warehouse*, John Wiley & Sons, Inc., New York, NY, USA, ISBN 0471569607.
- [7] A. Berson, and S.J. Smith (1997), *Data Warehousing, Data Mining, and Olap*, McGraw-Hill, Inc., New York, NY, USA, ISBN 0070062722.



Crescenzo Gallo was born in Carapelle (FG), Italy, in 1956. He received the B.S. in Computer Science (with honors) from the University of Bari, Italy, in 1978 and the M.S. in Information and Communication Technology (with honors) from the University of Milan, Italy, in 2007. During 1978-1980, he stayed with the Institute of Informatics (ISI), Bari, Italy as a researcher in Information Systems, and with Telespazio SpA, Rome, Italy participating to Landsat satellite projects of Italian (ASI) and European Space Agency (ESA). From 1982 to 2003 he has been a high school full-time teacher in Computer Science at Foggia, Italy, and since 1993 he has been a contract professor of Computer Science at the University of Foggia, Italy. Since January 2004 he is an Assistant Professor in Information Science and Technology at the Dept. of Economics, Mathematics and Statistics, University of Foggia, Italy. His primary research interests include information theory and algorithms, with special emphasis on artificial neural networks. Dr. Gallo is an IEEE Member since 1998, and an ACM professional member since 2004.



Michelangelo De Bonis was born in San Giovanni Rotondo, FG, Italy, in 1975. He graduated in Computer Engineering at the Polytechnic of Turin in 2000. He is an IEEE member and a certified Cisco instructor, teaches Informatics in high-school courses, and is a consultant on networks and information security. His primary research interests are about Artificial Neural Networks and Mobile Ad Hoc Networks, on which actually he collaborates at the Department of Economics, Mathematics and Statistics of the University of Foggia, Italy.



Michele Perilli was born in Foggia, Italy, in 1962. He received a degree in Computer Science from the University of Bari, Italy, in 1987, and degrees in Science and Information Technology (2005) and Information and Communication Technology (2006) from the University of Milano, Italy. During 1988-2000 he worked in Telecom Italia S.p.A. (Italian Carrier telephone company). He is a high school full time professor of informatics since 2001, and a contract professor at the University of Foggia, Italy, since 2007. He is an IEEE member and a certified Cisco Systems instructor. His primary research interest are in networking, protocols and routing.