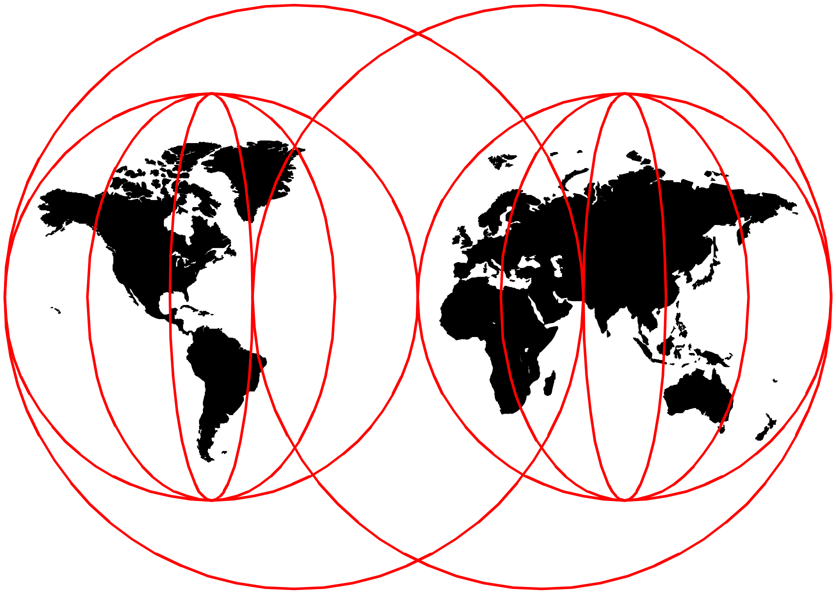
Business Intelligence Certification Guide

Joerg Reinschmidt, Allison Francoise



International Technical Support Organization

www.redbooks.ibm.com

SG24-5747-00

SG24-5747-00

International Technical Support Organization

Business Intelligence Certification Guide

January 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special notices” on page 131.

First Edition (January 2000)

This edition is intended to be product and operating system independent.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

The IBM Professional Certification Program offers the certification to become an IBM Certified Solutions Expert — Business Intelligence. If you are knowledgeable about IBM's Business Intelligence solutions and the fundamental concepts of DB2 Universal Database, and you are capable of performing the intermediate and advanced skills required to design, develop, and support Business Intelligence applications, you may benefit from this certification role.

This role is applicable to experts who qualify Business Intelligence opportunities, identify the business and technical requirements, and consult, architect and manage Business Intelligence solutions. This is a software-based test that is non-platform, non-product specific, for use by consultants and implementors.

The core requirement for this certification consists of two tests: Test 503, DB2 UDB V5 Fundamentals **or** Test 509, DB2 UDB V6.1 Fundamentals, and Test 515, Business Intelligence Solutions. For the objectives of this test, see the URL: `http://www.ibm.com/education/certify/tests/obj515.phtml`. The ordering of core tests is recommended but not required.

To help you become well prepared to perform this certification, this redbook provides all the information required to pass the Business Intelligence Solutions test.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization San Jose Center.

Joerg Reinschmidt is an Information Mining and Knowledge Management Specialist at the International Technical Support Organization, San Jose Center. He writes extensively and teaches IBM classes worldwide on Information Mining, Knowledge Management, DB2 Universal Database, and Internet access to legacy data. Before joining the ITSO in 1998, Joerg worked in the IBM Solution Partnership Center (SPC) in Germany as a DB2 Specialist, supporting independent software vendors (ISVs) to port their applications to use IBM data management products.

Allison Francoise is a Business Intelligence Specialist at the Business Intelligence Solution Center (BISC) in Dallas, Texas. She has 4 years of experience in Business Intelligence, implementing data warehouse proof-of-concept on the AS/400. She holds a degree in Mathematics from Morgan State University, Baltimore, Maryland. Allison is currently involved in implementation of Data Propagator on the AS/400.

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO redbook evaluation” on page 151 to the fax number shown on the form.
- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Business Intelligence — an introduction

This chapter gives an introduction to Business Intelligence (BI) that talks about the history of BI, a definition, the main BI related terms, implementation methods, and data warehousing components.

1.1 Certification test objectives

But first, here is information on the certification test. The test is divided into four sections with the following contents (and their distribution to the overall test):

Section 1 — BI Terms, Functions, and Differentiators (29%)

- Define Business Intelligence terms.
- Differentiate features and functions of data marts from those of a data warehouse.
- Articulate the benefits of business intelligence (for example: provide timely information, extend the use of query tools, help to understand your business, extend the number of people who can make use of it).
- Explain scheduling options (time/event/function).
- Illustrate how network communications impacts Business Intelligence architecture.
- Differentiate multi-dimensional database versus relational database warehouse.
- Differentiate between operational data store and data warehouse for update/refresh situations.
- Given BI data and customer requirement criteria, select appropriate tools to perform transformation, extraction, data modeling, data cleansing, loading, and propagation.
- Describe metadata management techniques and processes.
- Differentiate metadata from database data, database encryption, tool metadata, stored procedures.
- Given BI data and customer needs, implement metadata strategy.
- Given BI data, determine appropriate analysis techniques (Intelligent Miner, query, cluster, trend, discovery, predictive, explanatory, visualization).

- Given BI data and customer requirement criteria, select appropriate visualization and presentation techniques (charts, maps, reports, tables, agent-driven, messaging).
- Given customer requirement criteria, select appropriate front-end features based on criteria, such as presentation, level of interactivity, Web-versus-FAT client, static versus dynamic, and user skill level.

Section 2 — BI Customer Requirements Collection (23%)

- Identify business requirements of the customer as they relate to a Business Intelligence solution.
- Define the business goals and objectives of the customer.
- Determine number of users, types of queries, number of queries, user work tables.
- Evaluate existing hardware and software environment.
- Identify constraints (financing, political, timing, legal, technical, competition, legacy issues, customer skill level, and so on).
- Determine customer growth requirements.
- Identify geographical constraints (human language issues, computer language issues, currency issues, and so on).
- Identify critical success factors and how to validate them.
- Determine availability and recovery requirements (user uptime, maintenance windows, system maintenance, data mart, and data warehouse for update/refresh situations, disaster recovery, hardware/software failures, aggregation for data marts).

Section 3 — BI Data Sourcing/Movement (23%)

- Identify sources of data (operational, non-operational within the company, external to the company).
- Identify methods for extraction.
- Identify methods for transformation.
- Identify methods for cleansing.
- Identify methods for loading and load balancing.
- Define methods for data moving (differential, full-refreshed, incremental).
- Define methods for scheduling.
- Define methods for error detection and handling.
- Describe data modeling techniques and processes.

Section 4 — BI Solution Architecture (25%)

- Describe data warehouse management techniques and processes (conversion of data model to the data warehouse (DW); creation and loading of Enterprise Relational Data Warehouse; extraction and management of metadata; creation and loading of data marts from DW).
- Describe to a customer the infrastructure and organization necessary to support a data warehouse.
- Given source data tables, identify measures, attributes, and dimension.
- Given a BI scoped project, build logical data models (create fact tables, identify dimension, associate the attributes with dimensions).
- Given a BI business requirement, determine data that is applicable to the business question.
- Given a BI business requirement, transform it into a solution based on requirements, such as functionality, performance, security, size, availability, interface with front-end or other systems, connectivity, communications, and so on.
- Given a BI business requirement, select appropriate tools to build the solution (such as data extraction, scrubbing, transporting, transforming, data modeling, querying, business process modeling).
- Given a BI business requirement, select appropriate middleware tools.
- Given a BI business requirement, select appropriate presentation to the customer based on complexity of data, sophistication of tools, complexity of the data model, user skills, level of interactivity, deployment and accessibility, level of detail needed, and level of summarization needed.
- Given an architecture for a BI solution, determine the feasibility of the design based on customer performance expectations, growth requirements, and scalability factors.
- Given a Business Intelligence project, plan for education.

1.2 What is Business Intelligence?

Business intelligence is not business as usual. It’s about making better decisions easier and making them more quickly.

Businesses collect enormous amounts of data every day: information about orders, inventory, accounts payable, point-of-sale transactions, and of course, customers. Businesses also acquire data, such as demographics and mailing lists, from outside sources. Unfortunately, based on a recent survey, over 93% of corporate data is not usable in the business decision-making process today.

Consolidating and organizing data for better business decisions can lead to a competitive advantage, and learning to uncover and leverage those advantages is what business intelligence is all about.

The amount of business data is increasing exponentially. In fact, it doubles every two to three years. More information means more competition. In the age of the information explosion, executives, managers, professionals, and workers all need to be able to make better decisions faster. Because now, more than ever, time is money.

IBM Business Intelligence solutions are not about bigger and better technology — they are about delivering more sophisticated information to the business end user. BI provides an easy-to-use, shareable resource that is powerful, cost-effective and scalable to your needs.

Much more than a combination of data and technology, BI helps you to create knowledge from a world of information. Get the right data, discover its power, and share the value, BI **transforms information into knowledge**. Business Intelligence is the application of putting the **right information** into the hands of the **right user** at the **right time** to support the decision-making process.

1.3 Business driving forces

It can be noted that there are some business driving forces behind business intelligence, one being the need to improve ease-of-use and reduce the resources required to implement and use new information technologies. There are additional driving forces behind business intelligence, for example:

1. *The need to increase revenues, reduce costs, and compete more effectively*. Gone are the days when end users could manage and plan business operations using monthly batch reports, and IT organizations had months to implement new applications. Today companies need to deploy informational applications rapidly, and provide business users with easy and fast access to business information that reflects the rapidly changing business environment. Business intelligence systems are focused towards end user information access and delivery, and provide packaged business solutions in addition to supporting the sophisticated information technologies required for the processing of today's business information.
2. *The need to manage and model the complexity of today's business environment*. Corporate mergers and deregulation means that companies

today are providing and supporting a wider range of products and services to a broader and more diverse audience than ever before. Understanding and managing such a complex business environment and maximizing business investment is becoming increasingly more difficult. Business intelligence systems provide more than just basic query and reporting mechanisms, they also offer sophisticated information analysis and information discovery tools that are designed to handle and process the complex business information associated with today's business environment.

3. *The need to reduce IT costs and leverage existing corporate business information*. The investment in IT systems today is usually a significant percentage of corporate expenses, and there is a need not only to reduce this overhead, but also to gain the maximum business benefits from the information managed by IT systems. New information technologies like corporate intranets, thin-client computing, and subscription-driven information delivery help reduce the cost of deploying business intelligence systems to a wider user audience, especially information consumers like executives and business managers. Business intelligence systems also broaden the scope of the information that can be processed to include not only operational and warehouse data, but also information managed by office systems and corporate Web servers.

1.4 How to identify BI candidates?

The following discovery process will help you in assessing or identifying a candidate for business intelligence. The following section provides some questions that may help in the thought process — these questions are categorized by level of management and areas within a business, followed by some possible answers to the questions.

1.4.1 Senior executives of a corporation

When talking to a senior executive of a company, there are some questions that might help you to find out if this company is a prospect for a BI project in general, and whether the senior executive will be a supportive player during the process. Some of these questions are:

- How do you currently monitor the key or critical performance indicators of your business?
- How do you presently receive monthly management reports?
- How easily can you answer ad hoc questions with your current reporting systems?

- Can you quickly spot trends and exceptions in your business?
- Do you have to wait a long time (hours? days?) for answers to new questions?
- Is everyone on your management team working from the same information?

Depending on the response of an executive, there are certain needs that, if addressed in his responses, identify the executive as a BI project prospect. The answers to the previously mentioned questions would point to the following, if he is a candidate:

- Dissatisfaction is exhibited with the current reporting systems, especially in terms of flexibility, timeliness, accuracy, detail, consistency, and integrity of the information across all business users.
- Many people in the organization spend a lot of time re-keying numbers into spreadsheets.
- The senior executive is very vague about how key performance indicators are monitored.

1.4.2 IT vice presidents, directors, and managers

Addressing other, more technically-oriented executives, the questions to be asked would look like the following examples:

- How do your non-I/S end users analyze or report information?
- Do end users often ask IT to produce queries, reports, and other information from the database?
- Do end users frequently re-key data into spreadsheets or word processing packages?
- Does your production system suffer from a heavy volume of queries and reports running against the system?
- Would you like to see your end users receiving more business benefits from the IT organization?

The IT staff is a data warehousing prospect if the answers point to problem areas, such as:

- End users are relying on IT to perform most or all ad hoc queries and reports.
- End users have to re-key data into their spreadsheets on a regular basis.
- IT identifies end user dissatisfaction with the current reporting systems and processes.

- IT has a large backlog built up of end user requests for queries and reports.
- IT is concerned about end user queries and reports that are bogging down the production systems.

1.4.3 CFOs, financial vice presidents, and controllers

When talking to financially-oriented executives, there are some totally different questions to be asked to identify this part of the organization as an active supporter of a BI project. Some sample questions are shown below:

- How are your monthly management reports and budgets delivered and produced?
- How timely is that information?
- Do you spend more time preparing, consolidating, and reporting on the data, or on analyzing performance that is based on what the data has highlighted?
- Do all the company's executives and managers have a single view of key information to avoid inconsistency?
- How easy is it to prepare budgets and forecasts, and then to disseminate that critical information?
- Can you easily track variances in costs and overhead by cost center, product, and location?
- Is the year-end consolidation and reporting cycle a major amount of duplicated effort in data preparation and validation, and then in consolidation reporting?

The financial staff is a data warehousing prospect if the answers given to these questions are like these:

- Personnel like using spreadsheets, but they usually or often need to re-key or reformat data.
- They indicate in any way that their preferred reporting tool would be a spreadsheet if they did not have to constantly re-key great amounts of numbers into them.
- They admit that much time is spent in the production of reports and the gathering of information, with less time actually spent analyzing the data, and they can identify inconsistencies and integrity issues in the reports that have been produced.
- Budget collection is a painful and time consuming process and there is very little control available in the collection and dissemination process.

- The monthly management reports involve too much time and effort to produce and circulate, and do not easily allow queries and analysis to be run against them.
- Management information does not go into sufficient detail, especially in terms of expense control and overhead analysis.
- General dissatisfaction is expressed with the current information delivery systems.

1.4.4 Sales VPs, product managers, and customer service directors

After talking to the senior executive and to the technical and financial executives, there are some more possible sponsors for a BI project. These are the sales and marketing-oriented personnel, and their possible sponsorship may be evaluated with the following questions:

- How do you perform ad hoc analysis against your marketing and sales data?
- How do you monitor and track the effectiveness of a marketing or sales promotion program?
- How do you re-budget or re-forecast sales figures and margins?
- Do you have to wait a long time (days? weeks?) for sales management information to become available at month or quarter-end?
- How do you track best/worst performance of product/customers, and how do you monitor/analyze product/customer profitability?
- Do you know your customers' profiles: buying patterns, demographics, and so on?
- Are you and your staff using spreadsheets a lot, and re-keying great amounts of data?

The sales and marketing staff is a BI prospect if:

- Current reporting is very static and ad hoc requests must be accomplished through IT.
- Profitability versus volume and value cannot be easily analyzed, and the measurement of data is inconsistent; for example, there might be more than one way of calculating margin, profit, and contribution.
- There is no concept of re-planning and re-budgeting as it is too difficult to accomplish with the current systems.
- Suppliers cannot be provided with timely information, so it is very difficult to achieve reviews of their performance.

- Getting down to the right level of detail is impossible: for example, to the SKU level in a retail store.
- General dissatisfaction is expressed with the current process of information flow and management.

1.4.5 Operations and production management

The last group to be covered within this section is the management of operations and production. Their support can be evaluated by asking questions like these:

- How is the validity of the MRP model checked and how accurate do you think it really is?
- How do you handle activity based costing?
- How do you do handle ad hoc analysis and reporting for raw materials, on-time, and quality delivery?
- How do you handle production line efficiency, machine, and personnel efficiency?
- How do you evaluate personnel costs and staffing budgets?
- How do you handle shipments and returns, inventory control, supplier performance, and invoicing?

The operations and production staff is a DW prospect if:

- New projects cannot easily be costed out, and trends in quality, efficiency, cost, and throughput cannot be analyzed.
- The preferred access to information would be via a spreadsheet or an easy-to-use graphical user interface.
- Currently there is access to daily information only, which means much re-keying into spreadsheets for trending analysis and so on is required.
- The MRP model cannot easily be checked for accuracy and validity on a constant basis.

1.5 Main BI terms

Before we get into more detail about BI, this section will explain some of the terms related to Business Intelligence. As shown in Figure 1 on page 10, these definitions are very brief, and they will be explained in more detail later in the book.

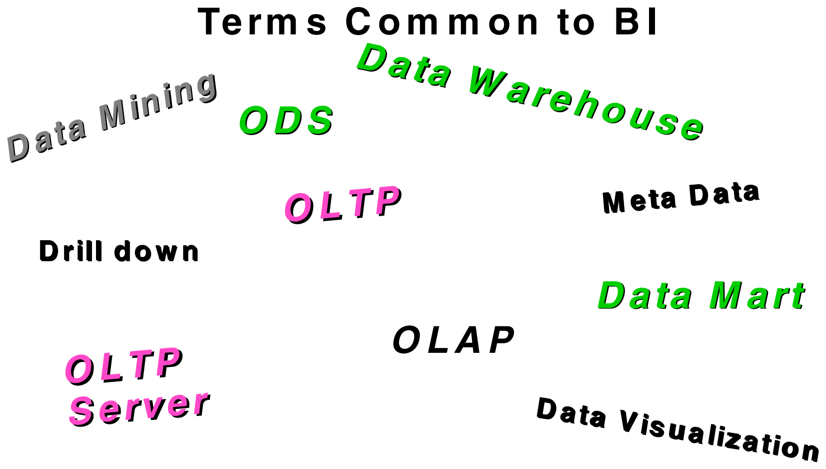


Figure 1. Common Business Intelligence terms

1.5.1 Operational databases

Operational databases are detail oriented databases defined to meet the needs of sometimes very complex processes in a company. This detailed view is reflected in the data arrangement in the database. The data is highly normalized to avoid data redundancy and "double-maintenance".

1.5.2 OLTP

On-Line Transaction Processing (OLTP) describes the way data is processed by an end user or a computer system. It is detail oriented, highly repetitive with massive amounts of updates and changes of the data by the end user. It is also very often described as the use of computers to run the on-going operation of a business.

1.5.3 Data warehouse

A data warehouse is a database where data is collected for the purpose of being analyzed. The defining characteristic of a data warehouse is its purpose. Most data is collected to handle a company's on-going business. This type of data can be called "operational data". The systems used to collect operational data are referred to as OLTP (On-Line Transaction Processing).

A data warehouse collects, organizes, and makes data available for the purpose of analysis — to give management the ability to access and analyze information about its business. This type of data can be called "informational data". The systems used to work with informational data are referred to as OLAP (On-Line Analytical Processing).

Bill Inmon coined the term "data warehouse" in 1990. His definition is:

"A (data) warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process."

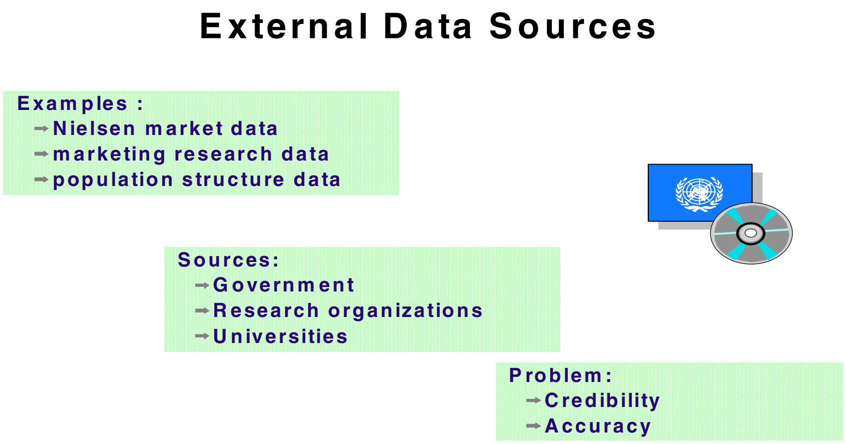
- **Subject-oriented** — Data that gives information about a particular subject instead of about a company's on-going operations.
- **Integrated** — Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.
- **Time-variant** — All data in the data warehouse is identified with a particular time period.

1.5.4 Data mart

A data mart contains a subset of corporate data that is of value to a specific business unit, department, or set of users. This subset consists of historical, summarized, and possibly detailed data captured from transaction processing systems, or from an enterprise data warehouse. It is important to realize that a data mart is defined by the functional scope of its users, and not by the size of the data mart database. Most data marts today involve less than 100 GB of data; some are larger, however, and it is expected that as data mart usage increases they will rapidly increase in size.

1.5.5 External data source

External data is data that can not be found in the OLTP systems but is required to enhance the information quality in the data warehouse. Figure 2 on page 12 shows some of these sources.



External Data Sources

Examples :
- Nielsen market data
- marketing research data
- population structure data

Sources:
- Government
- Research organizations
- Universities

Problem:
- Credibility
- Accuracy

Figure 2. External data sources

1.5.6 OLAP

On-Line Analytical Processing (OLAP) is a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user.

OLAP functionality is characterized by dynamic multi-dimensional analysis of consolidated enterprise data supporting end user analytical and navigational activities including:

- Calculations and modeling applied across dimensions, through hierarchies and/or across members
- Trend analysis over sequential time periods
- Slicing subsets for on-screen viewing
- Drill-down to deeper levels of consolidation
- Reach-through to underlying detail data
- Rotation to new dimensional comparisons in the viewing area

OLAP is implemented in a multi-user client/server mode and offers consistently rapid response to queries, regardless of database size and complexity. OLAP helps the user synthesize enterprise information through comparative, personalized viewing, as well as through analysis of historical and projected data in various "what-if" data model scenarios. This is achieved through use of an OLAP Server.

1.5.7 OLAP server

An OLAP server is a high-capacity, multi-user data manipulation engine specifically designed to support and operate on multi-dimensional data structures. A multi-dimensional structure is arranged so that every data item is located and accessed, based on the intersection of the dimension members that define that item. The design of the server and the structure of the data are optimized for rapid ad hoc information retrieval in any orientation, as well as for fast, flexible calculation and transformation of raw data based on formulaic relationships. The OLAP Server may either physically stage the processed multi-dimensional information to deliver consistent and rapid response times to end users, or it may populate its data structures in real-time from relational or other databases, or offer a choice of both. Given the current state of technology and the end user requirement for consistent and rapid response times, staging the multi-dimensional data in the OLAP Server is often the preferred method.

1.5.8 Metadata — a definition

Metadata is the kind of information that describes the data stored in a database and includes such information as:

- A description of tables and fields in the data warehouse, including data types and the range of acceptable values.
- A similar description of tables and fields in the source databases, with a mapping of fields from the source to the warehouse.
- A description of how the data has been transformed, including formulae, formatting, currency conversion, and time aggregation.
- Any other information that is needed to support and manage the operation of the data warehouse.

1.5.9 Drill-down

Drill-down can be defined as the capability to browse through information, following a hierarchical structure. A small sample is shown in Figure 3 on page 14.

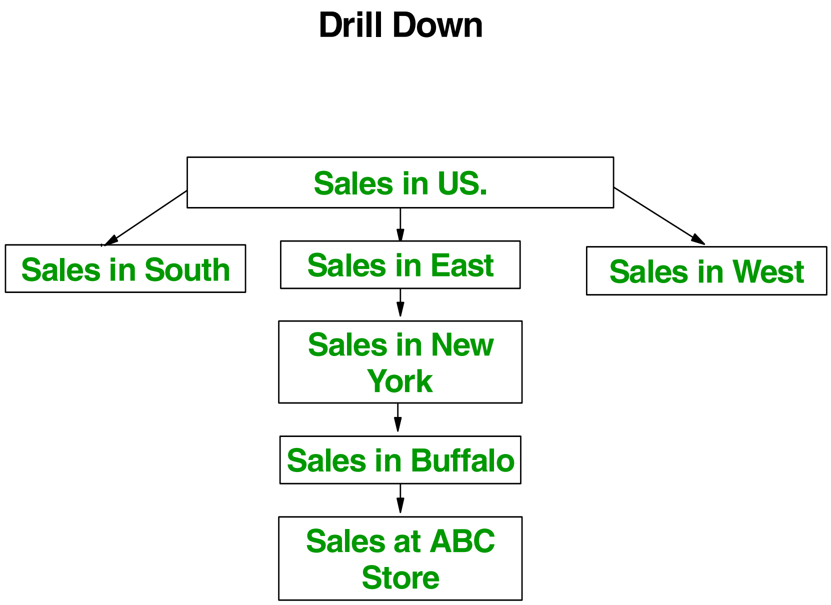


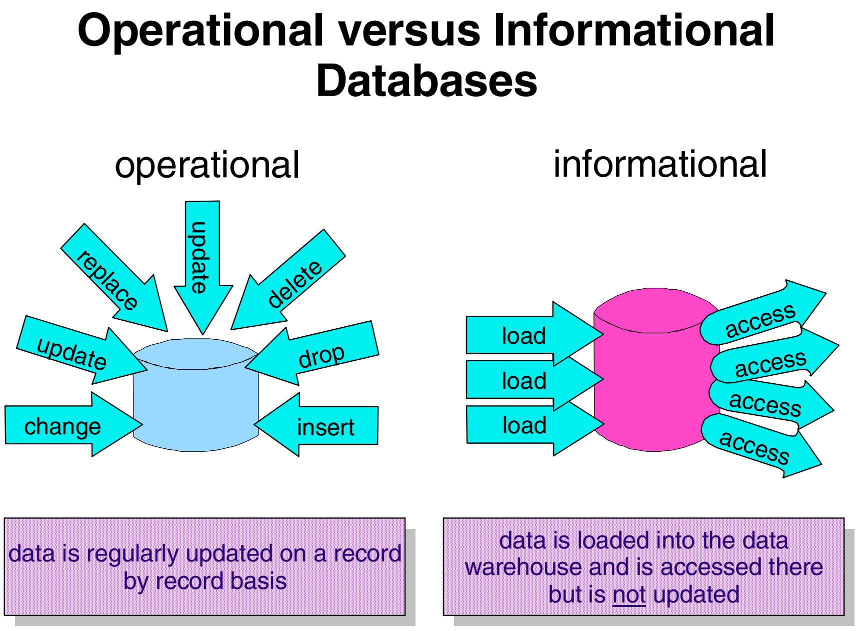
Figure 3. Drill-down

1.5.10 Operational versus informational databases

The major difference between operational and informational databases is the update frequency:

1. On operational databases a high number of transactions take place every hour. The database is always "up to date", and it represents a snapshot of the current business situation, or more commonly referred to as point in time.
2. Informational databases are usually stable over a period of time to represent a situation at a specific point in time in the past, which can be noted as historical data. For example, a data warehouse load is usually done overnight. This load process extracts all changes and new records from the operational database into the informational database. This process can be seen as one single transaction that starts when the first record gets extracted from the operational database and ends when the last data mart in the data warehouse is refreshed.

Figure 4 shows some of the main differences of these two database types.



Operational versus Informational Databases

data is regularly updated on a record by record basis

data is loaded into the data warehouse and is accessed there but is not updated

Figure 4. Operational versus informational databases

1.5.11 Data mining

Data mining is the process of extracting *valid*, *useful*, *previously unknown*, and *comprehensible* information from data and using it to make business decisions.

Chapter 2. BI implementations and warehouse concepts

The following chapter introduces different approaches that can be taken to implement a BI solution and shows some basic concepts of a data warehouse.

2.1 Different BI implementations

Different approaches have been made in the past to find a suitable way to meet the requirements for On Line Analytical Processing.

Figure 5 gives an overview of four major models to implement a decision support system.

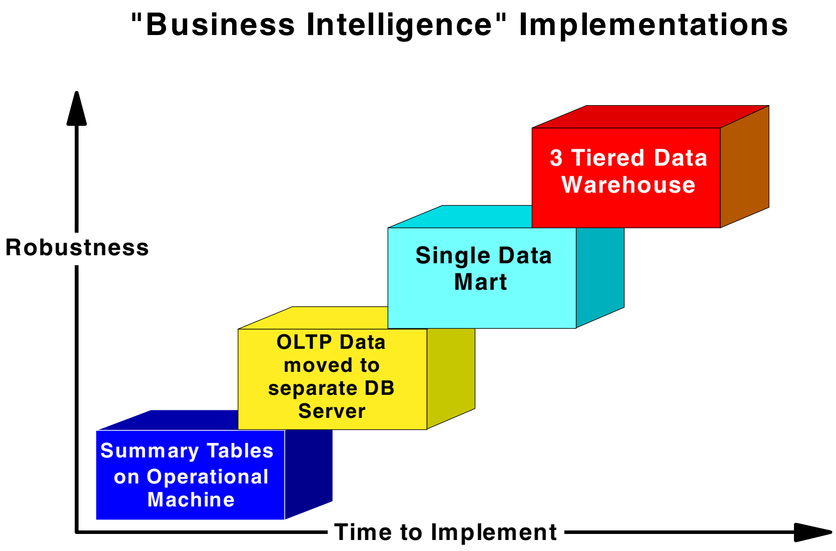

"Business Intelligence" Implementations
Robustness
3 Tiered Data Warehouse
Single Data Mart
OLTP Data moved to separate DB Server
Summary Tables on Operational Machine
Time to Implement

Figure 5. Business Intelligence implementations

The approaches shown are described below.

2.1.1 Summary table

A summary table on an OLTP system is the most common implementation that is already included in many standard software packages. Usually these summary tables cover only a certain set of requirements from business analysts. Figure 6 shows the advantages and disadvantages of this approach.

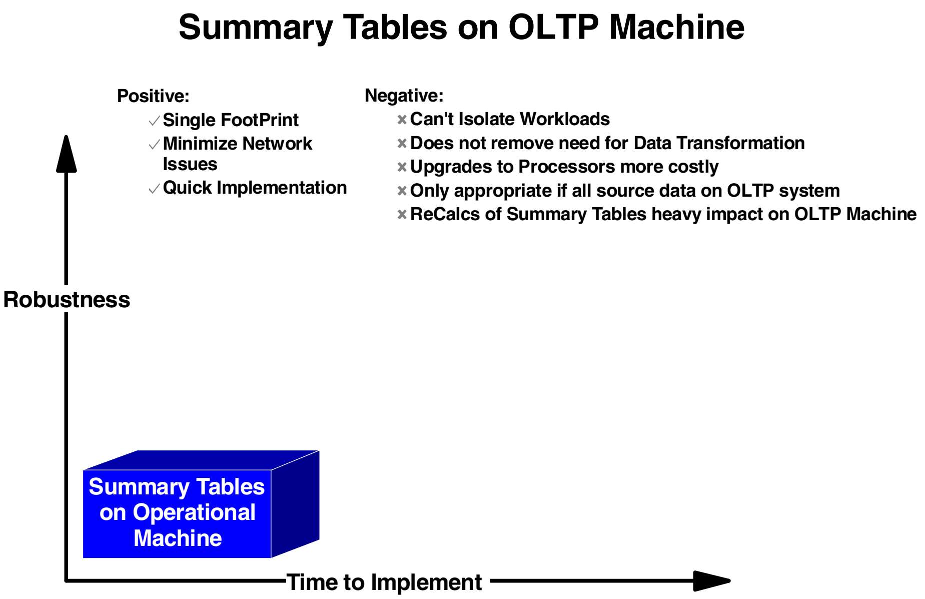
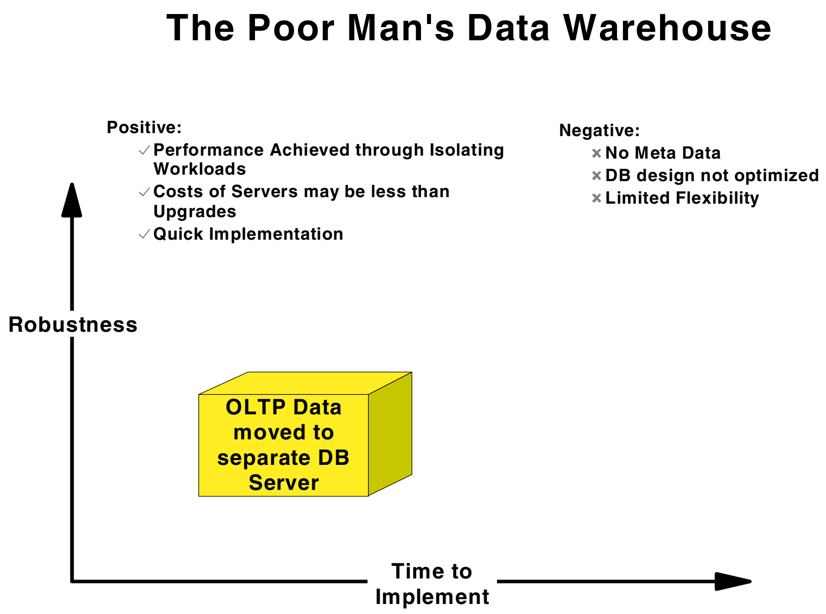

Summary Tables on OLTP Machine
Positive:
✓Single FootPrint
✓Minimize Network Issues
✓Quick Implementation
Negative:
×Can't Isolate Workloads
×Does not remove need for Data Transformation
×Upgrades to Processors more costly
×Only appropriate if all source data on OLTP system
×ReCalcs of Summary Tables heavy impact on OLTP Machine
Robustness
Summary Tables on Operational Machine
Time to Implement

Figure 6. Summary tables on OLTP

2.1.2 OLTP data at separate server

OLTP data moved to separate server — no changes in the database structure are made. This mirroring is a first step to offload the workload from the OLTP system to a separate dedicated OLAP machine. As long as no restructuring of the database takes place, this solution will not be able to track changes over time. Changes in the past can not be reflected in the database because the fields for versioning of slowly changing dimensions are missing. Figure 7 on page 19 shows this approach, sometimes called ‘A Poor Mans Data Warehouse’.



The Poor Man's Data Warehouse

Positive:

- ✓ Performance Achieved through Isolating Workloads
- ✓ Costs of Servers may be less than Upgrades
- ✓ Quick Implementation

Negative:

- × No Meta Data
- × DB design not optimized
- × Limited Flexibility

Figure 7. Poor man's data warehouse

The technique to move the original OLTP data regularly to a dedicated system for reporting purposes is a step that can be made to avoid the impact of long running queries on the operational system. In addition to the advantages in performance, security issues can be handled very easily in this architecture.

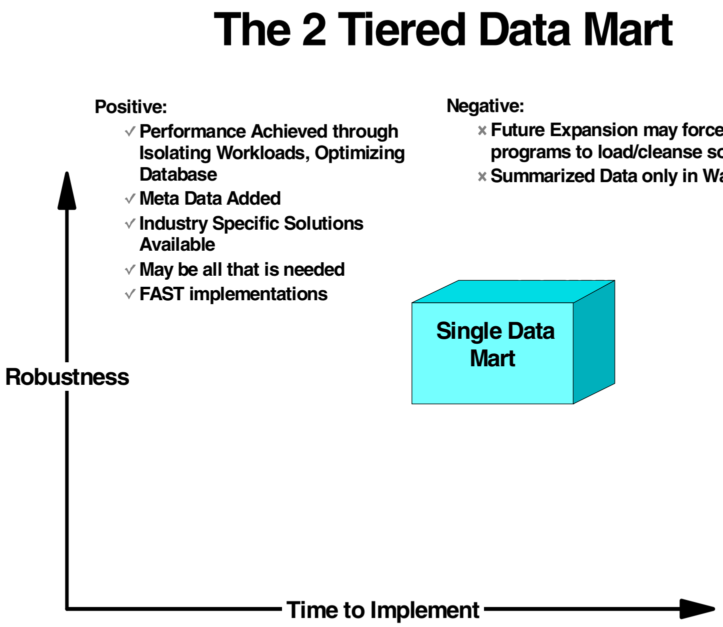
Totally isolated machines eliminate any interdependence between analysis and operational workload. The major problem that will still persist in this architecture is the fact that the database architecture has not changed or been optimized for query performance — the most detailed level of information is copied over to the dedicated analysis server.

The lack of summary tables or aggregations will result in long running queries with a high number of files and joins in every request. To build an architecture like this, file transfer or FTP can be sufficient for some situations.

2.1.3 Single data mart

A growing number of customers are implementing single data marts now to get the experiences with data warehousing. These single data marts are usually implemented as a proof of concept and keep growing over time. "A data warehouse has to be built — you cannot buy it!" This first brick in the data warehouse has to be kept under control — too many "single data marts" would create an administration nightmare.

The **two tiered** model of creating a single data mart on a dedicated machine includes more preparation, planning and investment. Figure 8 shows this approach.



The 2 Tiered Data Mart

Positive:

- ✓ Performance Achieved through Isolating Workloads, Optimizing Database
- ✓ Meta Data Added
- ✓ Industry Specific Solutions Available
- ✓ May be all that is needed
- ✓ FAST implementations

Negative:

- × Future Expansion may force new programs to load/cleanse source data
- × Summarized Data only in Warehouse

Figure 8. 2-tiered data mart

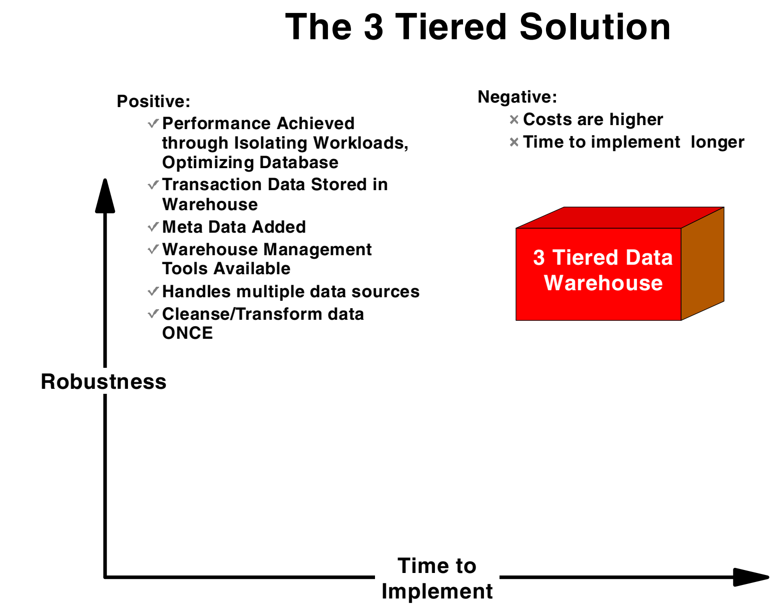
The major benefits of this solution compared to the other models are in performance, precalculated and aggregated values, higher flexibility to add additional data from multiple systems and OLTP applications, and better capabilities to store historical data.

Metadata can be added to the data mart to increase the ease-of-use and the navigation through the information in the informational database.

The implementation of a stand alone data mart can be done very quickly as long as the scope of the information to be included in the data mart is precisely limited to an adequate number of data elements.

The **three-tiered** data warehouse model consists of three stages of data stored on the system(s) (shown in Figure 9):

- OLTP data in operational databases
- Extracted, detailed, denormalized data organized in a Star-Join Schema to optimize query performance.
- Multiple aggregated and precalculated data marts to present the data to the end user.



The 3 Tiered Solution

Positive:

- ✓ Performance Achieved through Isolating Workloads, Optimizing Database
- ✓ Transaction Data Stored in Warehouse
- ✓ Meta Data Added
- ✓ Warehouse Management Tools Available
- ✓ Handles multiple data sources
- ✓ Cleanse/Transform data ONCE

Negative:

- × Costs are higher
- × Time to implement longer

Figure 9. 3-tiered data mart

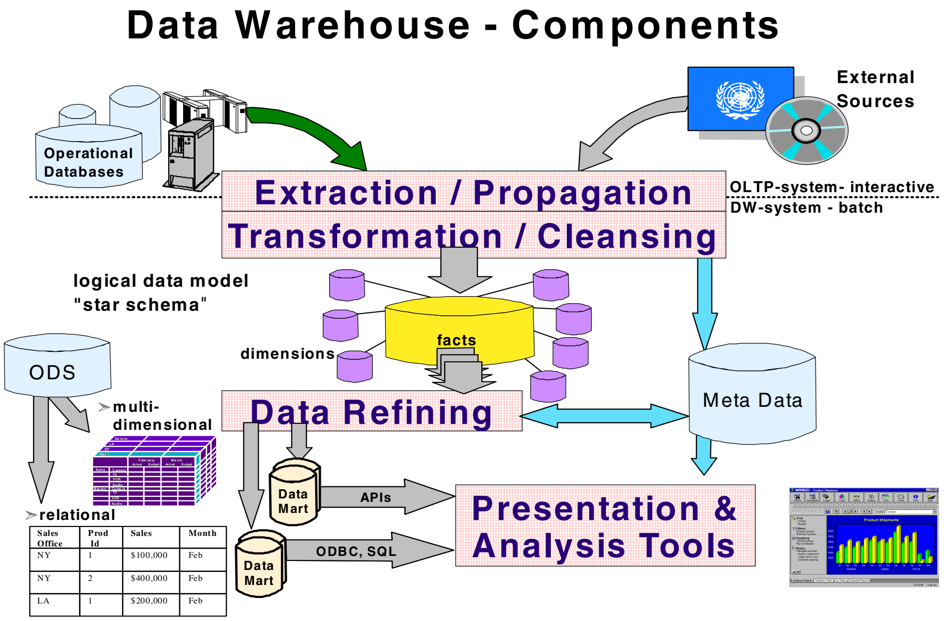
The characteristics of this model are:

- Departmental data marts to hold data in an organizational form that is optimized for specific requests — new requirements usually require the creation of a new data mart, but have no further influence on already existing components of the data warehouse.
- Historical changes over time can be kept in the data warehouse.
- Metadata is the major component to guarantee success of this architecture — ease-of-use and navigation support for end users.
- Cleansing and transformation of data is implemented at a single point in the architecture.

- The three different stages in aggregating/transforming data offer the capability to perform data mining tasks in the extracted, detailed data without creating workload on the operational system.
- Workload created by analysis requests is totally offloaded from the OLTP system.

2.2 Data warehouse components

Figure 10 shows the entire data warehouse architecture in a single view. The following sections will concentrate on single parts of this architecture and explain them in detail.



Sales Office	Prod Id	Sales	Month
NY	1	$100,000	Feb
NY	2	$400,000	Feb
LA	1	$200,000	Feb

Figure 10. Data warehouse components

This figure shows the following ideas:

- The processes required to keep the data warehouse up to date as marked are extraction/propagation, transformation/cleansing, data refining, presentation, and analysis tools.
- The different stages of aggregation in the data are: OLTP data, ODS Star-Join Schema, and data marts.

- Metadata and how it is involved in each process is shown with solid connectors.

The horizontal dotted line in the figure separates the different tasks into two groups:

- Tasks to be performed on the dedicated OLTP system are optimized for interactive performance and to handle the transaction oriented tasks in the day-to-day-business.
- Tasks to be performed on the dedicated data warehouse machine require high batch performance to handle the numerous aggregation, precalculation, and query tasks.

2.2.1 Data sources

Data sources can be operational databases, historical data (usually archived on tapes), external data (for example, from market research companies or from the Internet), or information from the already existing data warehouse environment. The data sources can be relational databases from the line of business applications. They also can reside on many different platforms and can contain structured information, such as tables or spreadsheets, or unstructured information, such as plain text files or pictures and other multimedia information.

2.2.2 Extraction/propagation

Data extraction / data propagation is the process of collecting data from various sources and different platforms to move it into the data warehouse. Data extraction in a data warehouse environment is a selective process to import decision-relevant information into the data warehouse.

Data extraction / data propagation is much more than mirroring or copying data from one database system to another. Depending on the technique, this process is either:

- **Pulling** (Extraction) or
- **Pushing** (Propagation)

2.2.3 Transformation/cleansing

Transformation of data usually involves code resolution with mapping tables (for example, changing *0* to *female* and *1* to *male* in the gender field) and the resolution of hidden business rules in data fields, such as account numbers. Also the structure and relationships of the data are adjusted to the analysis domain. Transformations occur throughout the population process, usually in more than one step. In the early stages of the process, the transformations are used more to consolidate the data from different sources, whereas, in the later stages the data is transformed to suit a specific analysis problem and/or tool.

Data warehousing turns data into information, on the other hand, **cleansing** ensures that the data warehouse will have valid, useful, and meaningful information. Data cleansing can also be described as standardization of data. Through careful review of the data contents, the following criteria are matched:

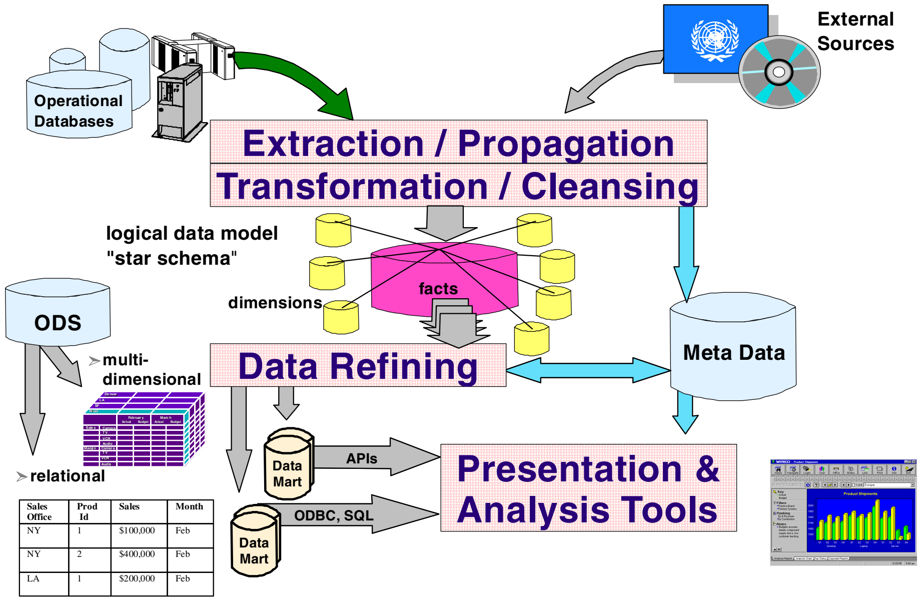
- Correct business and customer names
- Correct and valid addresses
- Usable phone numbers and contact information
- Valid data codes and abbreviations
- Consistent and standard representation of the data
- Domestic and international addresses
- Data consolidation (one view), such as house holding and address correction

2.2.4 Data refining

Data refining is creating subsets of the enterprise data warehouse, which have either a multidimensional or a relational organization format for optimized OLAP performance. Figure 11 on page 25 shows where this process is located within the entire BI architecture.

The atomic level of information from the star schema needs to be aggregated, summarized, and modified for specific requirements. This data refining process generates data marts that:

- Create a subset of the data in the star schema.
- Create calculated fields / virtual fields.
- Summarize the information.
- Aggregate the information.



Sales Office	Prod Id	Sales	Month
NY	1	$100,000	Feb
NY	2	$400,000	Feb
LA	1	$200,000	Feb

Figure 11. Data refining

This layer in the data warehouse architecture is needed to increase the query performance and minimize the amount of data that is transmitted over the network to the end user query or analysis tool.

When talking about data transformation/cleansing, there are basically two different ways the result is achieved. These are:

- **Data aggregation**: Change the level of granularity in the information.

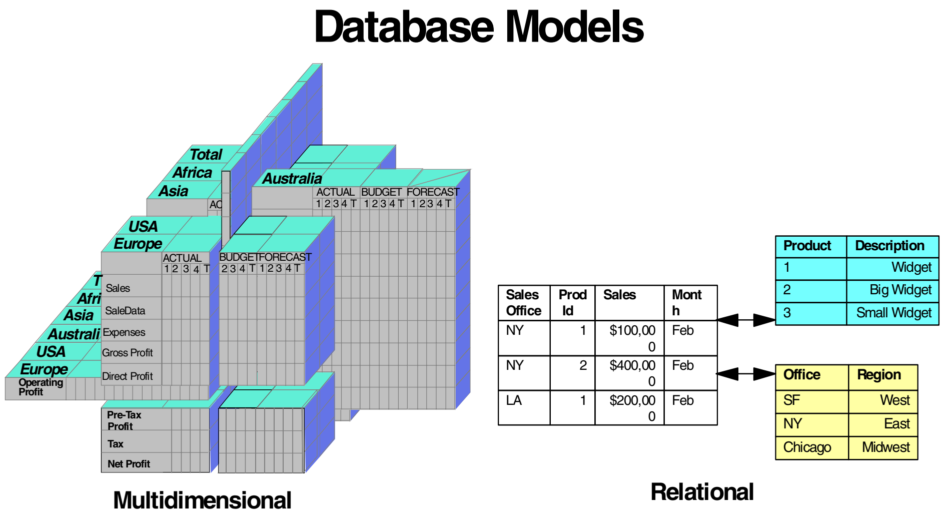
  Example: The original data is stored on a daily basis — the data mart contains only weekly values. Therefore, data aggregation results in less records.

- **Data summarization**: Add up values in a certain group of information.

  Example: The data refining process generates records that contain the revenue of a specific product group, resulting in more records.

2.2.5 Physical database model

In BI, talking about the physical data model is talking about relational or multidimensional data models. Figure 12 on page 26 shows the difference between those two physical database models.



Sales Office	Prod Id	Sales	Month
NY	1	$100,000	Feb
NY	2	$400,000	Feb
LA	1	$200,000	Feb

Product	Description
1	Widget
2	Big Widget
3	Small Widget

Office	Region
SF	West
NY	East
Chicago	Midwest

Figure 12. Physical database models

Both database architectures can be selected to create departmental data marts, but the way to access the data in the databases is different:

- To access data from a **relational** database, common access methods like SQL or middleware products like ODBC can be used.
- **Multidimensional** databases require specialized APIs to access the usually proprietary database architecture.

2.2.6 Logical database model

In addition to the previously mentioned physical database model, there also is a certain logical database model. When talking about BI, the most commonly used logical database model is the **Star-Join Schema**. The Star-Join Schema consists of two components shown in Figure 13:

- Fact tables
- Dimension tables

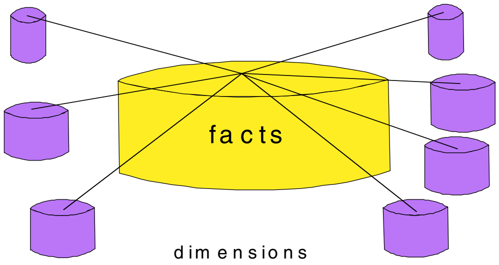


Figure 13. Logical data model

The following is a definition for those two components of the Star-Join Schema:

- Fact Tables — “what are we measuring?”

  Contain the basic transaction-level information of the business that is of interest to a particular application. In marketing analysis, for example, this is the basic sales transaction data. Fact tables are large, often holding millions of rows, and mainly numerical.

- Dimension Tables — "by what are we measuring?"

  Contain descriptive information and are small in comparison to the fact tables. In a marketing analysis application, for example, typical dimension tables include time period, marketing region, product type...

2.2.7 Metadata information

Metadata structures the information in the data warehouse in categories, topics, groups, hierarchies and so on. It is used to provide information about the data within a data warehouse, as given in the following list and shown in Figure 14:

- "Subject oriented", based on abstractions of real-world entities like ('project', 'customer', organization',...)
- Defines the way in which the transformed data is to be interpreted, ('5/9/99' = 5th September 1999 or 9th May 1999 — British or US?)
- Gives information about related data in the Data Warehouse.
- Estimates response time by showing the number of records to be processed in a query.

- Holds calculated fields and pre-calculated formulas to avoid misinterpretation, and contains historical changes of a view .

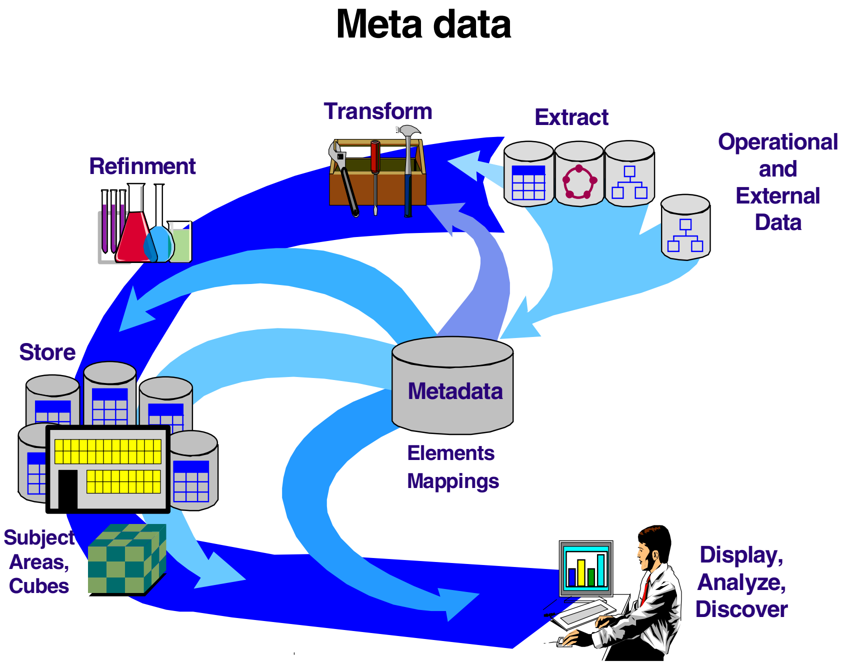
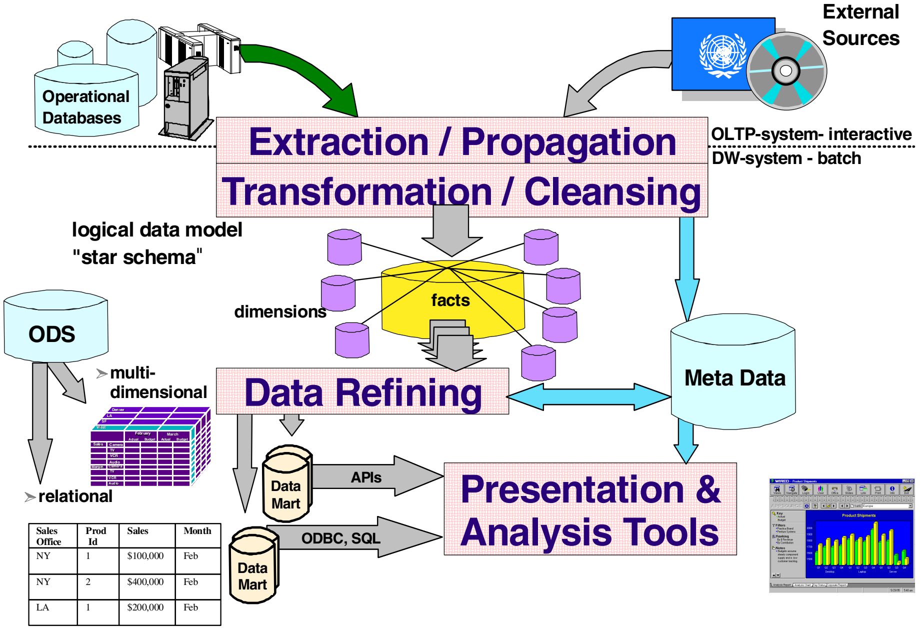


Figure 14. Metadata

The data warehouse administrator perspective of metadata is a full **repository** and documentation of all contents and all processes in the data warehouse, whereas, from an end user perspective, metadata is the **roadmap** through the information in the data warehouse.

2.2.8 ODS — operational data source

The operational data source (see Figure 15) can be defined as an updatable set of integrated data used for enterprise-wide tactical decision making. It contains live data, not snapshots, and has minimal history that is retained.



Sales Office	Prod Id	Sales	Month
NY	1	$100,000	Feb
NY	2	$400,000	Feb
LA	1	$200,000	Feb

Figure 15. ODS — Operational Data Store

Here are some features of an Operational Data Store (ODS):

An ODS is **subject oriented:** It is designed and organized around the major data subjects of a corporation, such as "customer" or "product." They are not organized around specific applications or functions, such as "order entry" or "accounts receivable”.

An ODS is **integrated:** It represents a collectively integrated image of subject-oriented data which is pulled in from potentially any operational system. If the "customer" subject is included, then all of the "customer" information in the enterprise is considered as part of the ODS.

An ODS is **current valued:** It reflects the "current" content of its legacy source systems. "Current" may be defined in different ways for different ODSs depending on the requirements of the implementation. An ODS should not contain multiple snapshots of whatever "current" is defined to be. That is, if "current" means one accounting period, then the ODS does not include more that one accounting period's data. The history is either archived or brought into the data warehouse for analysis.

An ODS is **volatile:** Since an ODS is current valued, it is subject to change on a frequency that supports the definition of "current." That is, it is updated to reflect the systems that feed it in the true OLTP sense. Therefore, identical queries made at different times will likely yield different results because the data has changed.

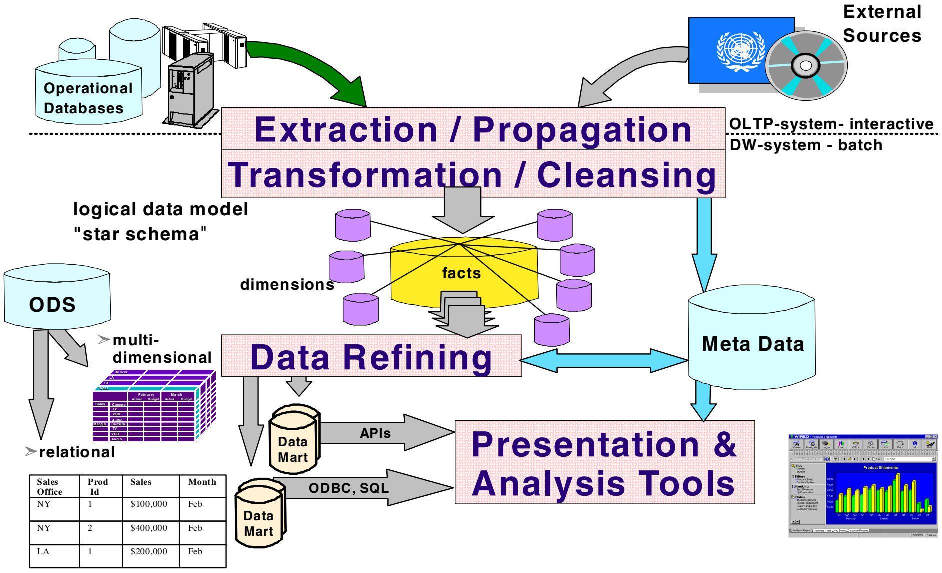
An ODS is **detailed:** The definition of "detailed" also depends on the business problem that is being solved by the ODS. The granularity of data in the ODS may or may not be the same as that of its source operational systems.

2.2.9 Data mart

Figure 16 shows where data marts are located logically within the BI architecture. The main purpose of a data mart can be defined as follows:

- Store pre-aggregated information.
- Control end user access to the information.
- Provide fast access to information for specific analytical needs or user group.
- Represents the end users view and data interface of the data warehouse.
- Creates the multidimensional/relational view of the data.
- Offers multiple "slice-and-dice" capabilities.

The database format can either be multidimensional or relational.



Sales Office	Prod Id	Sales	Month
NY	1	$100,000	Feb
NY	2	$400,000	Feb
LA	1	$200,000	Feb

Figure 16. Data mart

2.2.10 Presentation and analysis tools

From the end user’s perspective, the presentation layer is the most important component in the BI architecture shown in Figure 17 on page 32.

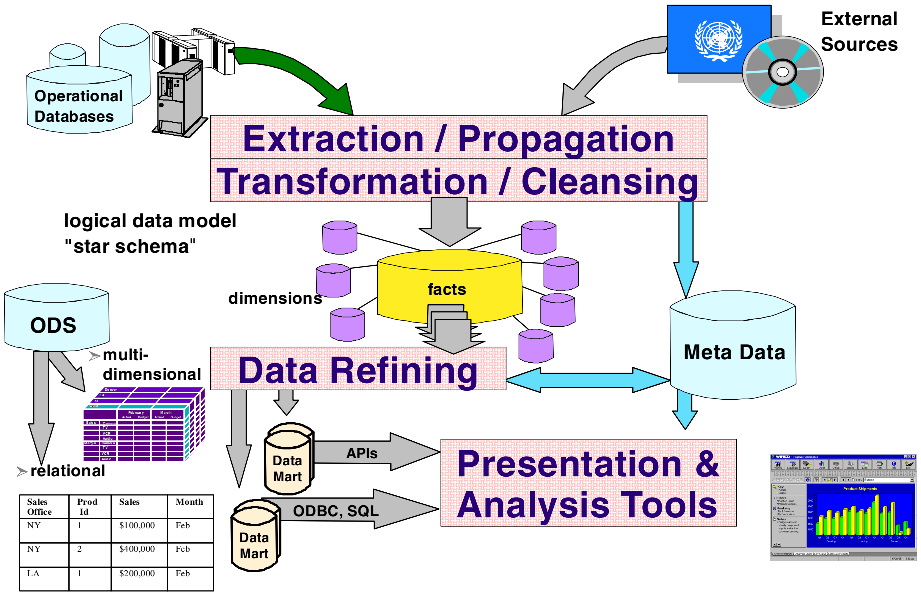
To find the adequate tools for the end users with information requirements, the assumption can be made that there are at least four user categories and the possibility of any combination of these categories.

- The "power user"

  Users that are willing and able to handle a more or less complex analysis tool to create their own reports and analysis. These users have an understanding of the data warehouse structure, interdependencies of the organization form of the data in the data warehouse.

- The "non-frequent user"

  This user group consists of people that are not interested in the details of the data warehouse but have a requirement to get access to the information from time to time. These users are usually involved in the day-to-day business and don't have the time or the requirement to work extensively with the information in the data warehouse. Their virtuosity in handling reporting and analysis tools is limited.



Sales Office	Prod Id	Sales	Month
NY	1	$100,000	Feb
NY	2	$400,000	Feb
LA	1	$200,000	Feb

Figure 17. Presentation and analysis tools

- Users requiring static information

  This user group has a specific interest in retrieving precisely defined numbers in a given time interval, such as:

  "I have to get this quality-summary report every Friday at 10:00 AM as preparation to our weekly meeting and for documentation purposes."

- Users requiring dynamic or ad hoc query and analysis capabilities

  Typically, this is a business analyst. All the information in the data warehouse might be of importance to those users, at some point in time. Their focus is related to availability, performance, and drill-down capabilities to slice and dice through the data from different perspectives at any time.

Different user-types need different front-end tools, but all can access the same data warehouse architecture. Also, the different skill levels require different visualization of the result, such as graphics for a high-level presentation or tables for further analysis. This aspect will be explained in detail later in this book.

Chapter 3. A Business Intelligence project

At first glance one might expect that Business Intelligence projects are very similar to any other IT project, with the typical phases of requirements analysis, design, development, test, rollout, production, and ongoing maintenance. Basically, this is true, because all of these phases are also found in the lifecycle of Business Intelligence projects. However, there are some characteristics that distinguish Business Intelligence projects from other IT projects.

First of all, it is very important to have the business departments involved in the project, because business analysts will directly access the data models, without an application layer that hides the complexity of the model (as is the case in traditional OLTP systems). To enable business analysts to navigate and manipulate the model, the structure of the data mart solution must be closely related to their perception of the business' objects and processes. This requires that groups of business specialists and IT specialists work together. Cultural issues between the business and IT departments may influence the project more than is usually the case in other IT projects.

The many different skills and resources required may be widely dispersed throughout the company. Some skills may not be available within the company or may be limited and have to be brought in from outside (consultants, and technical and tool specialists), because of the strong involvement of the business side of the house and the fact that a Business Intelligence project is, from a technical perspective, comparable to a systems integration project. Typically, more than one platform is involved, and many vendors and tools, multiple interfaces, integration with legacy systems, and client/server, and Web technologies have to be dealt with. The appropriate selection and coordination of the project team are key to the success of a Business Intelligence project.

The requirements for a Business Intelligence project are usually fuzzy and incomplete. The potential for additional requirements that occur way back in the development life cycle is very high, because users will recognize the capabilities of the technology when they are presented with and start working with the first preliminary models. That is why the development and delivery process for Business Intelligence solutions has to be iterative and designed for change.

Each individual business subject area should be targeted separately, to shorten the delivery cycle and provide business value to the company within a meaningful time frame. Plan for the delivery of business results as quick as possible (for example, using a rapid development approach in a pilot phase) and define the scope of the solution to fit into a time frame, not longer than six months. Starting with the pilot phase, work in short iteration cycles, to continuously enhance the solution and deliver business value to the users throughout the project, and align the solution as close as possible to the business. Then pick the next business subject area and, again, scope the project for not more than six months. Do not try to incorporate all aspects of the business into one model.

Business Intelligence projects tend to be cross-departmental. Therefore, even if only a specific business subject area is covered by the project, the business definitions and business rules must be standardized to be understood and valid on an enterprise level and standardized to ensure consistency and enable reuse. This characteristic could lead to lengthy discussions on how the business is looked at and interpreted among different business departments and could have an impact on the way the performance of the company is measured. The management of the Business Intelligence project must ensure that there is at least an official and agreed on definition for those measurements that are part of the deliverables (that is, data models, reports, and metadata catalog).

Business Intelligence solutions have to consolidate data from a lot of different sources from different lines of business throughout the company. The planning for the population subsystem that maps and transforms the data into the corporate-wide context that is needed in a Business Intelligence environment must consider data quality issues, which are usually discovered during this process. Resolving data quality issues and ensuring that only 100% correct, meaningful, and unambiguous data is delivered to the analysts can be a very complex and time-consuming process. However, it is of utmost importance to the success of the Business Intelligence project that the data in the analysis environment is correct, clean, validated and trusted by the business analysts. A major reason for the failure of Business Intelligence projects is the lack of trust in the analysis results due to data quality problems or ambiguous interpretations.

3.1 Who Is needed?

In this section we consider the roles and skill profiles needed for a successful Business Intelligence project. We describe the roles of the business and development project groups only. Not all of the project members that we describe are full-time members. Some of them, typically the Business Project Leader and the Business Subject Area Specialist, are part-time members. The number of people needed to accomplish a task depends on the organization and scope of the Business Intelligence project. There is no one-to-one relationship between the role description and the project members. Some project roles can be filled by one person, whereas, others need to be filled by more than one person.

3.1.1 Business Project Group

The Business Project Group is mainly concerned with the business value of the solution. The members of this group drive the project, because they are the ultimate consumers of the information delivered by the new solution. The business project group defines the requirements and the scope of the project. It is responsible for the alignment of the solution to the business goals of the company.

3.1.1.1 Sponsor

In general a Sponsor is needed for all types of projects. But in a Business Intelligence project we particularly need a Sponsor from a business department (for example, the Chief Financial Officer). The Sponsor plays a very important role and must have the trust of executive management. He or she has the business need for the new solution and the financial responsibility for the project. The Sponsor is also involved in making the key scoping decisions and supporting them throughout the project. He or she has to uphold the vision related to the new solution and reinforce and encourage the user community within the company. It is extremely important that the project team has a direct communication path to the Sponsor.

In large Business Intelligence projects there is also a need for an IT Sponsor, who is responsible for those parts of the project budget that are outside the scope of the Sponsor from the business department (especially for hardware and software installation, connectivity, and operations).

The Sponsor usually nominates a Business Project Leader who represents the business community and works closely with the Technical Project Manager.

3.1.1.2 Business Project Leader

The Business Project Leader should be a person from the line of business organization. He or she will also use the new solution and should be empowered and able to make detailed decisions from a business perspective during the project. The Business Project Leader should have a solid understanding of the business requirements. He or she works closely with the Technical Project Manager.

3.1.1.3 End user

End user representatives with business responsibility will work with the Business Intelligence solution and should, therefore, be part of the project as well. It is important to find end users who are open to new technologies. They should be able to share information about their detailed business processes and needs.

3.1.2 Development Project Group

The Development Project Group deals with the delivery of the Business Intelligence solution. This group works closely with the Business Project Group to map the business requirements to a technically feasible and manageable solution.

3.1.2.1 Technical Project Manager

The Technical Project Manager should have experience with Business Intelligence or Decision Support projects. He or she should be able to staff the project with qualified project members and build a team that can work together. This is critical to the success or failure of the project, because a Business Intelligence project needs a lot of different skills and a lot of different people who speak different business languages.

The Technical Project Manager is responsible for such tasks as coordinating resources, managing the project activities, tracking the project status, and setting up a communication structure for the project.

The Technical Project Manager should have strong communication skills and a technical background. He or she should know which processes are necessary in an end-to-end solution. The Technical Project Manager must be able to establish the link between the technical and the business part of the project and navigate through the political environment of the organization.

3.1.2.2 Business Intelligence Solution Architect

The Business Intelligence Solution Architect is in charge of the technical solution. He or she is knowledgeable about the architectures and products available to design the solution. He or she has to ensure that the different platforms, tools, and products can be integrated in an end-to-end solution that fits the requirements, is manageable, and can grow with increasing business demands. The Business Intelligence Solution Architect is involved in the design of all major components of the solution (that is, the data staging and population subsystem, databases and connectivity, information catalog, warehouse management subsystem, analysis applications and tools, and archiving solution). He or she drives the work of the various Platform and Tool Specialists.

3.1.2.3 Business Subject Area Specialist

The Business Subject Area Specialist should have knowledge of the business processes, applications, and data related to the specific business problem that the solution addresses. He or she also should know who is responsible for the definition of a key business measure, or who can decide which definition is correct. Because the Business Intelligence Solution Architect is also responsible for the quality of the information, he or she is heavily involved in validating the information provided by the solution. The Business Subject Area Specialist has the trust of the Business Project Leader and the Sponsor and his or her opinions are very important to them. This role is usually a very critical resource in the project, because it has to be filled from among the few key business analysts in the company, who cannot withdraw completely from their day-to-day business duties for the duration of the project.

3.1.2.4 Database Administrator

In cooperation with the Business Subject Area Specialist, the Database Administrator knows where to find and how to interpret the source data. He or she knows the structure of the data and the data relationships. The Database Administrator provides access to the source and target data. He or she is usually also responsible for security. The Database Administrator is the only person who can handle security for the Business Intelligence environment from a single point of control.

The Database Administrator should also be involved in validating the data model for the new Business Intelligence solution.

3.1.2.5 Platform Specialists

Usually more than one Platform Specialist is needed in a Business Intelligence project. For each legacy system (for example, OS/390 hosts, AS/400, and/or UNIX systems) that acts as a source for the Business Intelligence solution, a Specialist will be needed to provide access and connectivity. If the Business Intelligence environment will be multi tiered (for example, UNIX massive parallel processing (MPP) platforms or symmetrical multiprocessing (SMP) servers, and Windows NT departmental systems) Platform Specialists are needed as well.

Usually, people from the IT department of the company are involved, as far as the legacy environment is concerned. Due to the day-to-day operational duties of these people, the Technical Project Manager has to make sure to plan for and inform them as early as possible.

3.1.2.6 Tool Specialists

Many different tools are usually needed to build a Business Intelligence solution, from extraction, transformation, and cleansing tools to data access middleware, and from the population subsystem to data warehouse management and analysis tools for standard query and reporting, OLAP analysis, or data mining. The Tool Specialists must know how to install, implement, and tune these tools.

Very often, the Tools Specialists are provided by the vendors of the tools. Services packages offered by the vendors could also include an education package to transfer the necessary skills to the future administrators and/or users of the tools.

3.1.2.7 Extract Programmers

It is often necessary to plan for an additional Extract Programmer, even if extraction, transformation, and replication tools are going to be used, because the tool may not support a data source, or it may not be capable of the complex transformations needed to extract certain business rules hidden in some of the data items. Perhaps a temporary (prototype) solution is needed to allow the validation and quality assessment of the extracted source information by end users in the context of the solution. But be careful with this approach! You could end up with a maintenance nightmare, if the programs are not properly designed and documented; for example, they are managed like traditional application development projects.

3.2 The development process

Basically, a Business Intelligence project has to deal with three major topics:

- Infrastructure
- Data
- Application
- **Infrastructure** includes all the tasks necessary to provide the technical basis for the Business Intelligence environment. This includes the installation and implementation of new hardware and software, the connectivity between the legacy environment and the new Business Intelligence environment on a network, as well as on a database level, and the implementation of a population subsystem, an administration subsystem, and a management subsystem. Establishing the infrastructure for the first Business Intelligence solution is time consuming, but with the selection of scalable hardware and software components, the effort will decrease dramatically for the next project or delivery cycle.

- **Data** deals with data access, mapping, derivation, transformation, and aggregation according to the requirements and business rules, as well as with the proper definition of the data items in business terms (metadata). It also contains the tasks necessary to ensure the consistency and quality of the information being transferred to the Business Intelligence environment. The effort for the tasks involved in the data topic should decrease with each new Business Intelligence project, depending on the amount of data that can be reused from previous projects (or iterations).
- **Application** includes the gathering of the business requirements, the design of the model, and the implementation, visualization, and publication of the analysis results in terms of, for example, queries, reports, and charts. The effort needed for the tasks within the application topic is heavily dependent on the selected scope of the project.

The scope of a Business Intelligence project should be selected in such a way that a complete solution (that is, infrastructure, data, and application) for the business analysis domain selected can be offered and valuable results can be delivered to the business analysts within a reasonable timeframe (no longer than six months).

The Business Intelligence solution is then enhanced in an evolutionary and iterative way, as shown in Figure 18 on page 40.

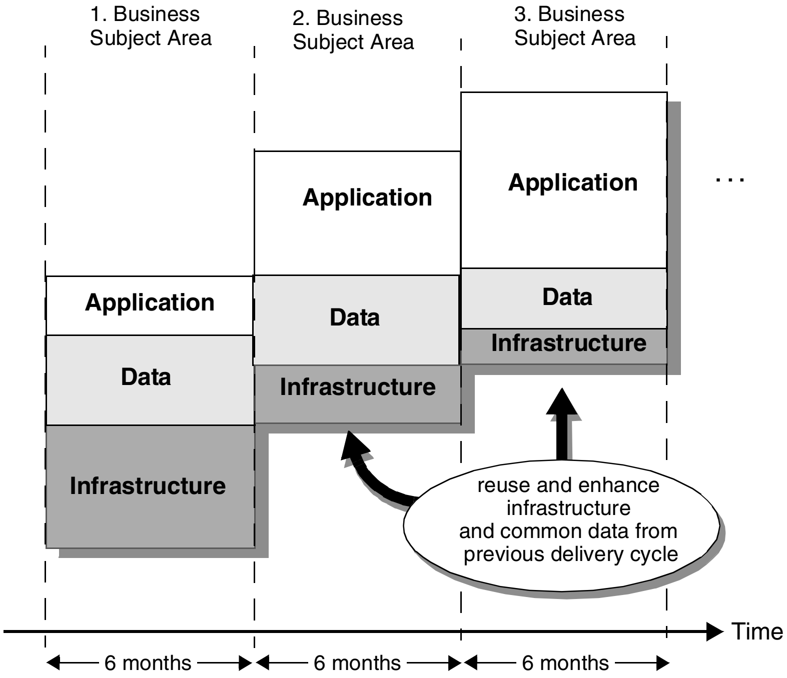


Figure 18. Iterative Data Mart development approach

As you can see in Figure 18, each consecutive delivery cycle leaves more room for application-related efforts by reusing as much of the infrastructure and data of the previous cycles as possible.

3.3 Planning a project

The following are essential questions that require good answers **before** the BI project can even begin.

1. *What are the specific strategic business objectives (drivers) that the solution is supposed to achieve?*

   The overriding reason many decision support projects fail is not that the projects were technically unfeasible. On the contrary, many of the technological challenges of data warehousing have proven answers. The most common cause for failure is that the warehouses did not meet the business objectives of the organization. Warehouses that do not satisfy the business user's needs are not accessed and eventually die.

2. *What are the specific, calculable measurements that will be used to evaluate the return on investment (ROI) of the BI system in meeting the company's business objectives?*

   Clear business objectives are measurable. This activity is critical since once the BI project is completed the management team will have to justify the expenditures. Moreover, it is important to understand that a data warehouse is **not** a project, it is a process. Data warehouses are organic in nature. They grow very fast and in directions you would have never anticipated. Most warehouses double in size and in the number of users in their first year of production. Once a cost justification can be quantified for the initial release, and the process for gaining funding for the follow-up releases is greatly simplified.

3. *Are the key users of the data warehouse identified and committed to the success of the project?*

   The users **always** dictate the success or failure of the warehouse. The users need to be heavily involved throughout the data warehousing project. To take it a step further, the users need to have a personal stake in the success of the project. It is amazing how quickly problems vanish when everyone has a vested interest in the project. Also, make it a point to educate them on the process of data warehousing. Teach them its benefits along with its limitations. This will aid in managing their expectations. A good rule of thumb is if you have gone more than two weeks without talking to your users, then it is time to set up a meeting. Keep in mind many times these people are the ones picking up the tab on these project.

4. *Is the organization trying to build a 500+ gig, "do-all, be-all" data warehouse on their first iteration?*

   Data warehouse projects stretch an organization in ways unlike that of OLTP projects. From a political perspective, an enterprise data warehouse requires consent and commitment from all of the key departments within a corporation. In addition, the learning curve of data warehouse project team is seldom understood. There will be a new and dizzying array of software tools (transformation, access, metadata, data cleansing and data mining) that will require tool-specific training. By adding massive amounts of data into the equation, the points of failure increase significantly. Moreover, large volumes of data will push the envelope of the RDBMS, middleware and hardware, and could force developers into using parallel development techniques if an MPP (massively parallel processing) architecture is needed. Keep in mind that the answer to many of these challenges comes in the form of a hefty price tag. As a result, adding the dimension of size is just too painful and costly for most enterprises to attempt during the first iteration. Bill Inmon said it best when he stated that, "Warehouses are best built in an iterative fashion." Do not misunderstand, this is not to recommend that a company should not build a fully functional, multiple terabyte, seven subject area, Web-enabled, end-to-end, enterprise data warehouse with a compete metadata interface. It simply means that the highest probability for success comes from implementing decision support systems (DSSs) in a phased approach. By using the first iteration as an opportunity to train the corporation, it will set the stage for bigger and better future implementations.

5. *Does the data warehouse project have support from executive management?*

   Any large-scale project, whether it is a data warehouse system, or if you are implementing that hot new vendor order management system, needs executive management on board. Moreover, their involvement is imperative in breaking down the barriers and the "ivory towers" in all of our companies. Their position allows them the ability to rally the various departments within a corporation behind the project. Any substantial project lacking executive management participation has a high probability of failure.

6. *Does the organization have a clear understanding of the concepts and tools involved in data warehousing?*

   If you do not have a data warehouse built, then the answer to this question will most likely be no. As a result, training and education will be required. Keep in mind that training is required at many levels. First, initial education is necessary to convey the concepts of what is a data warehouse, data mart, operational data store, star schema design and metadata. Second, data acquisition developers will probably need to be trained on a transformation tool. Third, data warehouse access developers will require significant training in an OLAP tool. Fourth, data administration developers will need training on a tool that will integrate all of the company's metadata into one repository. Fifth, more than likely there will be a Web component used to access the data warehouse and the metadata repository. Depending on your organization, additional training and outside consulting could be needed for each of these areas. Keep in mind that these are only the data warehousing specific training issues. There still needs to be an understanding of the hardware, middleware, desktop, RDBMS and coding language (COBOL, C++, and so on) of the transformation tool.

7. *Are there a highly experienced project manager and data warehouse architect that have experience building warehouses that will actively participate throughout the project?*

Data warehousing projects are fundamentally different from OLTP projects. OLTP projects are necessary in order to operate the day-to-day business of the company. Data warehouse projects are critical for making strategic decisions about an organization. In addition, data warehouses grow at an alarming rate during the first few years of production. An experienced data warehouse project leader understands these facts and keeps the vision of the project in concert with the real-world reality of decision support. In addition, the data warehouse architect must design a scalable, robust and maintainable architecture that can accommodate the expanding and changing data warehouse requirements. These fundamental challenges require highly experienced, senior level individuals. These positions can be filled via in-house resources or by consultants. If consultants are used to fill these roles, it is imperative that the consultants are highly skilled at knowledge transfer and that in-house employees have been assigned to shadow the consultants for both of these roles.

8. *Has an experienced consultant been brought in to do a readiness assessment of the organization?*

   This step is very important since an experienced hand can identify problem areas in the organization that can be dealt with early in the data warehouse project's life cycle. Identifying that person is another issue. Be wary of consultants without real-world, hands-on experience. It is one thing to be able to write or speak about data warehousing; it is entirely something else to have the experience needed to navigate through the political quagmires and to know what it takes to physically build a data warehouse. If you answered "No" or "I'm not sure" to any of these questions, then you will need to discover the answers before you head down the data warehousing trail. Without those answers, you might need to pack up your snowshoes and thermal underwear because your supervisor will be ordering you that one-way ticket on Siberian Airlines.

3.4 Success factors for a BI solution

In this section we summarize the success factors that we consider essential for Business Intelligence projects, in addition to the technical issues and challenges. These main success factors are:

- Scope the project to be able to deliver within at least six months.
- Select a specific business subject area; do not try to solve all business requirements within one project.
- Find a sponsor from the upper management of the business side of the company.

- Involve the sponsor throughout the project.
- Establish a sound information and communication structure that includes business and technical staff inside and outside the project.
- Define the contents and type of the deliverables of the project as early and in as much detail as possible.
- Together with the end users validate the results of the analysis phase (the initial dimensional models) against the deliverables definition.
- Deploy the solution quickly to a limited audience and iterate development.
- Establish commonly agreed on business definitions for all items within the scope of the project.
- Validate the quality and correctness of the information before making it available to the end user community.
- Keep the end users involved and informed throughout the project.
- Be prepared for political and cultural obstacles between business departments or between business and IT departments.

There are a number of examples of success indicators. Let us take a look, below, at some measures of success.

1. **Return on investment (ROI)**. A ROI can be achieved in a number of ways, such as:

   - Lower cost – Costs could be lowered through better inventory management, fewer dollars spent on unproductive measures, product promotions, and so on.
   - Improved productivity – Greater productivity could be expected from both IT and the user. Today user analysts may spend 80 percent of their time gathering data and only 20 percent analyzing the data. The data warehouse should reverse those numbers. IT will still be responsible for developing complex reports, as well as writing reports for production systems. The data warehouse can provide reporting tools with a well documented, clean and easily accessible database. This capability should significantly improve IT productivity.
   - Increased revenue – This could be a result of greater market share and increased sales as marketing is able to more effectively target customers and provide the right products at the right time to the right market. The effects of costs and revenues may be difficult to assign to the impact of the data warehouse. As the data warehouse is being implemented, the organization is not standing still. There are both internal and external

factors that impact costs and revenues, so the actual benefits of the data warehouse may be difficult to determine.

2. The data warehouse is used.

   One of the easiest categories to understand can be measured by the number of users and the total number of queries and reports generated. If queries and reports are run regularly, it is a good indication that the users are achieving some benefit.

3. The data warehouse is useful.

   The data warehouse may be used, but the users may find the benefits to be marginal and illusive. It is important to ask the users what they see as the benefits of the data warehouse, how it has changed the way they do business, how it may have improved their productivity and how it may have improved the quality of their decisions.

4. The project is delivered on time.

   This measure is problematic, as schedules are often set without an understanding of what is involved and how long each project task will take. "On time" is only relevant if a realistic schedule is the base for comparison.

5. The project is delivered within budget.

   This criterion may be difficult to achieve since the total costs of a data warehouse are difficult to determine. Initially, you may not have known how many users to expect, how many queries and reports they would be generating and the complexity and resources used by the queries and reports. You did not know how large the data warehouse would be or how many indexes and summary tables would be required and desired. You may not have anticipated needing a larger CPU. You may not have known that the software was more difficult than the vendors represented, resulting in teams of software consultants being required. You may not have anticipated needing to upgrade your network to support the increased line traffic. You may not have anticipated needing to raise the salaries of the data warehouse team, nor the increased cost of recruiting the talent required to make the project a success. All these factors will contribute to severely underestimating the budget. "Within budget" is only relevant if a realistic budget is the basis for comparison.

6. There is improved user satisfaction.

   Users may be internal, external, or both. In all cases, the goal is to have users who are happy with the features and capabilities, performance, quality of the data, and level of support.

7. There are additional requests for data warehouse functions and data.

You will know you were successful if other user departments are beating down your door with requests for access to the data warehouse, and current users are requesting new data and functions to be added to the existing data warehouse.

8. Business performance-based benchmarks.

   This is the most subjective of all the measures and will become the most controversial. Most industries have sets of industry averages, as well as to benchmark (the best) companies against which they make comparisons. For example, the industry average for quality, represented by the number of defects for a new car, may be three, while the best is one. With better information, a car manufacturer in the middle of the pack may have a goal to manufacture a mid-size sedan using eight worker days. The data warehouse may be able to provide improved, more complete and timelier information, and, with this information, the auto manufacturer may be able to achieve their productivity goals.

9. Goals and objectives are met.

   On the assumption that you have developed goals and objectives, success will be defined by how well these goals and objectives were met. No doubt, not all were met or were only partially met. A scorecard will give you an initial – and then an ongoing – measure of your project's success.

10. Business problems are solved.

    The data warehouse was developed for some specific reason. Perhaps marketing was unable to identify customer demographics for target marketing. If the data warehouse now provides this capability, it should be considered a success.

11. Business opportunity is realized.

    The identified opportunity might have been the ability to provide information to suppliers through the Web, so they would be able to respond more quickly to your demands for components that you need for your manufacturing process. If the supplier now has fewer stock-outs, the project is successful.

12. The data warehouse has become an agent of change.

    The world is changing, and the rate of change is accelerating dramatically. Successful organizations must be able to respond and respond quickly. Decisions must be made more quickly, but this can only happen with better and more timely information. There can be some fundamental changes to the business in the manner and speed in which decisions are made and the data warehouse can be the vehicle for that change.

3.5 Measures of failure

We may not be sure if the data warehouse is a success, but we will always know when we have failed. Some of the indications of failure are:

1. Funding has dried up.

   This could be because the sponsor has moved on or is no longer interested in the project. There could be other factors that have nothing to do with the success of the project, such as the company is being bought by barbarians who have no appreciation of your efforts or the data warehouse possibilities. Real failure results if the project is perceived as having no real benefit to the organization.

2. Users are unhappy with the quality of the data.

   If you took a shortcut and decided not to clean the data, users will reject the data warehouse. The users may even have told you not to spend the time to understand and clean the data – they want the data warehouse as soon as possible and do not care if the data is dirty. Do not believe them. Even if the initial users do not care if the data is dirty, other users in the organization who do not know how to interpret the dirty data do care. These new users would have to learn how to decipher the dirty data, or they would get the wrong results if they do not realize it is dirty. If the quality of the data is poor, the data warehouse will be a failure.

3. Users are unhappy with the query tools.

   The notion of "one size fits all" is inappropriate for selecting tools. Power users need and want different tools other than those fitted to the inexperienced and technologically intimidated users. The needs of each user segment must be met with the appropriate tool and the appropriate training and environment. Users will let you know when they do not like the query tool that you have foisted on them.

4. Only a small percentage of users take advantage of the data warehouse.

   You may have expected everyone in marketing to be actively writing queries. They all went through training and graduated with a query tool certificate. However, it turns out that whenever anyone needs some information, they all go to only one person in the department who is a frequent and knowledgeable user.

5. Poor performance is a result.

   Response time is so bad that users only launch queries right before leaving on extended vacations. It is not only that response time is bad, but there does not seem to be any means of improving performance that anyone is aware of. There are four indicators of performance to be considered:

- Query response time
- Report response time
- Time to load/update/refresh the data warehouse
- Machine resource

6. The data warehouse has the inability to expand (it is not scalable).

   You expect the data warehouse to grow, but if the technical architecture does not allow for the expansion of the number of users and the size of the database, the infrastructure must then be recast at enormous cost and lost opportunity.

7. Data is not integrated.

   One of the goals for the data warehouse is the ability to integrate data coming from many and heterogeneous source files and databases. You wanted departments to be able to share data and to have data of record for the organization; data that everyone in the organization would accept as correct. A proliferation of data marts that do not have common keys or common data definitions is an indication of islands of data marts that cannot be shared.

8. Extract, transform and load steps do not fit in the batch window.

   Extraction, transformation and load can consume up machine resources and time. A number of data warehouses have failed because of the very lengthy time required for these processes. This problem is especially critical to global companies who have to be up and running in several time zones and are, therefore, starting with a reduced batch window.

3.5.1 Other critical success factors

If a factor or characteristic is critical to the success of a project, we shall call it a critical success factor (CSF). The absence of that factor or characteristic dooms the project. CSFs provide a measure for the completion and quality of the project. By knowing and understanding what is very important, the project manager can make a case for adequate budget, resources, schedule improvement, and management commitment.

Examples of critical success factors

These critical success factors are mandatory for a successful data warehouse project:

1. Common data definitions are used.

The definitions in most organizations make the Tower of Babel look communicative. Every department has its own set of definitions for business terms which are often defined differently by other departments. To make matters worse, these departmental definitions are rarely documented. Department heads assume that everyone shares their understanding of the business and their definition of the major business terms. Not wanting to appear stupid, most employees do not question the meaning of business terms. While it is not possible to gain definitional concurrence among departments, each project must have a glossary of business terms that support the project.

2. Well-defined transformation rules exist.

   As the data is brought over from the source systems to the data warehouse, much of it will be transformed in one way or another. The data may be specifically selected, re-coded, summarized, integrated with other data or changed in some other way. The rules for the transformations are critical to the users getting what they expect and need.

3. Users are properly trained.

   In spite of what the vendors tell you, users must be trained, and the training should be geared to the level of user and the way they plan to use the data warehouse. In addition to the tool, users should learn about the availability of predefined queries and reports. Users must learn about the data and the power users should have more in-depth training on the data structures. Just-in-time training will solidify and reinforce the skills learned in class as the students immediately begin using the data warehouse at the conclusion of the class.

4. Expectations are communicated to the users.

   IT is often unwilling or afraid to tell the users what they will be getting and when.

   - Performance expectations: Users must know that not all of their queries will have sub-second response time. A query joining two tables of 10 million rows each will take minutes or even hours, and the users should expect such.

   - Expectations of availability include the time and days the system is scheduled to be accessible (that is, 6 a.m. to 11 p.m., Monday through Saturday) as well as the percentage of time planned for availability (that is, 97 percent availability during scheduled hours). A service level agreement (SLA) will normally document an availability agreement.

   - Function includes what data will be accessible, what predefined queries and reports are available, the level of detail data, and how the data is integrated and aggregated.
   - The expectation of simplicity is the ease of use. Users do not want a complex system.
   - The expectations of accuracy are for both the cleanliness of the data as well as an understanding of what the data means.
   - Timeliness is when the data will be available (that is, three days after month end) as well as the frequency of refreshing the data (such as daily, weekly, monthly).
   - Schedule expectations when the system is due for delivery. Since not all the users will be getting access on the first implementation, each user needs to know when they will get their turn.
   - The expectation for support comes into play as the users have problems. Where will they go for help? How knowledgeable will the support be for the query tools? How well will they understand the data?

   All these expectations should be documented in a scope agreement. In addition to the scope agreement document, every opportunity must be seized to clarify expectations for the users, especially when a casual comment may create a misunderstanding of what they will be getting.

5. User involvement is ensured.

   There are three levels of user involvement:

   - Build it; they will use it.
   - Solicit requirements input from the users.
   - Have the users involved throughout the project.

   To have the users involved throughout the project is by far the most successful approach. A commitment by the users to the project is critical to the project's success, and the users involvement is an excellent indicator of that commitment.

6. The project has a good sponsor.

   The best sponsor is from the business side, not from IT. The sponsor should be well connected, willing to provide an ample budget, and able to get other resources needed for the project. The sponsor should be accepting of problems as they occur and not use those problems as an excuse to either kill the project or withdraw support. Most importantly, the sponsor should be in serious need of the data warehouse capabilities to solve a specific problem or gain some advantage for his or her department.

7. The team has the right skill set.

   Without the right skills dedicated to the team, the project will fail. The emphasis is on "dedicated to the team." It does little good to have skills somewhere in the organization if they are unavailable to the project. The critical roles should be reporting directly to the project manager. Matrix management does not allow the project manager to control these resources. Without this control, there are no guarantees that the people will be available when needed.

8. The schedule is realistic.

   The most common cause of failure is an unrealistic schedule, which is usually imposed without the input or the concurrence of the project manager or the team members. Most often, the imposed schedules have no rationale for specific dates but are only means to "hold the project manager to a schedule." Those imposing the schedule usually have little concept of the tasks and effort required.

9. The project has proper control procedures (change control).

   There will always be changes in the scope, but the scope must be controlled, and change control must be implemented just as it is in transactional systems.

10. The right tools have been chosen.

    The first decisions to be made are the categories of tools (extract, transform and load, data cleansing, OLAP, ROLAP, data modeling, administration, and so on). Many of these tools are expensive, not just for their initial costs and maintenance costs, but in training, consulting, and in terms of the internal people required to implement and support the tool. The tools must match the requirements of the organization, the users, and the project. The tools must work together without the need to build interfaces or write special code.

You are going to have to determine the critical success factors for your organization and your project. With these in mind, and after they have been documented, you will be able to compete for the scarce resources that you will need for your project (good people, budget, time, and so on).

3.6 Process for a successful data warehouse

To achieve a successful implementation of a BI project related to a data warehouse implementation, there is a 6-step checklist you may follow. These six steps are:

- Establish the project
- Prepare the project
- Initiate the database
- Explore the database
- Implement
- Iterate/Expand

3.6.1 Establish the project

To establish the project the following tasks have to be performed:

1. **Gain corporate commitment and sponsorship.**

   The very first step to be taken for a successful implementation of the project is to make sure the project is well understood in the long term vision. Once the supporting parties (sponsors) know that the overall project may not solve their needs within days, they have to be assured that the project will deliver fast results in certain areas and of the overall positive effect of the project.

   A BI project requires cross-functional cooperation and therefore representatives of all business areas should be involved in the project planning. Any problems related to the overall costs of the project need to be addressed from the beginning, as a BI project may not be feasible to run on an operational system and it also is very likely to grow over time. Within this team the project priorities should be defined to avoid loosing the support of some sponsors.

2. **Define high-level architecture.**

   After having assured the sponsorship, the next step is to design and build the planned BI architecture. It has to be clear whether the overall goal is to build a enterprise-wide data warehouse or to build a miniature and add on over time. Long term result may be similar, but if you know where you intend to go, you may avoid some "do-overs".

3. **Target the opportunities.**

   Make sure everybody understands the business problem that is intended to be solved with the BI project. Attributes of a good initial project are:

   - The source data exists or can be easily acquired
   - The end users are willing to commit their time
   - Results will have immediate value

   In addition, the scope of the project has be taken into consideration. Attributes of the scope to consider are:

   - Target a single subject area
   - From 2 to 6 sources at most
   - From 2 to 4 consumers
   - Common usage across consumers
   - Can be achieved in 3-6 months
   - Incremental investment

4. **Establish realistic goals.**

   These goals for the overall project should be specific, achievable, and measurable. Sample goals for the project might be:

   For the I/T department:

   - Build client/server solution skills
   - Incorporate 8 to 10 data warehouse tables

   For the end user:

   - Reproduce x, y and z reports from data warehouse
   - Have at least hard copy metadata
   - Be able to track sales trends for last 12 months by product

5. **Develop high level project plan.**

   The checklist for the high level project plan may look as follows:

   - Incorporate iterative principles
     - Get minimal data all the way through the process
     - Debug the process
     - Add more data/subject areas
     - Incorporate discoveries into next cycle
   - Establish time limits (deadlines)

3.6.2 Prepare the project

After the project has been defined precisely, the tasks involved in the preparation of the project are:

1. **Define the tasks to be done (major categories).**

   The first step within the project preparation should address all questions about what needs to be done and who does what within the project. typical tasks are:

   - Data acquisition
   - Data Modeling
   - Operations
   - Metadata
   - Tools selection
   - Support

2. **Gather high-level requirements.**

   Next, the high level requirements have to be defined. These can be separated into business and technical requirements.

   Business requirements

   - Process(es) involved
   - Critical success factors
   - Business entities, attributes, and relationships (hierarchical, horizontal, and so on)
   - Business measurements
   - Types of users (executive, novice/casual, analyst, power, developer)
   - Budget for project

   Technical requirements

   - Physical topology, such as hardware/software) and network configuration
   - Logical topology
   - Source database
   - Warehouse database issues such as data needed, structure, transportation, transformation, cleansing, propagation requirements like size and growth, and operations (security, availability, automation)

3. **Assemble project team.**

Once the project is defined and the plan has been agreed on, the next necessary step is to get the project team defined. The following groups and responsibilities may be involved:

- **Technical staff** (skills needed) to act as project leaders, design and implement the warehouse database, design and implement the data marts and metadata, and to assure database and SQL performance. The people in this group are typically database programmers, database administrators, and database analysts.
- **Business professional staff**, such as subject matter expert(s) and business analysts; they are the representatives to give input on the required information and typical queries.
- **Corporate sponsors** will be responsible for the executive steering presence. Representatives of the sponsoring areas should be recruited from both end user and I/T executive sponsors.
- **End user staff** is required to provide input on topics to be used, such as identify end user tool classes, data access/query, report writers, multi-dimensional database (MDD) management systems, advanced decision support, and Executive Information Systems (EIS).

3.6.3 Initiate the database

The next step in the overall project will be to initiate the database. This step again can be split up into several small processes that are described in the following.

1. **Gather detailed user requirements.**

   The user requirements may be different depending on the user type, such as executive, casual/novice, business analyst, developer, and so on. All these different user types may have different skills, require different tools, and have certain requests to the overall solution that will influence the database design. They also differ in the expectations they have to the project, such as performance, frequency of access, amount of data per access, availability, access to metadata, drill down / roll up perceptions, and so on.

2. **Identify transformation and derivation attributes.**

   Within this step it may be a good idea to extract (create the snapshot) the operational data into a separate system in order to perform some detailed investigation of the quality of the source data. The data should be analyzed to get an idea of the effort to be taken when performing the:

   - Cleansing to check the data for validity, consistency, accuracy, correctness, and trustworthiness.

- Mapping/translation (codes to characters, and so on) when consolidating different data sources.
- Calculations necessary to aggregate data.
- Summarizations required.

3. **Model facts and dimensions.**

   As the next step in order to initialize the database, the facts and dimensions need to be defined. The team members involved in this step need very good knowledge of the available data, the business itself, and the required analyzes to be performed. Figure 19 shows a sample for the definition of a fact table and the related dimension tables.

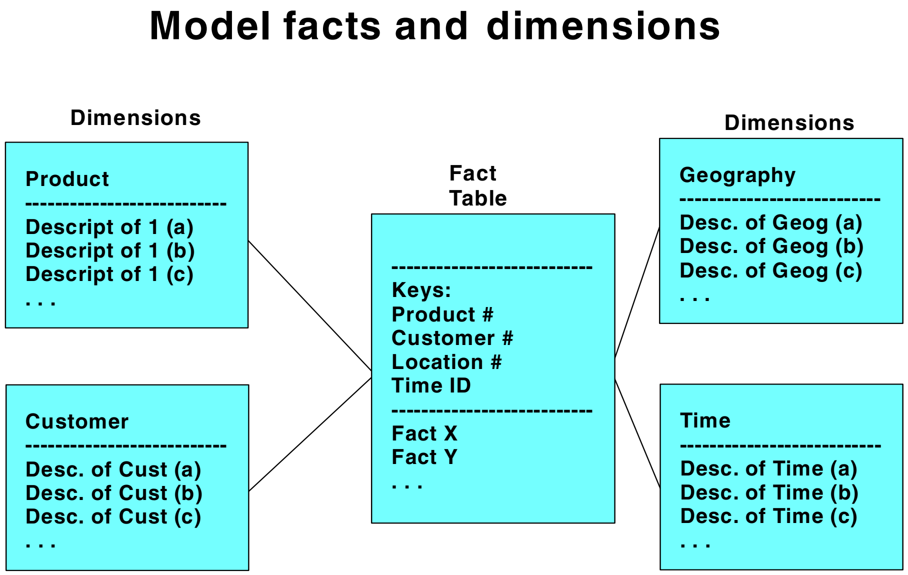


Figure 19. Facts and dimensions

4. **Architect the database (including metadata).**

   The last step related to the database design is to make a decision of the data warehouse architecture. These considerations cover topics, such as the definition whether the data warehouse is supposed to be implemented using a 1- 2- or 3-tier hierarchy and the decision about the database model to be either relational (ROLAP) or multi-dimensional (MDD).

   This step ends with a mapping of the subject area model into reality.

5. **Design the infrastructure.**

After the database design has been finished, the infrastructure needs to be considered as well. Topics to be investigated within this process are:

- Hardware/software configuration required for the implementation
- Propagation methods required to fill the architectural needs
- Frequency of propagation
- Availability of the overall system

6. **Acquire source data.**

   The database is defined, the infrastructure is configured and implemented, now it is time to determine some initial sample source data for testing. This process should also involve the end users in a sample selection, as they will be able to identify the data necessary for this step.

7. **Populate the data warehouse database.**

   After the test has been successful, the base dimension tables need to be populated. To do this, follow these steps:

   - Manually extract and load fact table
   - Implement only necessary transformations. Do not get hung up on one phase of the process, and do not try to identify big glitches in all phases.
   - Iterate on the loading process until the output is clean enough for the end user to access.

3.6.4 Explore the database

After loading the sample data into the data warehouse, the database should be explored during its testing phase to verify the design, usage, and so on. This will help to:

- Identify propagation and preparation dependencies.
- Monitor end user usage, such as access patterns and performance.
- Tune database and tools for optimal performance, such as iteration on table design as needed (summarize, aggregate, and so on).
- Plan and schedule update process/cycle.
- Define monitoring and control procedures.
- Define backup and recovery methods.
- Design archiving and retrieval plans/techniques.
- Create rollout plan for full implementation.

3.6.5 Implement the solution

The final implementation step of the overall solution itself consists of the following steps:

1. **Prepare the production environment.**

   The sequence to activate the environment should be to install and test the solution in the following order:

   a. Acquisition/propagation process

   b. Warehouse update/population

   c. Query environment and end user tool installation

2. **Train the users.**

   Operators and end users should then be trained to understand the overall solution. Topics, such as escalation procedures, recovery procedures, operational and data dependencies are relevant for the operators, whereas, the end user should mainly understand topics, such as the tools to use and the escalation process to solve problems.

3. **Define/initiate the support process.**

   The following questions should be documented thoroughly to achieve an optimum support structure:

   - How do you call?
   - When do you call?
   - Whom do you call?
   - What can you expect?
   - How do you escalate?
   - How is satisfaction measured?

4. **Move into production.**

3.6.6 Iterate to expand the warehouse

Once the overall solution is implemented and is being used, the new solution should be investigated to evaluate a future expansion of the existing solution, or to evaluate new projects. Actions that should be taken to do this are, for example:

- Continue to monitor the usage.
- Start defining requirements for new information.
- Repeat the first five process steps by either:
  - Adding incremental information to initial warehouse, or
  - Adding additional sets of information to solve new problems

Once this process has been implemented, the questions can be asked:

1. Question: When are you finished?

   Answer: NEVER (for a complete warehouse).

2. Question: When is it considered successful?

   Answer: When it is being used and when the users are creating new requirements.

3.7 Measuring data warehouse results

The only way we will know if we are successful is to monitor and measure the project. The measurements are both subjective and objective. Just like certain medical tests, some of these measures are invasive and may have negative consequences, such as an impact on performance or the stability of the system. Some measures are costly; they require knowledgeable people and machine resources to carry them out. The clever project manager will select the appropriate metrics by evaluating both cost and impact. The metrics that should be considered are:

1. **Functional quality**

   Do the capabilities of the data warehouse satisfy the user requirements? Does the data warehouse provide the information necessary for the users to do their job?

2. **Data quality**

   If the data warehouse data is of poor quality, the users will reject it. There are two means of measuring quality:

   - Ask the users if their reports are accurate.
   - Use a software tool to provide a scorecard on the quality of the data.
   - Be aware that the software tools cannot evaluate all types of data quality.

3. **Computer performance**

   There are four indicators of performance we should consider:

   - Query response time
   - Report response time
   - Time to load/update/refresh the data warehouse
   - Machine resource

Some organizations have established benchmark performance numbers for known queries and reports, and they exercise and measure these benchmarks while periodically looking for impending performance problems. There are a number of tools that measure performance. Most of the database management systems have imbedded capabilities to measure database performance. Third-party utilities supplement this capability. A number of the query and report tools have response time metrics.

4. **Network performance**

   The ability of the network to handle the data traffic will directly impact response time. Network software measures line load, line traffic and indicates conditions where an activity was waiting for line availability. Besides the software, network administrators must be available to analyze the results and take appropriate action.

5. **User satisfaction**

   Users must be polled shortly after being given the data warehouse capability and then polled periodically to identify changes in their level of satisfaction and to watch trends.

6. **Number of queries**

   Many of the query tools provide metrics on the number of queries executed by department and by individuals.

7. **What data is accessed**

   Many organizations have data that is never accessed. This is the result of inaccurate or incomplete requirements gathering or the users changing their minds. Sometimes IT loads all of the source data fearing the user will ask for something they did not anticipate in the requirements gathering phase. IT has been beaten up so often by the users "who want all the data, want to keep it forever, and want the system delivered yesterday." There are tools that will identify what data is actually being accessed and how often.

8. **Satisfies scope agreement**

   A scope agreement documents what functions the users will be getting and when. It is appropriate to review the scope document and determine which functions might not have been satisfied and why.

9. **Benefits achieved**

   Before the project began you estimated the benefits, both tangible and intangible, for your project. Now you need to measure the tangible benefits and make some approximations for the intangibles. Since the benefits will not materialize the first day the system is installed, measurement should wait at least two months after implementation.

No project is perfect initially. There are always opportunities for improvement. In many cases, these improvements are necessary for the project to even survive. By measuring results, you and your project will be in a position to know where the resources must be directed to make the necessary improvements that every data warehouse project needs.

Chapter 4. BI data sourcing/movement

All of the main BI related applications, such as a data warehouse, OLAP, and Data Mining, rely on consistent, clean data. These data are gathered from various data sources and are combined into the data warehouse. This chapter talks about the data movement, the process of gathering all information into one single data source, in detail.

4.1 Data replication — a definition

Data replication in relation to the data warehouse, has the following characteristics:

- **Control** — data replication ensures consistency of results, irrespective of when or how the data is copied or manipulated.
- **Management** — provides the ability to build and reuse function.
- **Flexibility** — allows mixing and matching of functions and techniques where needed.
- **Ease of maintenance** — enables a rapid, cost effective response to changes in the structure or location of the source or target databases.
- **Integration of metadata** — provides links to the metadata of both source and target data, using or producing the metadata as required.
- **Performance** — provides ways to support large data sources at a variety of levels of synchronization.
- **Variety of sources** — supports the wide variety of data sources, which are characteristic of today's IS environment through a single approach or through a consistent and interlinked set of approaches.
- **Business context** — preserves the relationships imposed by the business processes when data is replicated

4.2 Data replication process

The steps in the process of the data replication should be well defined. In logical sequence, these steps are:

1. **Identify the source data.**

   In data warehousing, the source data is at a minimum defined and, in general, usually exists prior to any attempt to replicate it. Therefore, the process of data replication must first focus on obtaining this definition from where ever it already exists, rather than enabling the creation of a new definition.

2. **Identify or define the target data.**

   In contrast, the target data often does not exist in advance of defining the replication process. Ideally, the structure of the target data should be defined through the data modeling process.

   However, in some instances, particularly when the replication process is used to populate some types of derived data, the definition of the target data structure may form part of the definition of the replication process. This step, therefore, must equally support both: acquiring an existing target data definition and creating such a definition if required.

3. **Create the mapping between source and target.**

   When the definitions of both source and target data are available, the next task is to define how the source data is transformed into the target data. This mapping definition is required to handle a variety of different types of transformation. These range from relatively simple physical types, such as from EBCDIC (extended binary-coded decimal interchange code) to ASCII (American standard code for information interchange), to rather complex processes that combine a number of pieces of source data to generate new data in the target environment.

4. **Define the replication mode.**

   There are two basic modes of data replication: **refresh and update**. Refresh mode involves a bulk transfer of data from source to target. Update mode identifies and transfers only changed data from the source to the target environment. It is often necessary to define, in advance, the mode of replication to be used. However, in some circumstances, this decision can be made at runtime. The choice of mode is based on the time dependencies found in the source data and required in the target.

5. **Schedule the process of replication.**

   The actual replication of data is usually scheduled to occur separately from the definition process (steps 1 to 4). In addition, the replication itself is often a repeated process, taking place at defined intervals, such as daily, weekly, monthly, and so on. The scheduling process should be capable of triggering immediate replication, but the mandatory requirement is for later and repeated triggering.

6. **Capture the required data from the source.**

   This is the first step in the actual replication process itself. The capture step, in common with the following steps, is expected to take place according to the defined schedule, and without further human intervention.

The underlying method for extracting the data is dependent on the technology of the source data source and the chosen mode of replication. In terms of its timeliness, capture can range from real-time (or synchronous), through near real-time, to batch (both asynchronous).

7. **Transfer the captured data between source and target.**

   The transfer process must support a fully heterogeneous environment, where the source and the target may reside on different types of machine, in different formats, and in different locations, and is connected by links of varying capability

8. **Transform the captured data based on the defined mapping.**

   The transformation step may take place in the source environment or the target environment, or it may be distributed over both. Different types of transformation operate at different levels: changing one or more fields within a single record, combining records from different sources, and aggregating records.

9. **Apply the captured data to the target.**

   Data can be applied to the target in two basic ways:

   - Incoming data replaces existing data.
   - Incoming data is appended to existing data.

   The rationale and approach which is used depends on several factors: the required replication mode, the ways in which the data was captured, and the time dependencies of the source and target data.

10. **Confirm the success or failure of the replication.**

    Any of the above steps in the replication process may fail. Within the overall process, fall-back plans should exist to overcome specific failures. However, if the process cannot be completed, this information must be made available to the appropriate person.

11. **Document the outcome of the replication in the metadata.**

    The replication tool documents the success or failure of each step in the metadata. This provides the end user with information on the data currency in the target system.

12. **Maintain the definitions of source, target, and mapping.**

    As business needs change, there is a need to update the definition of the replication process to reflect changes in the source data, new requirements for the target data, and new data transformations.

4.3 Capture — an introduction

Capture is a component of data replication that interacts with a source data to obtain a copy of some or all of the data contained therein, or a record of changes that have occurred there (see Figure 20).

Capture

- Captures base table changes from journal
- Timestamps changes
- Maintains transaction consistency
- Automatically maintains staging tables

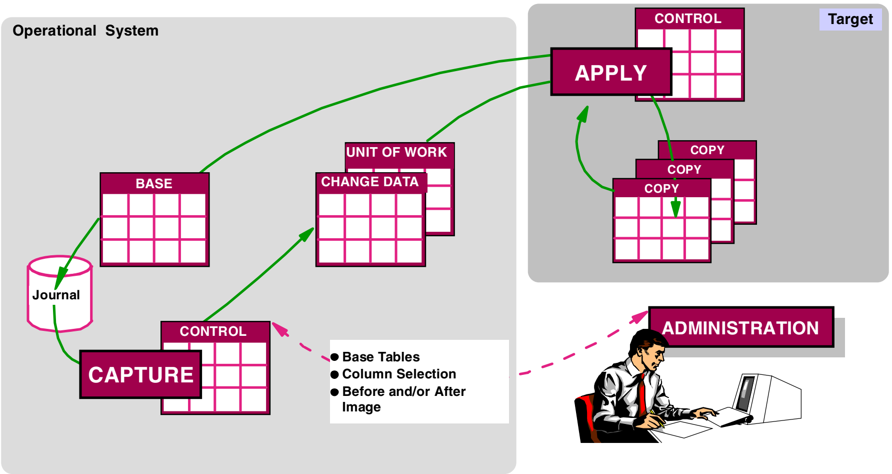


Figure 20. Capture

In general, not all of the data contained in the source is required. Although all of the data could be captured and the unwanted data then discarded, it is more efficient to capture only the required subset. The capture of such a subset, without reference to any time dependency of the source, is called **static capture**.

In addition, where databases change with time, we may need to capture the history of these changes. In some cases, performing a static capture on a repeated basis is sufficient. However, in many cases we must capture the actual changes that have occurred in the source. Both performance considerations and the need to transform transient or semi-periodic data into periodic data drive this requirement. This type is called **incremental capture**.

4.3.1 Static capture

Static capture essentially takes a snapshot of the source data at a point in time. This snapshot may contain all of the data found in the source, but usually it contains only a subset of the data.

Static capture occurs in a number of cases, although, it is not as common as the more complex incremental capture. Examples are:

Static capture occurs from the first time a set of data from a particular operational system is to be added to the data warehouse, where the operational system maintains a complete history of the data and the volume of data is small. In these cases, it is seldom necessary to move all data from a particular source to the target system. Many fields in the operational databases exist solely for technical reasons, which are related to maintaining integrity or improving performance. In legacy systems, there may be redundant and/or obsolete data, therefore, such data is irrelevant to the task of managing the business and need not be captured.

4.3.2 Incremental capture

Incremental capture is the method of capturing a record of changes that take place in a source data set. Incremental capture recognizes that most data has a time dependency, and thus requires an approach to efficiently handle this. The volume of changes in a set of data is almost always some order of magnitude smaller than the total volumes. Therefore, an incremental capture of the changes in the data rather than a static capture of the full resulting data set is more efficient. Incremental capture must collect the changes in such a way that applying these changes to the target builds a valid representation of the data in that environment. However, incremental capture is substantially more complex than static capture.

4.3.3 Delayed capture

Delayed capture occurs at predefined times, rather than with the occurrence of each change. In periodic data, this behavior produces a complete record of the changes in the source. In transient and semi-periodic data, however, the result in certain circumstances may be an incomplete record of changes that have occurred. These problems arise in the case of deletions and multiple updates in transient and semi-periodic data.

4.3.4 Data capture techniques

There are several data capture techniques that we will discuss in the following section. We previously distinguished between static and incremental capture. Here we will further discuss the subject in detail.

Static capture is the simplest technique; its basic functionality is in subsetting the captured data.

Incremental capture, however, is not a single topic. On closer examination, it can be divided into five different techniques as shown below, each with its own strengths and weaknesses. The first three types are immediate capture — changes in the source data are captured immediately after the event causing the change to occur. Immediate capture guarantees the capture of all changes made to the operational system, irrespective of whether the operational data is transient, semi-periodic, or periodic.

1. **Application-assisted capture** — depends on the application that changes the operational data to also store the changed data in a more permanent manner.
2. **Triggered capture** — depends on the database manager to store the changed data in a more permanent manner.
3. **Log/journal capture** — depends on the database manager's log/journal to store the changed data. Because of their ability to capture a complete record of the changes in the source data, these three techniques (techniques 2 - 4) are usually used with incremental data capture. However, in some environments, technical limitations prevent their use. In such cases either of the following two delayed capture strategies can be used if the business requirement allow:
4. **Timestamp-based capture** — selects data that has changed based on timestamps provided by the application that maintains the data.
5. **File comparison** — compares versions of the data to detect changes.

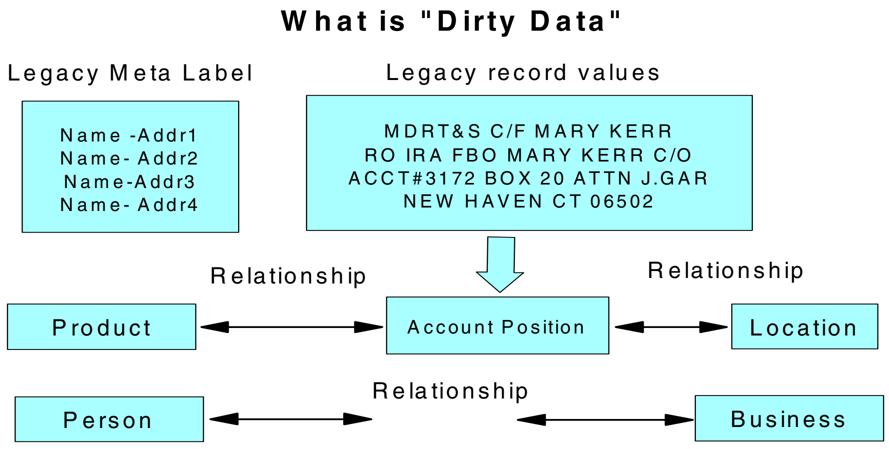
4.4 Cleansing

Consultants emphasize that data has become one of the most valuable corporate assets. It is used and reused in various business intelligence applications to support sophisticated analysis and decision-making processes that make the company more competitive. But the value of data is clearly dependent upon its quality (see Figure 21 on page 69 for a sample of ‘dirty data’). Decisions based on flawed data are suspect and can dearly cost the company. According to INFORMATION IMPACT International, Inc., "It has been demonstrated that non-quality data can cause business losses in excess of 20 percent of revenue and can cause business failure." With this in mind, it makes definite corporate sense to thoroughly cleanse any data prior to storing it in a secondary site, such as a data warehouse, and utilizing it in the decision-making process.

4.4.1 Perform data quality assessment

Anyone who researches the topic of data quality quickly learns that good data is a critical success factor in data warehousing. Experts recommend a quality assessment for any data collected for a data warehouse or data mart. The curious fact is that many IT managers attempt to skip this quality assessment and cleansing phase.

One of the main reasons for this omission might be the amount of time that data cleansing takes and its often unpredictable cost. In a data management industry special report entitled "Data Integrity and Cleansing," Curt Hall writes, "By far the largest 'unexpected' labor cost in data warehousing involves data cleanup and its associated data loading processes." He estimates that as much as 70 percent of the effort is devoted to data cleansing and transformation.



Vality Technology Inc.

Figure 21. Dirty data

4.4.1.1 What is “Dirty Data” ?

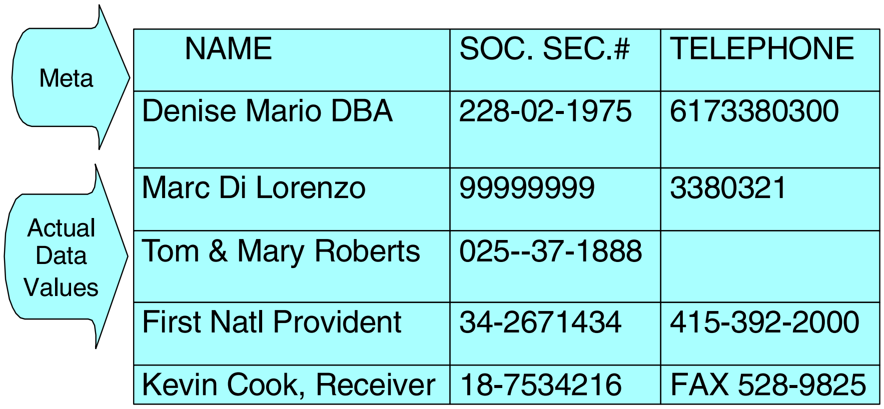
Ken Orr, coinventor of the Warnier-Orr Design methodology and data warehouse expert, says it best:

"Metadata can be described as a description of what you wish were in your data fields."

The problem may lie not only with the data but with the metadata (data about data — information on the fields names, file names, data, and so on). Some examples of this problem are:

- Legacy Info buried in freeform fields:

  How will you determine and extract entity relationships? Different keys from different data sources that identify the same entity are going to be a problem. Having multiple keys that links to some and not all of the data correctly is worse than having no keys. Figure 22 shows a sample of database sources with 'freeform' data fields.

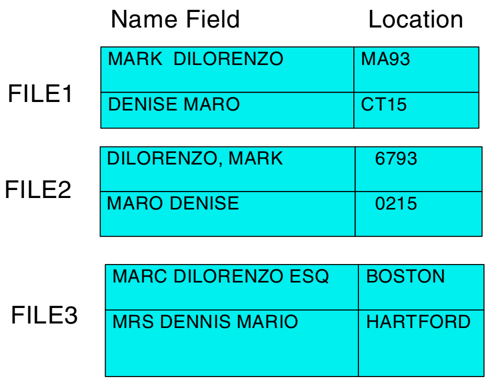

Meta
Actual Data Values

NAME	SOC. SEC.#	TELEPHONE
Denise Mario DBA	228-02-1975	6173380300
Marc Di Lorenzo	99999999	3380321
Tom & Mary Roberts	025--37-1888	
First Natl Provident	34-2671434	415-392-2000
Kevin Cook, Receiver	18-7534216	FAX 528-9825

Figure 22. Freeform fields

- Lack of legacy standards

  Unlimited formats, structures, attributes and code sets are contained within the fields with the same labels creating a nightmare for consolidation, and movement of the data to the warehouse. Figure 23 shows an example of three different data sources, all expected to providing the same information.



	Name Field	Location
FILE1	MARK DILORENZO	MA93
	DENISE MARO	CT15
FILE2	DILORENZO, MARK	6793
	MARO DENISE	0215
FILE3	MARC DILORENZO ESQ	BOSTON
	MRS DENNIS MARIO	HARTFORD

Figure 23. Lack of standards

4.4.2 Data cleansing techniques

The first step in data cleansing is analyzing the operational data to determine what type of discrepancy that could exist when the data is extracted, transformed, and then loaded into the data warehouse (see Figure 24).

Working in conjunction with the users at this time could help determine what to look for, which will be based on the subject area.

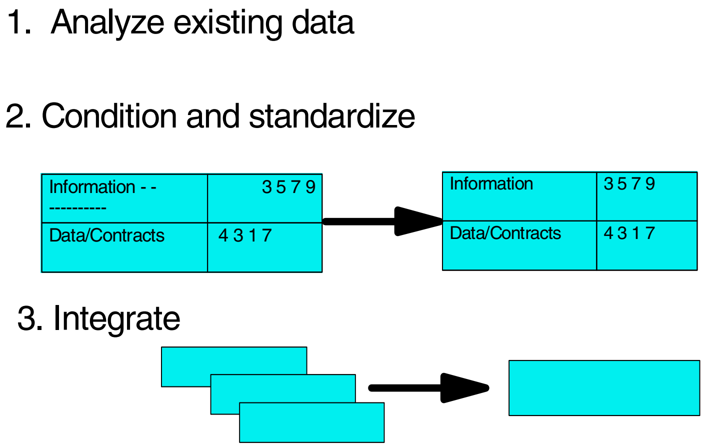


Figure 24. Data cleansing

4.4.2.1 Data conditioning

Data conditioning and standardization is necessary to keep, for example, certain allowed ranges for some fields. This can best be accomplished by using re-engineering tools that perform the following tasks:

- Parsing
- Data typing
- Pattern analysis
- Business rule discovery

4.4.2.2 Data integration

Data integration may include the merging and purging of data from different sources. The removal of redundant data can be accomplished by using the following techniques:

- De-duplication (merge/purge)
- Matching (integration by entity)
- Householding (integration by group)

4.5 Data transformation

The transformation component of data replication sits logically between *capture* and *apply*. It accepts data in the format of the source from the *capture* component and changes it to the format that *apply* will use in the target. In data warehousing, the changes in format required can range from the very simple to the highly complex.

4.5.1 Transformation functions

Transformation encompasses a wide and varied range of functionality. Ultimately, the function exists to meet a business requirement. To distinguish between different types of functionality requires descending to a more technical level of requirement definition. Classification of this functionality and assessment of its technical complexity depends on two factors:

- The relationship between the number of input and output records
- The types of computation applied at a field level within the records

From a consideration of this mix of business and technical requirements, six general transformation functions emerge:

- Selection
- Separation/concatenation

- Normalization/denormalization
- Aggregation
- Conversion
- Enrichment

The first four functions represent different ways in which input and output relate at a record level, while the last two operate at a field level.

4.5.1.1 Selection

Selection (or subsetting) is the process that partitions data according to predefined criteria. Selection is the simplest form of transformation, operating on one input record and generating at most one output record for each input record. Selection is often included as part of the capture component. However, there are circumstances where selection may take place in a subsequent, separate step from capture. For example, the technical structure of the source data may make it difficult to select the required subset. In this case, it may be more appropriate to capture all the data, convert it into a more useful format, and then select the subset required. Similarly, if a subset of the source data has been captured, and it needs further subdivision in order to feed multiple targets with different subsets of the data, then further selection is needed as part of the transformation phase. *Entity*, *attribute*, and *occurrence* are the different dimensions of selection.The definition of these dimensions are as follows:

Entity dimension — the simplest way to subset the data is by entity. In this case, all information pertaining to a particular subject is selected for capture. The output is an image of the total contents of the entity, often in a simpler structure than in the source. In DB2 SQL, selecting along the entity dimension equates to:

```
SELECT          *

FROM            CUSTOMER_FILE
```

Attribute dimension — in the attribute dimension, some attributes are selected from all occurrences of the data for a selected entity. In DB2 SQL, selecting along this dimension equates to:

```
SELECT          NAME, CITY, STATE

FROM            CUSTOMER_FILE
```

Occurrence dimension — in the occurrence dimension, all of the data about a selected set of occurrences is selected. In general, occurrences are selected based on the contents of one or more fields in each record. In DB2 SQL, selecting along this dimension equates to:

```
SELECT          *

FROM            CUSTOMER_FILE

WHERE            CITY = 'New York'
```

4.5.1.2 Separation/concatenation

Separation splits the information relating to one business item into a number of separate records based on the business key. Subsetting is done to simplify an end user's view, which support different data uses or for security reasons.

Concatenation is the reverse of separation, joining information about the same item together. The concatenation process allows an input record to be extended with more details about the primary subject. For example, different types of product information may be stored and maintained in different operational databases — packaging types and sizes for a product is stored in the manufacturing systems, while prices come from the marketing systems. These are concatenated based on the product number as the key in the data warehouse.

4.5.1.3 Normalization/denormalization

Normalization involves splitting a single input record into multiple outputs. An example of normalization is to take a product inventory record (keyed on product number), which contains both a description of the product and details of where it is stocked. The normalization process splits this record into two parts, one part contains the product data and the keyed on product number. The other contains data on stocking locations and the keyed on location code. Now, in order to maintain the relationship between product and stocking location, one of the output records must contain the key of the other record (as a foreign key relationship), otherwise, a third record link will be required.

Denormalization is the opposite of this. For example, names and addresses from a customer file may be added to a sales record whose input contains only customer number. Information contained in this data set is not confined to a single set of related data, but there may exist a number of sets of such data in a structure known as repeating groups. This is a many-to-one record transformation, taking a number of records from two or more normalized tables and joining them on the basis of their keys into one record to optimize on the performance of the system.

4.5.1.4 Aggregation

Aggregation is the transformation process that takes data from a detailed level to a summary level. Mathematically, aggregation consists of grouping the data according to some criterion and totaling, averaging, or applying some other statistical method to the resultant set of data. Examples include:

- Summaries of sales by time period (daily, weekly, monthly)
- Summaries of expenses by geographical area
- Averages of productivity by organizational unit or other grouping

The aggregation function can operate only on sets of identically structured records. For each set of detailed input records, the output is a smaller set of summary records, where significant amounts of detail have been removed. Figure 25 shows a sample data aggregation schema.

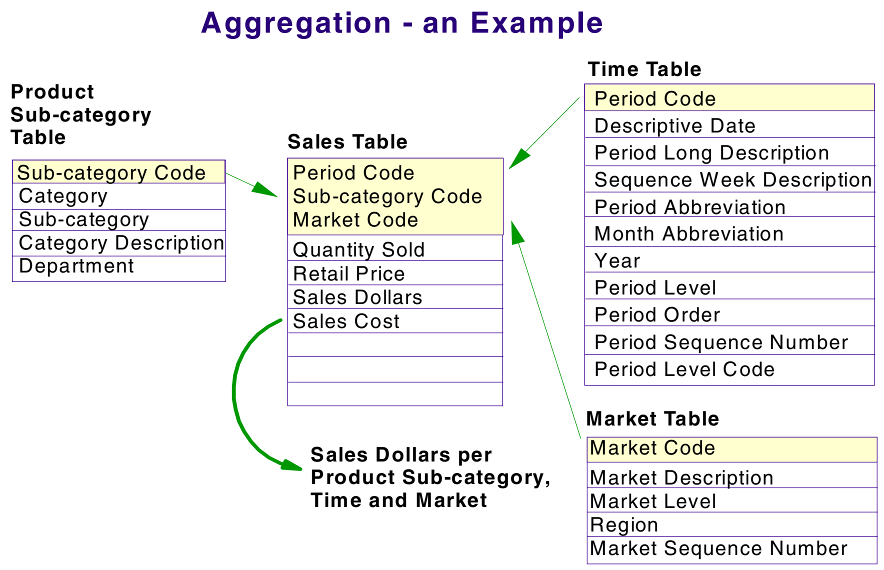


Figure 25. Aggregation

4.5.1.5 Conversion at field level

Conversion operates at a **field level** as opposed to the above mentioned aggregation, which operates at a record level. Its function is to change data from one form to another. The process takes input from a single data field and applies a rule to transform it into another form. This rule may be an algorithm or a lookup table.

Algorithmic conversion:

All the logic required in algorithmic conversion can be included within the conversion process itself. Examples of this type of conversion:

- Converting mixed-case text to all uppercase or all lowercase
- Converting between measurement systems, such as imperial to metric
- Converting from codes to description, where the set of possible values is small and unchanging, such as m to male or f to female

Conversion by lookup:

Conversion that cannot be expressed as a simple algorithm instead uses a lookup table. This approach provides a greater level of flexibility in the relationship between input and output, particularly in its ability to easily extend the relationship over time. Examples are:

- ASCII to EBCDIC and similar code-page conversions.
- Converting from codes to descriptions with a large or open-ended set of code values, such as ISO (International Standardization Organization) country codes to country names.

4.5.1.6 Enrichment

Enrichment is a transformation function that combines data from two or more fields in one or more records to create a new field or fields in the output record. Enrichment can be categorized into a single record, multi-field or multi-record cases.

Single record enrichment:

In single record enrichment all of the required input information comes from one record. This is technically the simplest type of enrichment because the input fields all come from the same source and are guaranteed to be available to the transformation component at the same time. The single record enrichment can be divided into single-field and multi-field cases.

Single-field enrichment usesinput from a single field within the one record and creates one or more new fields that represent a different view of the data.

Multi-field enrichment allows interaction between the fields within a single input record. The outcome may be the creation of a new field in the output record or the update of an existing field. Examples of this type of enrichment are:

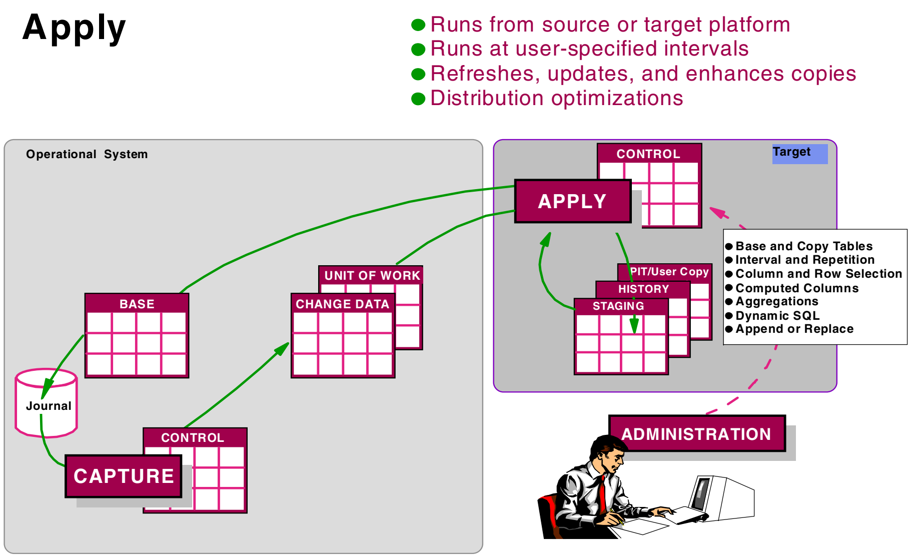
- Creation of a field containing a demographic category, where the category depends on a combination of the age, sex, and income.
- Extension of a product description field to include attributes, such as weight, color, or size, which are to be found in separate input fields.

Multi-record enrichment:

In multi-record enrichment, the input fields are no longer restricted to the same source record, but may come from more than one record. The output of multi-record enrichment is always the creation of one or more new fields in the output. In some cases, the new field represents business information that previously did not exist. An example is creating a sales analysis code that represents the success of selling different product types into different market segments, requiring a logical combination of product sales numbers and customer market segmentation data.

4.6 Apply — an introduction

The *apply* component of data replication takes the output of the capture component (either directly or via transformation and/or data transfer) and applies this data to the target system, as shown in Figure 26.



Apply

- Runs from source or target platform
- Runs at user-specified intervals
- Refreshes, updates, and enhances copies
- Distribution optimizations

- Base and Copy Tables
- Interval and Repetition
- Column and Row Selection
- Computed Columns
- Aggregations
- Dynamic SQL
- Append or Replace

Figure 26. Apply

Apply operates in one of four modes. These modes are listed below in order of increasing technical complexity:

1. Load

   In load, *apply* loads or reloads the target data set, so that any existing target data is completely replaced by the incoming captured data. Load is the simplest and broadest type of apply.

2. Append

   In append, *apply unconditional* appends the incoming captured data to the existing target data. The existing data is preserved, but depending on the contents of the captured data and the DBMS of the target, new records may duplicate existing ones or may be rejected.

3. Destructive merge

   In this mode, *apply* merges the incoming captured data into the existing target data. Where the keys of the existing and incoming data match, the existing data is updated accordingly; where they do not match, new records are added.

4. Constructive merge

   This mode is similar to destructive merge, but with one important difference. Where the keys of existing and incoming data match, *apply* marks existing data as superseded but does not overwrite it. Incoming records are therefore always added to the target.

The choice of which mode to use in any particular circumstance depends on the type of time dependency required in the target data:

- Snapshot data

  A snapshot, a static view of the data at a point in time, is created through the load mode. After the initial load, *append* can expand the snapshot by using incoming data from a different source.

  A snapshot remains static and at some stage is either deleted or replaced by a set of data for another point in time. This replacement could result from a second load process, or a set of changes could be merged into the existing snapshot. This is a destructive merge, because a snapshot does not preserve any historical view data.

- Transient data

  Transient data is also created through a load and optional append and maintained via destructive merge. From the viewpoint of the apply component, transient and snapshot data are virtually indistinguishable.

The only difference is the update frequency: transient data is updated on an ongoing basis, while a snapshot is updated at intervals.

- Periodic data

  Periodic data — a historical view of the business — is also created by a load. After the initial load, an append can expand the data using incoming data from a different source, under certain conditions. Constructive merge is the most effective mode for maintaining periodic data, because records must never be deleted from periodic data. Append can also update periodic data, but only if a mechanism exists to mark superseded records after the captured data has been appended.

4.6.1 Methods for loading

There are two utilities that can be used to populate tables in DB2 databases:

- The LOAD utility
- The IMPORT utility

The LOAD utility is used for loading or appending data to a table where large amounts of data will be inserted. The LOAD utility can move data into tables, create an index, and generate statistics.

4.6.1.1 LOAD Utility

There are three phases of the load process, shown in Figure 27 on page 80.

1. **Load**, when the data is written into the table.
2. **Build**, when the indexes are created.
3. **Delete**, when the rows that caused a unique constraint violation are removed from the table.

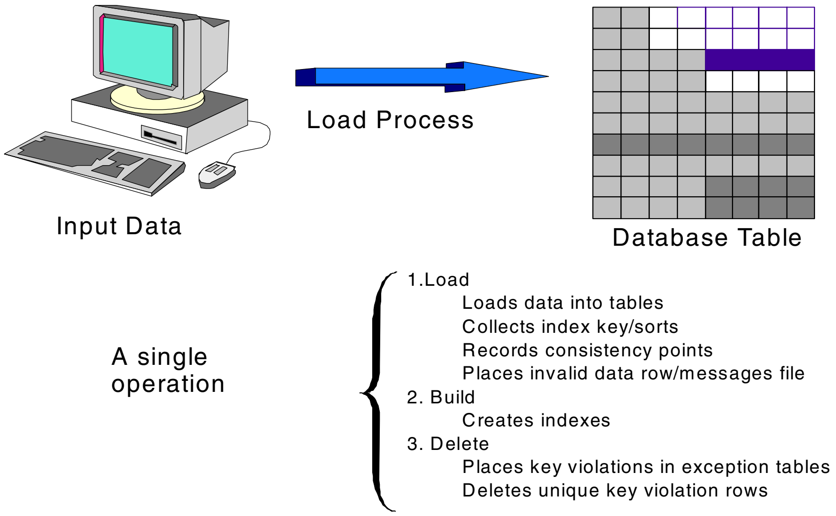


Figure 27. Load

All phases of the load process are part of one operation that is completed only after all three phases complete successfully. The LOAD utility will generate messages during the progress of each phase. Should a failure occur during one of the phases, then these messages can assist one in deciding the recovery actions available.

Load Phase

During the load phase, data is stored into a table and index keys are collected. Messages let you know how many input rows have been successfully loaded during the operation.

Build Phase

During the build phase, indexes are created based on the index keys collected in the load phase. The index keys are sorted during the load phase. If a failure occurs, the build must be restarted from the beginning of the build phase.

Delete Phase

During the delete phase, all rows that have violated a unique constraint are deleted.

The input data for the load process must be in one of the three file formats:

- **Integrated Exchange Format (IXF):** This is the preferred method for exchange between relational database managers. You can export a data file from a host database to the DB2 UDB server. In general, an IXF file consists of an unbroken sequence of variable length records. An IXF file also has the table definition stored within it along with the data.
- **Delimited ASCII (DEL):** This type is used for exchanging files with a wide variety of industry applications, especially other database products. This is a commonly used way of storing data that separates column values with a special delimiting character.
- **Non-delimited ASCII (ASC):** Non-delimited ASCII files are used for loading data from other applications that create flat text files with aligned column data, such as those produced by word processing programs. Each ASCII file is a stream of ASCII characters consisting of data values organized by row and column. Rows in the data stream are separated by a line feed.

4.6.1.2 Load balancing

DB2 with its parallel data loader allows all available processors to be used simultaneously to load a single data table. This feature should be used when loading large volumes of data into the data warehouse.

The input data file is partitioned (split into several smaller files), the loading then can process each partition in parallel, hence, reducing the time required to load the data.

4.7 Data model

The purpose of modeling is to provide an accurate record of some aspect of the real world in some particular context. This provides the user of the model with a clearer understanding of how the modeled objects behave, along with the ability to predict the consequences of any action with the environment and the impacts of any change to it.

Business data modeling provides a view of the business that focuses on the data used, allowing the design of applications that support the way the business operates. Business data modeling, therefore, aims to provide:

- A record of accurate and meaningful business data definitions.
- Identification of valid, consistent business data structures that contain sufficient information to run and manage the business.
- An indication of the similarities and differences between data from different sources and the relationships between them.

Business process modeling focuses on business activities, providing:

- A record of accurate and meaningful business process definitions
- Identification of the relationships between and within business processes

These models are closely related because any process will use certain data. Identifying which data is created or modified in particular processes is a particularly important aspect of that relationship.

Entities, attributes, and relationships:

The most common forms of business data modeling use the entity relationship approach (Chen 1976). In this approach (shown in Figure 28) an entity is any category of an object in which the business is interested. Each entity has a corresponding business definition, which is used to define the boundaries of the entity — allowing one to decide whether a particular object belongs to that category or entity. Figure 28 shows an entity called Product.

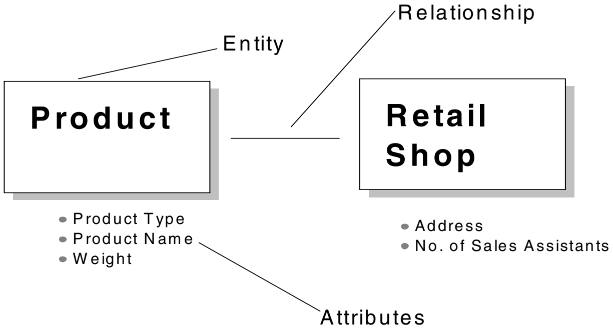


Figure 28. Example of entity relationship model

Product is defined as any physical item that may be stocked in one or more of the retail stores in the company. Whether this definition is appropriate or not depends on the use to which the model is put. In this sense, an entity may be quite specific at one extreme, or very generic at the other extreme. Each entity has a number of attributes associated with it. An attribute is any characteristic of an entity that describes it and is of interest to the business.

The second major element of the entity relationship (ER) model is the relationship. A relationship exists between the entities in a model and describes how the entities interact. This interaction is usually expressed as a verb. In the example, the relationship between "Product" and "Retail shop" is defined as "Retail shop stocks Product".

Chapter 5. BI solution architecture

This chapter talks about the general BI solution architecture, which covers such aspects as:

- Approach by Industry
- Applications
- Tools

5.1 Business Intelligence application areas by industry

Let us first have a closer look at the approach for a BI solution architecture that is tailored according to the specific requirements of different industries.

5.1.1 Retail industry BI applications

In the retailing industry, management uses advanced information technologies to improve performance and achieve their objectives in a broad array of applications. The following may be particularly relevant at this time:

Forecasting. Using scanning data to forecast demand and, based on the forecast, to define inventory requirements more accurately.

Ordering and replenishment. Using information to make faster, more informed decisions about which items to order and to determine optimum quantities.

Marketing. Providing analyses of customer-specific transaction data. Enabling retailers to know not only what is selling but who is buying it. Strengthening consumer 'pull'.

Quick response. Quick response technologies can help speed merchandise to the shelf, reduce inventory levels needed to increase turns; requires powerful information system and strong communications backbone.

Merchandising. With quick, detailed access to sales and inventory data, can focus more precisely on store-ad-item level performance. Can buy more strategically, defining the right merchandise for the market at any point in time. Can refine inventory, do continuous merchandise planning and flow management.

Distribution and logistics. Helping distribution centers manage increased volumes. Can use advance shipment information to schedule and consolidate inbound and outbound freight.

Transportation management. Developing optimal load consolidation plans and routing schedules.

Inventory planning. Helping identify the inventory level needed, item by item, to ensure a given grade of service.

Stock location planning. Helping warehouse planners assign products to locations so as to minimize distances, improve efficiency.

Finished goods deployment. Balancing supply, demand, and capacity to determine how to allocate limited quantities of finished goods.

Space management. Planograms developed at headquarters will automatically adjust to reflect store-specific sales patterns and category space.

Card technology. Proprietary and frequent-shopper cards are 'swiped' at a POS device that instantly captures the customer and transaction information, from which customer purchase patterns can be projected.

5.1.2 Insurance BI applications

The typical questions the insurance industry looks to be answered by the BI solution mainly is related to do risk analysis for new customers. The overall objectives look like the following:

- **Claims and premium analysis**. The ability to analyze detailed claims and premium history by product, policy holder, claim type, and other specifics. Enabling the insurer to set reserves based on detailed, timely data. To analyze severity trends based on experience and reduce severity using a methodology that crosses line-of-business characteristics and integrates outside data sources accessed on-line. Pricing is then based on appropriate performance factors.
- **Customer analysis**. The ability to analyze client needs and product usage patterns. Develop marketing programs based on client characteristics. Conduct risk analysis and cause-of-loss determination across products. Produce profitability reports by client and identify opportunities. Provide customer-support personnel with detail information about each client, improving client service.
- **Risk analysis**. To understand the risk of introducing a new product or insuring a new customer. To identify high-risk market segments and opportunities in specific segments. Relate market segments to each other and qualify their combined risk. Project trends based on external data. To reduce frequency of claims.

5.1.3 Banking, finance and securities BI applications

The banking, finance, and security industry usually is mostly interested in the analysis of profitability such as the following:

- **Customer profitability analysis**. The ability to understand the overall profitability of an individual customer/household, current and long term. To consolidates activity based costing and sales data for a year or more. Provides the basis for high-profit sales and relationship banking. Maximizes sales to high-value customers; reduces costs to low-value customers. Provides the means to maximize profitability of new products and services.
- **Credit management**. The ability to understand credit issues by product. Establishes patterns of credit problem progression by customer class and type. Provides early warnings to help customers avoid credit problems. Provides the means to manage credit limits as conditions improve or deteriorate. Provides a more accurate valuation of the bank's credit portfolio. Forecasts the impact of change in credit policy. Reduces credit losses.
- **Branch sales**. Providing customer information to the branch for improving customer service and account selling. Facilitates cross selling. Reduces paper work. Improves customer support. Strengthens customer loyalty. Improves sales.

5.1.4 Telecommunications industry BI applications

The telecommunications industry is looking for information such as:

- **Customer profiling and segmentation**. The ability to analyze customer and product usage history to determine high-profit product profiles and customer segments. Provides detailed, integrated customer profiles based on product-usage history, competition, and channel patterns. Performs longitudinal analysis of residential and personal calling behavior. Allows the development of individualized frequent-caller programs. Links to highly targeted marketing for focused customer segments. Provides the means for understanding future personal, household, and related business calling needs. Provides information for effective incentives analysis and management.
- **Customer demand forecasting**. The ability to analyze customers' historical product usage to forecast future product needs or service activity. Provides basis for churn analysis and control for improving customer retention. Permits in-depth understanding to identify needed new products and services. Links to network investment, price modeling, and competitive analysis.

5.1.5 Manufacturing industry BI applications

In the manufacturing industry, management is using advanced information technologies to improve performance and achieve their objectives in a broad array of applications. The following may be particularly relevant at this time:

- **Sales/Marketing**. Providing analyses of customer-specific transaction data. Enabling retailers to know not only what's selling but who's buying it. Strengthening consumer 'pull'.
- **Forecasting**. Using scanning data to forecast demand and, based on the forecast, to define inventory requirements more accurately.
- **Ordering and replenishment**. Using information to make faster, more informed decisions about which items to order and optimum quantities.
- **Purchasing/Vendor Analysis**. Helping purchasing managers understand the different cost and timeliness factors of each of their parts suppliers.
- **Distribution and logistics**. Helping distribution centers manage increased volumes. Can use advance shipment information to schedule and consolidate inbound and outbound freight.
- **Transportation management**. Developing optimal load consolidation plans and routing schedules.
- **Inventory planning**. Helping identify the inventory level needed, item by item, to ensure a given grade of service.
- **Stock location planning**. Helping warehouse planners assign products to locations so as to minimize distances, improve efficiency.
- **Finished goods deployment**. Balancing supply, demand, and capacity to determine how to allocate limited quantities of finished goods.

5.2 Business Intelligence product set

We will take a look at the products and tools provided by IBM (and its key partners) for supporting a business intelligence software environment — these products are listed in Figure 29. We will use the IBM Business Intelligence Structure to categorize and describe these products.

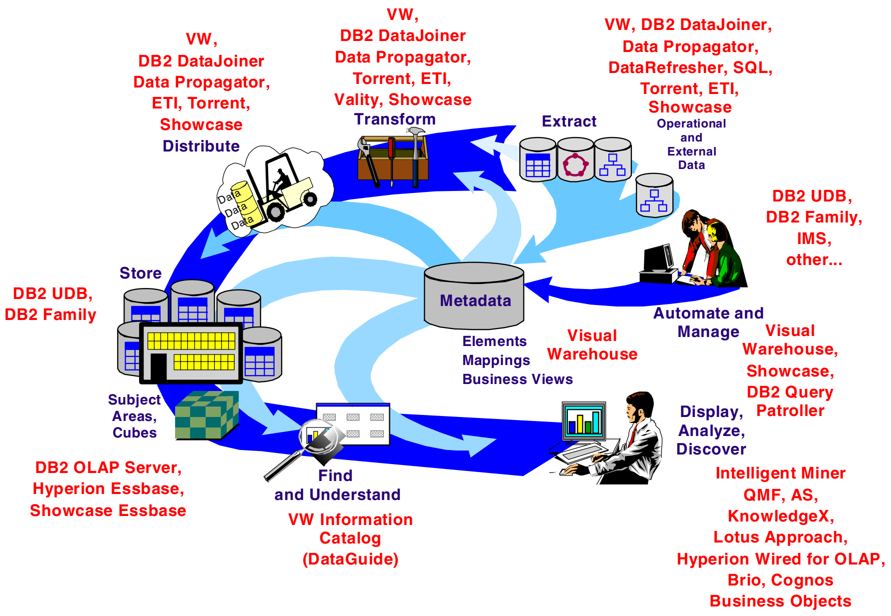


Figure 29. Business intelligence product set

5.2.1 Business Intelligence applications

IBM's business intelligence applications are marketed under the DecisionEdge brand name. **DecisionEdge is a Customer Relationship Management (CRM)** solution that allows organizations to analyze consumer behavior with the objective of increasing market share and customer profitability. To date, IBM has announced DecisionEdge packages for the finance, insurance, telecommunications, and utilities industries. Each DecisionEdge offering provides integrated hardware, software, consulting services, and business applications centered on an industry-specific data model. **DecisionEdge for telecommunications**, for example, analyzes customer information measuring profitability, predicting customer behavior, analyzing attrition, and assists in the creation of tailored customer marketing programs. **DecisionEdge for Finance**, **Banking, and Securities** offers pre-defined solutions in the areas of marketing and sales, and risk and profitability analysis. All DecisionEdge packages support the OS/390, AS/400, UNIX, and Windows NT operating environments, and include the VALEX marketing automation and campaign management software developed by Exchange Applications.

DecisionEdge also capitalizes on IBM's heavy investment in information mining research. By utilizing the Intelligent Miner development environment, DecisionEdge provides the optional **Intelligent Miner for Relationship Marketing** application to help the business user obtain a better understanding of key business issues such as customer segmentation, and potential buying and loyalty behavior. IBM is placing increasing emphasis on the use of business intelligence applications and is bringing applications to market in several industry areas including student administration, retail banking, local and state human services, and e-commerce. Business intelligence applications are also available for the DB2 OLAP Server (see description below). This product (which was developed by IBM and Hyperion Solutions) employs the same API as Hyperion Essbase, and it can, therefore, be used with the many industry-specific third-party application packages available for Essbase.

5.2.2 Business Intelligence tools

Business intelligence tools can be broken down into three categories:

- Query and reporting
- Online analytical processing (OLAP)
- Information mining

5.2.2.1 Query and Reporting

The main IBM query and reporting offering is the **Query Management Facility** (QMF) family of tools. Recently, IBM introduced QMF for Windows, a native Windows version of QMF that supports access not only to DB2 databases, but it also supports any relational and non-relational data source supported by its DB2 Data Joiner middleware product (see description below). QMF host objects are compatible with QMF for Windows, extending the enterprise query environment to Windows and the Web. Output from QMF can be published to the Web, and can be passed to other Windows applications like Lotus 1- 2-3, Microsoft Excel, and other desktop products via Windows OLE.

To increase the scope of its query and reporting offerings, IBM has forged relationships with Brio Technology, Business Objects, and Cognos. IBM intends the relationships with these tool vendors to be more than mere joint marketing deals — they also involve agreements to integrate the products from these companies with IBM’s business intelligence offerings, for example, in the area of metadata interchange.

5.2.2.2 Online Analytical Processing (OLAP)

IBM's key product in the OLAP marketplace is the **DB2 OLAP Server**, which implements a three-tier client/server architecture for performing complex multidimensional data analysis. The middle tier of this architecture consists of an OLAP analytical server developed in conjunction with Hyperion Solutions, which is responsible for handling interactive analytical processing and automatically generating an optimal relational star schema based on the dimensional design the user specifies. This analytical server runs on Windows NT or UNIX and can be used to analyze data managed by a DB2 Universal Database server. Support for Oracle servers is planned for a future release. The DB2 OLAP Server supports the same client API and calculation engine as Hyperion Essbase, and any of the many third-party GUI- or Web-based tools that support Essbase can act as clients to the DB2 OLAP Server.

The value of the DB2 OLAP server lies in its ability to generate and manage relational tables that contain multidimensional data, in the available Essbase applications that support the product, and features within Visual Warehouse for automating the loading of the relational star schema with information from external data sources such as DB2, Oracle, Informix, IMS, and VSAM.

5.2.2.3 Information mining

IBM has put significant research effort into its **Intelligent Miner for Data** product, which runs on OS/390, OS/400, UNIX and Windows NT, and can process data stored in DB2 databases, any relational database supported by DB2 Data Joiner, and flat files. Intelligent Miner Version 1, released in 1996, enabled users to mine structured data stored in relational databases and flat files, and offered a wide range of different mining algorithms. Intelligent Miner Version 2 features a new graphical interface, additional mining algorithms, DB2 Universal Database exploitation, and improved parallel processing. Intelligent Miner is one of the few products on the market to support an external API, allowing result data to be collected by other products for further analysis (by an OLAP tool, for example). Intelligent Miner has good data visualization capabilities, and unlike many other mining tools, supports several information mining algorithms.

IBM also offers its **Intelligent Miner for Text** product, which provides the ability to extract, index, and analyze information from text sources such as documents, Web pages, survey forms, and so on.

5.3 Access enablers

Client access to warehouse and operational data from business intelligence tools requires a client database API. IBM and third-party business intelligence tools support the native DB2 SQL API (provided by IBM's Client Application Enablers) and/or industry APIs like ODBC, X/Open CLI, and the Hyperion Essbase and ESRI APIs.

Often, business information may be managed by more than one database server, and IBM's strategic product for providing access to this data is its **DB2 Data Joiner** middleware server, which allows one or more clients to transparently access data managed by multiple back-end database servers. This *federated* database server capability runs on Windows NT, OS/400, and UNIX, and can handle back-end servers running IBM or non-IBM data products, for example, IBM DB2, Informix, Microsoft SQL Server, Oracle, Sybase, VSAM, IMS, plus any ODBC, IBI EDA/SQL or Cross Access supported data source. Features of this product that are worthy of note include:

- Transparent and heterogeneous database access using a single dialect of SQL.
- Global optimization of distributed queries with query rewrite capability for poorly coded queries.
- Stored procedure feature that allows a global DB2 Data Joiner procedure to transparently access data or invoke a local procedure on any DB2 Data Joiner-supported database. This feature includes support for Java and Java Database Connectivity (JDBC).
- Heterogeneous data replication (using IBM DataPropagator, which is now integrated with DB2 Data Joiner) between DB2, Informix, Oracle, Sybase and Microsoft relational database products.
- Support for Web-based clients (using IBM's Net.Data product).

IBM's **Net.Data** Web server middleware tool (which is included with DB2) supports Web access to relational and flat file data on a variety of platforms, including DB2, DB2 DataJoiner-enabled databases, and ODBC data sources. Net.Data tightly integrates with Web server interfaces, and supports client-side and server-side processing using applications written in Java, REXX, Perl, C++, or its own macro language.

5.4 Data warehouse modeling and construction

IBM supports the design and construction of a data warehouse using its Visual Warehouse product family and data replication tools, and via third-party relationships with Evolutionary Technologies International (for its ETI²EXTRACT Tool Suite) and Vality Technology (for its Integrity Data Reengineering tool).

The **Visual Warehouse** product family is a set of integrated tools for building a data warehouse, and includes components for defining the relationships between the source data and warehouse information, transforming and cleansing – acquired source data, automating the warehouse load process, and managing warehouse maintenance. Built on a DB2 core platform, Visual Warehouse can acquire source data from DB2, Informix, Microsoft, Oracle, Sybase, IMS databases, VSAM and flat files, and DB2 Data Joiner-supported sources.

Organizations have the choice of two Visual Warehouse packages, both of which are available with either Brio Technology, Business Objects or Cognos add-ins for information access. The base package, Visual Warehouse, includes:

- DB2 Universal Database for metadata storage.
- A Visual Warehouse Manager for defining, scheduling, and monitoring source data acquisition and warehouse loading operations.
- A Visual Warehouse agent for performing the data capture, transformation and load tasks.
- The Visual Warehouse Information Catalog (formerly known as DataGuide) for exchanging metadata between administrators and business users.

The second package, **Visual Warehouse OLAP**, adds the DB2 OLAP Server to the mix, allowing users to define and load a star schema relational database, as well as to perform automatic precalculation and aggregation of information as a part of the load process.

Visual Warehouse provides several features that make the implementation and management of a data warehouse more efficient: its use of agent technology, its management capabilities, its handling of metadata, and its ability to invoke user-written and third-party tools to perform additional processing outside the scope of the product.

The first of these, its use of agent technology, is intended to satisfy the performance requirements for loading large warehouse information stores. Data is acquired and loaded into an information store by warehouse agents whose job it is to move information directly from one or more data sources to one or more warehouse information stores. Unlike many competing products, information does not have to pass through a central intermediate server that might otherwise become a performance bottleneck as data volumes grow. Visual Warehouse agents run on OS/400, OS/2, UNIX, and Windows NT, and, depending on the volumes of data being moved, any given implementation may have one or many agents running concurrently. The source data to be captured, transformed and loaded into the warehouse information store by one or more agents is defined in a business view. The definition, scheduling and monitoring of business view operations is handled by the Visual Warehouse Manager, which runs under Windows NT.

In addition to initiating agent activities, the Visual Warehouse Manager can also be used to schedule user-written data capture and transformation applications, as well as applications available from IBM business partners. This facility is employed by Visual Warehouse to enable the loading of Hyperion Essbase multidimensional data, and to integrate other non-agent-driven processing such as ETI²EXTRACT programs, IBM data replication jobs, and Vality data cleansing processes.

Visual Warehouse also plays a key role in managing the metadata associated with the IBM business intelligence environment. In such an environment there are two types of metadata to be managed — technical metadata and business metadata. Technical metadata is associated with the design, building and operation of a data warehouse, whereas business metadata is used in conjunction with the business intelligence tools used to access and analyze warehouse data.

The Visual Warehouse Manager employs its own DB2-based metadata store for managing the technical metadata associated with the building and managing of a data warehouse. As mentioned earlier, IBM has developed interfaces to products from Hyperion Solutions, Evolutionary Technologies International, and Vality Technology for metadata interchange with Visual Warehouse. Metadata can also be exchanged with business intelligence tools from Brio Technology, Business Objects, and Cognos.

Included with Visual Warehouse is the Visual Warehouse Information Catalog (formerly known as DataGuide). The objective of this information catalog is to document and manage the business and underlying technical metadata that helps business users access and exploit the business intelligence environment. Business users can browse this metadata using both graphical- and Web-based interfaces.

Metadata in the Visual Warehouse Information Catalog is stored in a DB2 database and can be accessed and maintained using supplied SQL and application APIs, and can be imported and exported using files formatted in a documented tag language. IBM supplies a variety of sample applications that use these interfaces to exchange metadata with third-party products (Hyperion Essbase, Bachman DBA, Microsoft Excel, for example). Visual Warehouse Manager's technical metadata can also be imported into the information catalog. With Visual Warehouse, IBM supports the Metadata Coalition's Metadata Interchange Specification (MDIS) for moving metadata into and out of the Visual Warehouse Information Catalog.

IBM's data replication capabilities are based on its **DataPropagator Relational** product, which has now been integrated into DB2 Universal Database (for homogeneous data replication), and DB2 Data Joiner (for heterogeneous data replication). The replication facility captures data changes from DB2 source databases, and applies those changes to a DB2-managed data warehouse. Data changes are transported from the source to the target warehouse via staging tables. SQL is used to retrieve and transform data from the staging tables and apply it to the DB2-based warehouse at user-defined intervals. DB2 Data Joiner can also act as a data source or target for the replication facility, which means it can be used to replicate data from a third-party relational DBMS to a DB2-based data warehouse, or to replicate data from a DB2 data source to a data warehouse managed by a non-IBM relational DBMS.

Other IBM products for data warehouse construction include **DataPropagator NonRelational**, for capturing data changes from IMS databases, and **Data Refresher** for capturing and transforming data stored in non-relational databases and files such as IMS and VSAM.

IBM partner Evolutionary Technologies International markets the **EXTRACT Tool Suite** for generating warehouse data capture and transformation applications. This consists of:

- A Data Conversion Tool for defining data cleanup and transformation rules and generating data acquisition programs.
- Pre-built Data System Libraries (DSLs) for key operating and database environments including SAP, IDMS, IMS, VSAM, and leading relational database products. A DSL defines the native access method to be used for processing data, the grammar for generating application programs, and the business rules available to the Data Conversion Tool.

- A Master ToolSet for extending, creating and maintaining DSLs.

IBM has been working with ETI to optimize the DB2 DSL (to support parallel loading, for example), and to integrate EXTRACT with Visual Warehouse in the areas of metadata interchange and EXTRACT program scheduling. One of the key benefits EXTRACT adds to Visual Warehouse is support for additional data sources and application packages such as SAP.

Vality's **Integrity** Data Reengineering tool complements both Visual Warehouse and ETI•EXTRACT by adding a capability to analyze the *content* of data extracted from operational systems and enhance the quality of data before it is loaded into a data warehouse. During the data reegineering process, unique data entities are identified in data from multiple systems, allowing the data to be merged, reconciled and consolidated, even when there is no common key to support the merge. Important metadata that is discovered in this process can be used to validate and adjust the data model for the data warehouse information store. As with ETI, IBM has worked with Vality to integrate Integrity with Visual Warehouse in the areas of metadata interchange and program scheduling.

5.4.0.1 Data management

Data management in the business intelligence environment is provided by **DB2 Universal Database**, which offers intelligent data partitioning and parallel query and utility processing on a range of IBM and non-IBM multiprocessor hardware platforms. DB2 Universal Database also supports both partition and pipeline parallelism, SQL CUBE and ROLLUP OLAP operations, integrated data replication, dynamic bit-mapped indexing, user-defined types, and user-defined functions.

The **DB2 Spatial Extender** enables geo-spatial data to be incorporated into a relational DBMS. The product is a joint development effort between IBM and Environmental Systems Research Institute (ESRI), a leading GIS developer. IBM is initially delivering the DB2 Spatial Extender on DB2 DataJoiner, and plans to add this capability to the next release of DB2 Universal Database. GIS tools and applications can use either an ESRI or an SQL API to access and analyze geo-spatial data. Existing tools and applications that support the ESRI API will also work unmodified with the DB2 Spatial Extender.

5.5 What is OLAP?

During the last ten years, a significant percentage of corporate data has migrated to relational databases. Relational databases have been used heavily in the areas of operations and control, with a particular emphasis on transaction processing (for example, manufacturing process control, brokerage trading). To be successful in this arena, relational database vendors place a premium on the highly efficient execution of a large number of small transactions and near fault tolerant availability of data.

More recently, relational database vendors have also sold their databases as tools for building data warehouses. A data warehouse stores tactical information that answers "who?" and "what?" questions about past events. A typical query submitted to a Data Warehouse is: "What was the total revenue for the eastern region in the third quarter?"

It is important to distinguish the capabilities of a Data Warehouse from those of an OLAP (On-Line Analytical Processing) system. In contrast to a data warehouse, which is usually based on relational technology, OLAP uses a multidimensional view of aggregate data to provide quick access to strategic information for further analysis.

OLAP enables analysts, managers, and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information. OLAP transforms raw data so that it reflects the real dimensionality of the enterprise as understood by the user.

While OLAP systems have the ability to answer "who?" and "what?" questions, it is their ability to answer "what if?" and "why?" that sets them apart from data warehouses. OLAP enables decision making about future actions.

A typical OLAP calculation is more complex than simply summing data, for example: "What would be the effect on soft drink costs to distributors if syrup prices went up by $.10/gallon and transportation costs went down by $.05/mile?"

OLAP and data warehouses are complementary. A data warehouse stores and manages data. OLAP transforms data warehouse data into strategic information. OLAP ranges from basic navigation and browsing (often known as "slice and dice"), to calculations, to more serious analyses such as time series and complex modeling. As decision makers exercise more advanced OLAP capabilities, they move from data access to information to knowledge.

5.5.1 Who uses OLAP and why?

OLAP applications span a variety of organizational functions. Finance departments use OLAP for applications such as budgeting, activity-based costing (allocations), financial performance analysis, and financial modeling. Sales analysis and forecasting are two of the OLAP applications found in sales departments. Among other applications, marketing departments use OLAP for market research analysis, sales forecasting, promotions analysis, customer analysis, and market/customer segmentation. Typical manufacturing OLAP applications include production planning and defect analysis.

Important to all of the above applications is the ability to provide managers with the information they need to make effective decisions about an organization's strategic directions. The key indicator of a successful OLAP application is its ability to provide information as needed, that is, its ability to provide "just-in-time" information for effective decision-making. This requires more than a base level of detailed data.

Just-in-time information is computed data that usually reflects complex relationships and is often calculated on the fly. Analyzing and modeling complex relationships are practical only if response times are consistently short. In addition, because the nature of data relationships may not be known in advance, the data model must be flexible. A truly flexible data model ensures that OLAP systems can respond to changing business requirements as needed for effective decision making.

Although OLAP applications are found in widely divergent functional areas, they all require the following key features:

- Multidimensional views of data
- Calculation-intensive capabilities
- Time intelligence

5.5.1.1 Multidimensional views

Multidimensional views are inherently representative of an actual business model. Rarely is a business model limited to fewer than three dimensions. Managers typically look at financial data by scenario (for example, actual vs. budget), organization, line items, and time; and at sales data by product, geography, channel, and time.

A multidimensional view of data provides more than the ability to "slice and dice"; it provides the foundation for analytical processing through flexible access to information. Database design should not prejudice which operations can be performed on a dimension or how rapidly those operations are performed. Managers must be able to analyze data across any dimension, at any level of aggregation, with equal functionality and ease. OLAP software should support these views of data in a natural and responsive fashion, insulating users of the information from complex query syntax. After all, managers should not have to understand complex table layouts, elaborate table joins, and summary tables.

Whether a request is for the weekly sales of a product across all geographical areas or the year-to-date sales in a city across all products, an OLAP system must have consistent response times. Managers should not be penalized for the complexity of their queries in either the effort required to form a query or the amount of time required to receive an answer.

5.5.1.2 Complex calculations

The real test of an OLAP database is its ability to perform complex calculations. OLAP databases must be able to do more than simple aggregation. While aggregation along a hierarchy is important, there is more to analysis than simple data roll-ups. Examples of more complex calculations include share calculations (percentage of total) and allocations (which use hierarchies from a top-down perspective).

Key performance indicators often require involved algebraic equations. Sales forecasting uses trend algorithms such as moving averages and percentage growth. Analyzing the sales and promotions of a given company and its competitors requires modeling complex relationships among the players. The real world is complicated -- the ability to model complex relationships is key in analytical processing applications.

5.5.1.3 Time intelligence

Time is an integral component of almost any analytical application. Time is a unique dimension because it is sequential in character (January always comes before February). True OLAP systems understand the sequential nature of time. Business performance is almost always judged over time, for example, this month versus last month, this month versus the same month last year.

The time hierarchy is not always used in the same manner as other hierarchies. For example, a manager might ask to see the sales for May or the sales for the first five months of 1995. The same manager might also ask to see the sales for blue shirts but would never ask to see the sales for the first five shirts. Concepts such as year-to-date and period over period comparisons must be easily defined in an OLAP system.

In addition, OLAP systems must understand the concept of balances over time. For example, if a company sold 10 shirts in January, five shirts in February, and 10 shirts in March, then the total balance sold for the quarter would be 25 shirts. If, on the other hand, a company had a head count of 10 employees in January, only five employees in February, and 10 employees again in March, what was the company's employee head count for the quarter? Most companies would use an average balance. In the case of cash, most companies use an ending balance.

5.5.2 Logical data model

Why do we need a logical data model prior to starting a data warehouse implementation?

- **Recognize redundancy**
  - The model explains the specific location of an information element in the Data Warehouse
  - By tracking the usage — denormalization techniques can be used to eliminate frequently needed joins
- **Alterations in the future**
  - The visualization of the structure offers the ability to plug in new information elements in the "right" place and helps on impact analysis of an alteration
- **Completeness of scope**
  - The development group needs a guideline throughout the whole lifecycle of the data warehouse — the logical data model is the build-plan to implement a Data Warehouse and detects antonyms and synonyms ("Account_Name" = "Customer_Name" ?)

The underlying model of information in a data warehouse is designed in a star-join schema (see Figure 30). It is comprised of two different types of tables with different characteristics:

- Fact tables — “what are we measuring?”
  - Contain numerical values to measure the companies performance
  - Short record structure — many records in the table
- Dimension tables
  - Contain descriptive information about the numerical values in the fact table
  - Long, denormalized record structure — less records than in the fact table

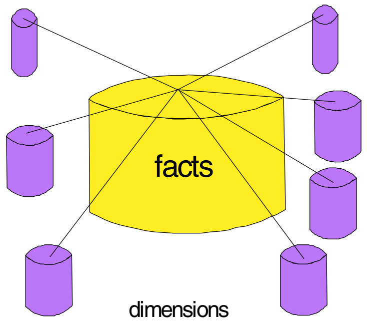
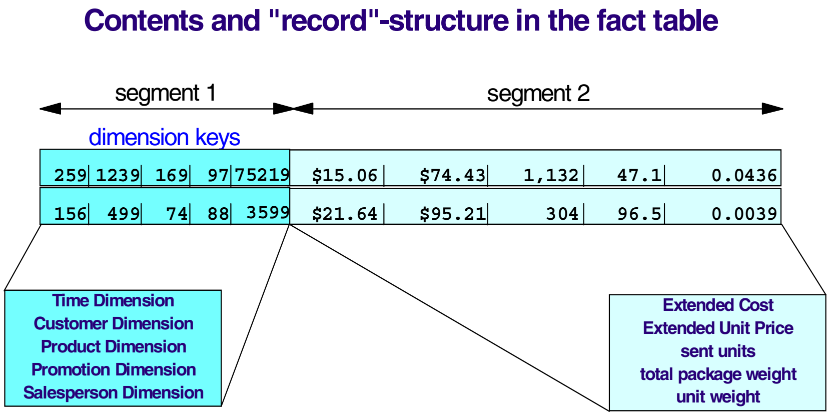


Figure 30. Star-Join Schema

5.5.2.1 Fact table contents

Information that goes into the fact table (shown in Figure 31 on page 100) has to have some characteristics:

- **Numerical**
  - The normal query in a Data Warehouse aggregates thousands of records — therefore the values have to be numerical to generate averages or summaries
- **Additive values**
  - To summarize a high number of values, they have to be additive or at least semi-additive to generate useful information without creating misleading information
- **Continuously valued**
  - To evaluate and watch changes in a company over time, the values should be capable to have virtually every value in a broad range of possible values
  - This evaluated information must be suitable to reflect the companies performance over time



259	1239	169	97	75219	$15.06	$74.43	1,132	47.1	0.0436
156	499	74	88	3599	$21.64	$95.21	304	96.5	0.0039

Figure 31. Fact table — structure

- Decide precisely what a fact record represents (for example, line item of an invoice)
- All facts in the record have to refer directly to the dimensions in segment 1 of the record (no weekly sales, ...)

Figure 32 shows a “good” and a “bad” fact table design.

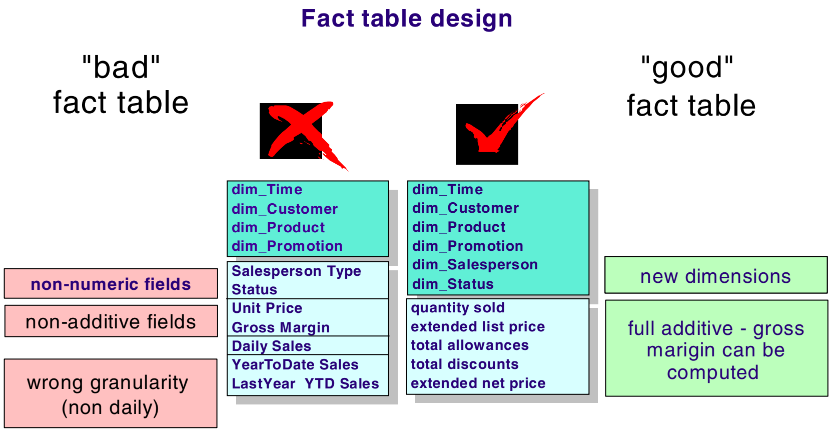


Figure 32. Good and bad fact table

This figure shows two fact table designs — the left fact table contains data that does not follow the basic rules for fact table design. The data elements in this table contain values that are:

- Not numeric — no way to summarize these fields
- Not additive because discounts, rebates, and so on, are hidden in the unit price
- Not directly related to the given key structure — this means not additive

The non-numeric values have to be converted into dimensions if necessary — and be split into their elementary parts to calculate any required value from the elementary values.

Data with wrong granularity has to be removed from the table and stored in specialized data marts with the required granularity. Mixed granularity in the fact table is the first step to misinterpretation.

The fact table on the right is more accurate. Each key represents one dimension to be used to group the data. Each value in the second segment of the record is directly related to the key combination

Figure 33 shows another example of a Star-Join schema:

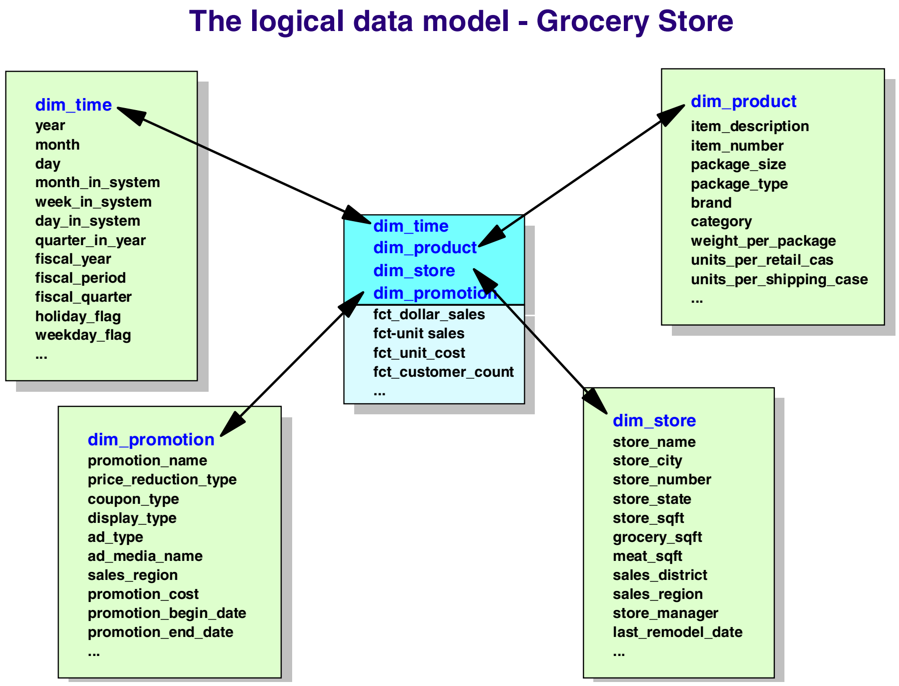
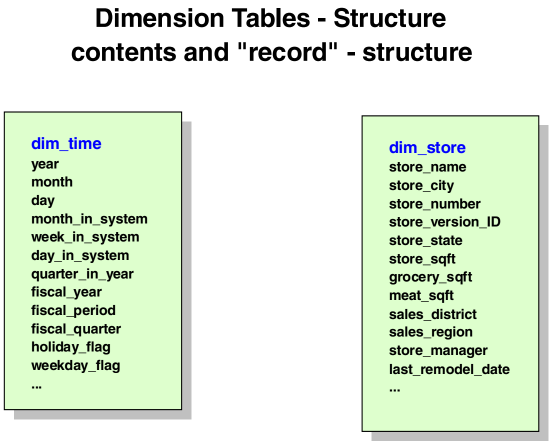


Figure 33. Star join example

5.5.2.2 Dimension table contents

Dimension table information has to be as descriptive as possible (see Figure 34 on page 102). As such it has to contain:

- **Descriptive attributes**
  - To identify the "circumstances" that generated a record in the fact table
- **Denormalized record structure**
  - Avoid multiple joins to find the description for a given dimension attribute (the "customer group" in the "customer"-dimension has to be a text field and not a numerical value referencing another record in another table,...)
- **Textual descriptions**
  - The attribute contents in a dimension table are used as row and column headings in the presentation layer of a data warehouse — therefore textual descriptions help to simplify report generation.



Dimension Tables - Structure
contents and "record" - structure

dim_time
year
month
day
month_in_system
week_in_system
day_in_system
quarter_in_year
fiscal_year
fiscal_period
fiscal_quarter
holiday_flag
weekday_flag
...

dim_store
store_name
store_city
store_number
store_version_ID
store_state
store_sqft
grocery_sqft
meat_sqft
sales_district
sales_region
store_manager
last_remodel_date
...

Figure 34. Dimension table

Dimension table content and “record” structure is as follows:

- Dimension tables contain several attributes to **describe the entity in detail**.
- **All attributes** can be used to **create different views** to the information in the fact table.
- **Creativity** in finding specific attributes usually **pays off**!
- Attributes should appear in **text-format** — these values will be used as group-headers in reporting.

- **Avoid shortcuts** or non-numerical values to represent a status or group.
- Values that are **not important to the OLTP** system may have a powerful meaning in a data warehouse.
- The OLTP primary key is **not necessarily the primary key** in the data warehouse.

Ralph Kimball says, for example:

“The nine decision points of a complete database design for a dimensional data warehouse consist of deciding on the following:

1. The processes, and hence the identity, of the fact tables
2. The grain of each fact table
3. The dimensions of each fact table
4. The facts, including precalculated facts
5. The dimension attributes with complete descriptions and proper terminology
6. How to track slowly changing dimensions
7. The aggregations, heterogeneous dimensions, minidimensions, query modes and other physical storage decisions
8. The historical duration of the database
9. The urgency with which the data is extracted and loaded into the data warehouse"

The following will further explain the statements in this quote:

Identify the processes

Before starting with the database design, certain investigations should be performed, such as:

- Identify the data extract opportunities for "fact" information
- Work with existing sources of used data
- Where is the data generated/collected?
- What is the meaning of an existing record in the fact table at the lowest level?

Define the granularity

The meaning of an individual record in the fact table defines the granularity of the fact table. Things to consider are:

- Individual invoice/order line item
- Daily, weekly, monthly snapshot

- Cumulative measurements per group
- Define the multi-part key of each fact table

Define the dimensions

Next the dimensions need to be defined. Dimensions can be described as the driver of the data marts and they contain the enterprise's vocabulary. Usually the contents will be used as column headers in reports.

The targets and measurements for the dimension definition are:

- To find out the basic dimensions — refer to the organizational structure of the company such as brand managers, country managers, channel managers, ...
- Never create measurements without having a target
  - "Nice-to-know" information increases the complexity of the data warehouse architecture over time
  - Focus on information that can generate ROI for the company
  - Data warehouse information is no replacement for reporting
- Never create target numbers without having the reliable source to collect the information
- Choose dimensions with long-range perspective in mind.
- Additional dimensions can be created or added to the model without influencing the granularity of the fact table.
- Shared dimensions (conformed dimensions) have to be exactly the same or at least a subset of each other.

Choosing the facts

Facts must be expressed at the uniform level implied by the granularity of the model. The facts should be:

- Numerical
- "As additive as possible"
- Continuously valued

Additional facts can be added to the fact table as long as they are consistent with the grain of the table.

Storing Pre-calculations

There are some reasons why pre-calculated values should be stored in the fact table, as it will:

- Avoid misinterpretation of given values
- Speed up access to highly requested values
- Cost of a possible wrong interpretation is greater than the cost of additional disk space
- Minimize network traffic

Rounding out the dimension tables

Rounding out the dimension tables means to make sure the following criteria are met:

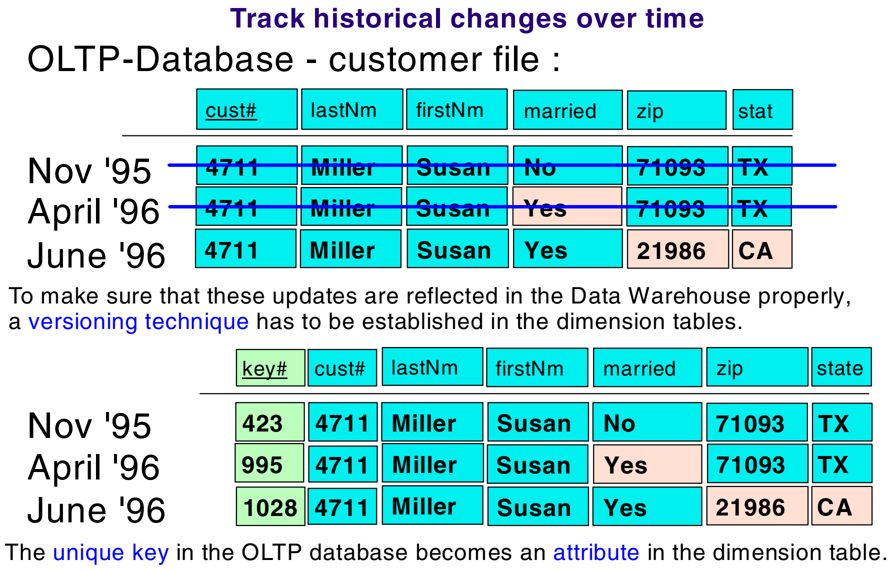
- Add descriptive attributes to the dimension tables in text format
- Attributes must be generated automatically by the extraction process — translation tables
- Only in special situations descriptive attributes can be added manually to dimension tables
- No cryptic abbreviations — column header function

Duration of the data warehouse

One basic question when defining the data warehouse is: How long do we have to keep historical data in the data warehouse for trend analysis or comparisons that influence our future decision making process? There are certain questions that will help to make a decision related to this question, such as:

- What is the expected size of the data warehouse?
- What is the meaning of 5-year old data to our business today?
- Is the same level of granularity needed for X year old data?

Figure 35 shows an example of slowly changing dimensions.



Track historical changes over time

OLTP-Database - customer file :

	cust#	lastNm	firstNm	married	zip	stat
Nov '95	~~4711~~	~~Miller~~	~~Susan~~	~~No~~	~~71093~~	~~TX~~
April '96	~~4711~~	~~Miller~~	~~Susan~~	~~Yes~~	~~71093~~	~~TX~~
June '96	4711	Miller	Susan	Yes	21986	CA

To make sure that these updates are reflected in the Data Warehouse properly, a versioning technique has to be established in the dimension tables.

	key#	cust#	lastNm	firstNm	married	zip	state
Nov '95	423	4711	Miller	Susan	No	71093	TX
April '96	995	4711	Miller	Susan	Yes	71093	TX
June '96	1028	4711	Miller	Susan	Yes	21986	CA

The unique key in the OLTP database becomes an attribute in the dimension table.

Figure 35. Slowly changing dimensions

5.6 Visual Warehouse

Visual Warehouse is an integrated product for building and maintaining a data warehouse or data mart in a LAN environment. Visual Warehouse does not simply create a data warehouse or an informational database; it provides the processes to define, build, manage, monitor, and maintain an informational environment. It integrates many of the business intelligence component functions into a single product. It can be used to automate the process of bringing data together from heterogeneous sources into a central, integrated, informational environment.

Visual Warehouse can be managed either centrally or from the workgroup environment. Therefore, business groups can meet and manage their own information needs without burdening information systems resources, thus enjoying the autonomy of their own data mart without compromising overall data integrity and security in the enterprise.

5.6.1 Data sources supported

Visual Warehouse provides the capability to extract and transform data from a wide range of heterogeneous data sources, either internal or external to the enterprise, such as the DB2 family, Oracle, Sybase, Informix, Microsoft SQL Server, VSAM, IMS, and flat files (for example, from spreadsheets). Data from these sources is extracted and transformed based on metadata defined by the administrative component of Visual Warehouse. The extract process, which supports full refreshes of data, can run on demand or on an automated scheduled basis.

5.6.2 Data stores supported

The transformed data can be placed in a data warehouse built on any of the DB2 UDB platforms, including DB2 for Windows NT, DB2 for AIX, DB2 for HP-UX, DB2 for Sun Solaris, DB2 for SCO, DB2 for SINIX, DB2 for OS/2, DB2 for OS/400, and DB2 for OS/390, or on flat files. Visual Warehouse provides the flexibility and scalability to populate any combination of the supported databases.

Visual Warehouse also supports Oracle, Sybase, Informix, and Microsoft SQL Server using IBM DataJoiner.

5.6.3 End user query tools

Once the data is in the target data warehouse, it is accessible by a variety of end user query tools. Those tools can be from IBM, such as Lotus Approach, or QMF for Windows, or from any other vendors whose products comply with the DB2 Client Application Enabler (CAE) or the Open Database Connectivity (ODBC) interface, such as Business Objects, Cognos Impromptu, and Brio Query. The data can also be accessed using a popular Web browser with additional Web infrastructure components.

5.6.4 The architecture of Visual Warehouse

The Visual Warehouse architecture shown in Figure 36 provides a fully distributed Client/Server system that lets users reap the benefits of network computing. The architecture consists of the following major components:

- Server
- Administrative Clients
- Agents
- Control Database
- Target Databases

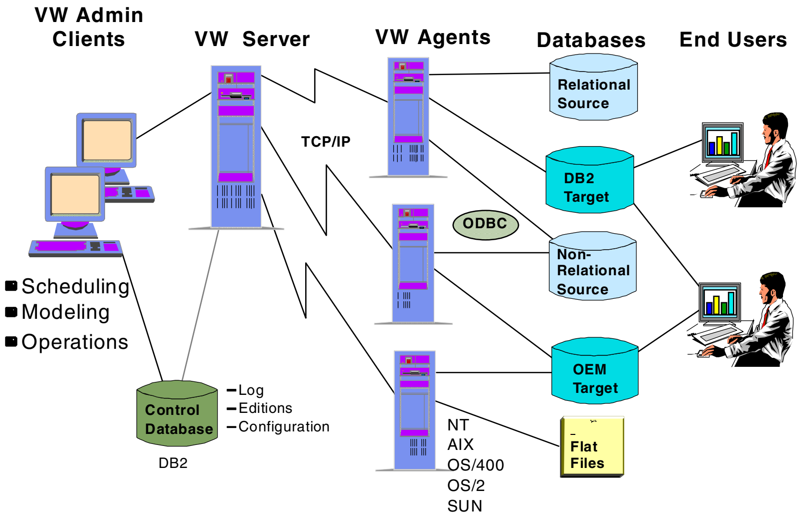


Figure 36. Visual Warehouse architecture

5.6.4.1 Visual Warehouse Server

Visual Warehouse Server, which runs on a Windows NT workstation or server, controls the interaction of the various data warehouse components and provides for automation of data warehousing processes by a powerful scheduling facility, which allows calendar-based scheduling as well as event-based scheduling. The server component monitors and manages the data warehousing processes. It also controls the activities performed by the Visual Warehouse agents.

5.6.4.2 Visual Warehouse administrative clients

The Administrative Client, which also runs on a Windows NT workstation or server, provides an interface for administrative functions, such as defining the business views, registering data resources, filtering source data, defining the target data warehouse databases, managing security, determining the data refresh schedules, and monitoring the execution of the data warehouse processes. Visual Warehouse can support an unlimited number of administrative clients and provides comprehensive security facilities to control and manage client access to the administrative functions.

5.6.4.3 Visual Warehouse agents

Visual Warehouse agents handle access to the source data, filtering, transformation, subsetting, and delivery of transformed data to the target warehouse under the direction of the Visual Warehouse Server.

Visual Warehouse agents run on Windows NT, OS/2, AS/400, AIX, and Sun Solaris. Visual Warehouse supports an unlimited number of agents. Because multiple agents can participate in the population of a data warehouse, the throughput can significantly increase when multiple agents act simultaneously. The agents primarily use ODBC drivers as the means of communicating with different data sources and targets.

The Visual Warehouse agents architecture is a key enabler for scalable business intelligence solutions.

5.6.4.4 Visual Warehouse control database

A control database must be set up in DB2 to be used by Visual Warehouse to store control information used by the Visual Warehouse Server. The control database stores all the metadata necessary to build and manage the warehouse. The information in the control database includes the mappings between the source and target data, the schedules for data refresh, the Business Views, and operational logs. The control database is managed by the Visual Warehouse Administrator and used by the Visual Warehouse agents. When a request for service is made to the Visual Warehouse Server, the control information pertinent to that request is retrieved from the control database and sent to the appropriate agent that actually provides the service. Note that different warehouses could use different control databases.

Advanced DB2 features, such as triggers and stored procedures, can be used in conjunction with the Visual Warehouse control data to provide an advanced operating environment. For instance, DB2 triggers can be used to monitor log inserts and to send out alert signals through DB2 stored procedures when a certain event occurs.

5.6.4.5 Visual Warehouse target databases

Target databases in a data warehouse contain the Visual Warehouse data stored in structures defined as *Business Views* (BVs). When Visual Warehouse populates a BV, data is extracted from the source, transformed according to the rules defined in the BV, and then stored in the target database. Multiple databases could be used as target databases for a data warehouse.

5.6.5 Metadata

Metadata is data about data. Examples of metadata include data element descriptions, data type descriptions, attribute/property descriptions, range/domain descriptions, and process/method descriptions. The repository environment encompasses all corporate metadata resources, database catalogs, data dictionaries, and navigation process. Metadata includes things like the name, length, valid values, and description of a data element. Metadata is stored in a data dictionary and repository. It insulates the data warehouse from changes in the schema of operational systems.

Visual Warehouse stores all the metadata in its control database and is integrated with DataGuide, IBM's metadata management tool, which is part of the Visual Warehouse solution. The data warehouse model, which defines the structure and contents of the data warehouse, is stored in the metadata repository. For each data source to be accessed, Visual Warehouse first extracts the metadata that describes the contents of the data source and places it in the metadata repository. This metadata is then used to extract, filter, transform, and map the source data to the data warehouse.

The metadata of Visual Warehouse can then be transferred to the Information Catalog managed by DataGuide. With DataGuide, users can create an Information Catalog, which contains graphical representations of the metadata. DataGuide can be integrated with DB2 CAE entitled decision support tools, which can be used to view the metadata specific to an object of interest in the DataGuide Information Catalog.

5.6.5.1 Technical versus business metadata

Metadata users can be broadly placed into the categories of business users and technical users. Both of these groups contain a wide variety of users of the data warehouse metadata. They all need metadata to identify and effectively use the information in the data warehouse.

Therefore, we can distinguish between two types of metadata that the repository will contain technical and business metadata.

Technical metadata provides the developers and technical users of the decision support system the confidence that the data in the data warehouse is accurate. In addition, technical metadata is absolutely critical for the ongoing maintenance and growth of the warehouse. Without technical metadata, the task of analyzing and implementing changes to a decision support system is significantly more difficult and time consuming.

The business metadata is the link between the data warehouse and the business users. Business metadata provides these users with a road map for access to the data in the data warehouse and its data marts. The business users are primarily executives or business analysts and tend to be less technical; therefore, they need to have the DSS system defined for them in business terms. The business metadata presents in business terms what reports, queries and data are in the data warehouse, location of the data, reliability of the data, context of the data, what transformation rules were applied and from which legacy systems the data was sourced.

Types of metadata sources

There are two broad types of metadata sources — formal and informal. These sources comprise the business and technical metadata for an organization.

Formal metadata sources are those sources of metadata that have been discussed, documented and agreed upon by the decision-makers of the enterprise. Formal metadata is commonly stored in tools or documents that are maintained, distributed and recognized throughout the organization. These formal metadata sources populate both technical and business metadata.

Informal metadata consists of corporate knowledge, policies and guidelines that are not in a standard form. This is the information that people "just know." This type of information is located in the "company consciousness" or it could be on a note on a key employee's desk. It is not formally documented or agreed upon; however, this knowledge is every bit as valuable as that in the formal metadata sources. Often informal metadata provides some of the most valuable information since it tends to be business related. It is important to note that often much of the business metadata is informal. As a result, it is critical that this metadata is captured, documented, formalized and reflected in the data warehouse. By doing this you are taking an informal source of metadata and transforming it into a formal source. Since every organization differs, it is difficult to say where your informal sources of metadata are; however, following is a list of the most common types of informal metadata:

- Data Stewardship
- Business Rules
- Business Definitions
- Competitor Product Lists

5.6.5.2 Metadata for the masses

The new trend is to move reporting and OLAP functions to the Web within an Internet or, more commonly, an intranet structure.

By moving these functions from the desktop to the Web, it changes the architecture from "client centric" to "server centric." This allows the business users to use their familiar Web browsers to access the metadata to navigate them through the warehouse and its data marts. This n-tier architecture provides for thinner clients and logic distributed among multiple communicating servers. A Web client (browser) can access the Web server to go through the CGI gateway or other Internet service to act as a middle tier. It's important to note that the Web-enabled OLAP tools are not quite as mature as their desktop brothers; however, most of the vendors are working at bringing their OLAP tool suites up to speed as quickly as possible.Web access comes in two broad flavors: static and dynamic. Static Web access uses a standard Web browser (Netscape or Microsoft Explorer) to generate static pages of HTML on an event-driven or time-driven basis. This method of access is very reliable, cost effective and easy to implement. Dynamic Web access references the OLAP and ad hoc query forms of business-user access. These access types require longer response times and a more sophisticated toolset which is still maturing in the marketplace.

5.7 Intelligent Miner for Data

Organizations generate and collect large volumes of data which they use in daily operations, for example, billing and inventory. The data necessary for each operation is captured and maintained by the corresponding department. Yet despite this wealth of data, many companies have been unable to fully capitalize on its value because information implicit in the data is not easy to discern. However, to compete effectively today, taking advantage of high-return opportunities in a timely fashion, decision-makers must be able to identify and utilize information hidden in the collected data. For example, after identifying a group of married, two-income, and high net worth customers, a bank account manager sends these customers information about the growth mutual funds offered by the bank, in an attempt to convince them to use the bank's services rather than those of a discount broker.

Data mining is the process of extracting valid, previously unknown, and ultimately comprehensible information from large databases and using it to make crucial business decisions. The extracted information can be used to form a prediction or classification model, identify relations between database records, or provide a summary of the database(s) being mined. Data mining consists of a number of operations each of which is supported by a variety of techniques such as rule induction, neural networks, conceptual clustering, association discovery, and so on. In many real-world domains such as marketing analysis, financial analysis, fraud detection, and so on, information extraction requires the cooperative use of several data mining operations and techniques. In this paper we present a variety of data mining techniques, discuss how they can be used independently and cooperatively to extract high quality information from databases, and present a multi-component data mining framework.

The goal of identifying and utilizing information hidden in data has three requirements:

- First, the captured data must be integrated into organization-wide views, instead of department-specific views, and often supplemented with open source and/or purchased data.
- Second, the information contained in the integrated data must be extracted, or mined.
- Third, the mined information must be organized in ways that enable decision-making.

The organization of the mined information, and the type of information that must be mined are driven by the overall objective of each decision-making operation. For example, by establishing as his objective to identify good prospective customers for mutual funds, the bank's account manager indicates that he wants to segment the database of bank customers into groups of related customers, for example, urban, married, two-income, mid-thirties, low risk, high net worth customers, and so on, and establish the vulnerability of each group with regard to various types of promotional campaigns. Data mining systems satisfy these three requirements. These requirements imply that a data mining system must interact with a data warehouse which organizes an organization's operational data in ways that facilitate analysis, and must interface with decision support systems (DSSs) which are used by decision-makers in their daily activities. While interaction with a data warehouse is not a hard requirement since most data mining systems can also work from data stored in flat files or operational databases, mining the con-tents of a warehouse usually results in higher quality information because of the diverse but complementary types of data warehouses store.

5.7.1 Hypothesis verification and information discovery

Traditionally the goal of identifying and utilizing information hidden in data has been achieved through the coupling of data warehouses with query generators, and data interpretation systems, such as SAS. Under this scheme the decision-maker must hypothesize the existence of information of interest, convert the hypothesis to a query, pose it to the warehouse, and interpret the returned results with respect to the decision being made. For example, the bank account manager must hypothesize that married, two-income, and high net worth customers buy mutual funds, a rather complex and unlikely hypothesis for any bank account manager to make, and after constructing and posing the appropriate query, interpret the results to establish whether this group constitutes a good set of prospective customers for the bank's discount brokerage service. Systems supporting this operation are called verification-driven data mining systems. Such systems suffer from two problems. First, they require the decision-maker to hypothesize the desired information. Second, the quality of the extracted information is based on the user's interpretation of the posed query's results.

Due to the complexity of the stored data, and of the data interrelations, verification-driven data mining is not sufficient for decision-making. It must be complemented with the ability to automatically discover important information hidden in the data and then present it in the appropriate way. The corresponding systems are called discovery-driven data mining systems. For example, a discovery-driven data mining system applied to the bank's customer database may discover many different groups of bank customers with common characteristics, which may be comprised of college students with low balances relying on a monthly check received by their parents, old married couples relying on social security and pension benefits, or so on, in addition to the group of married, two-income, and high net worth customers. By recognizing the account manager's goal, the discovery-driven system not only identifies the latter as the most appropriate group, but furthermore establishes which of the customers in the group will be good candidates for each type of promotional campaign that can be executed by the bank.

In order to expedite the information discovery operation while maintaining the quality of the extracted information and enabling the decision-maker to take advantage of niche, emerging opportunities, the next generation data mining systems will combine verification-driven with discovery-driven data mining operations. Verification-driven data mining will allow the decision-maker to express and verify organizational and personal domain knowledge and hypotheses, while discovery-driven data mining will be used to refine these hypotheses, as well as identify information not previously hypothesized by the user.

The following presents some of the practical applications of data mining, where each application uses one or more data mining operations. We first provide an overview of the types of operations you will encounter, and the kind of information that each provides. Figure 37 on page 115 shows some examples of applications, operations, and techniques used in data mining, with some of the relationships between them. This figure does not pretend to be complete, but gives an idea of the three levels that you encounter. *Applications* are seen at the business level, where decisions are made. *Operations* are handled by a data mining expert at the information level. He then uses one or several data mining tools that provide the *techniques* to operate on the data, as shown at the bottom of Figure 37 on page 115.

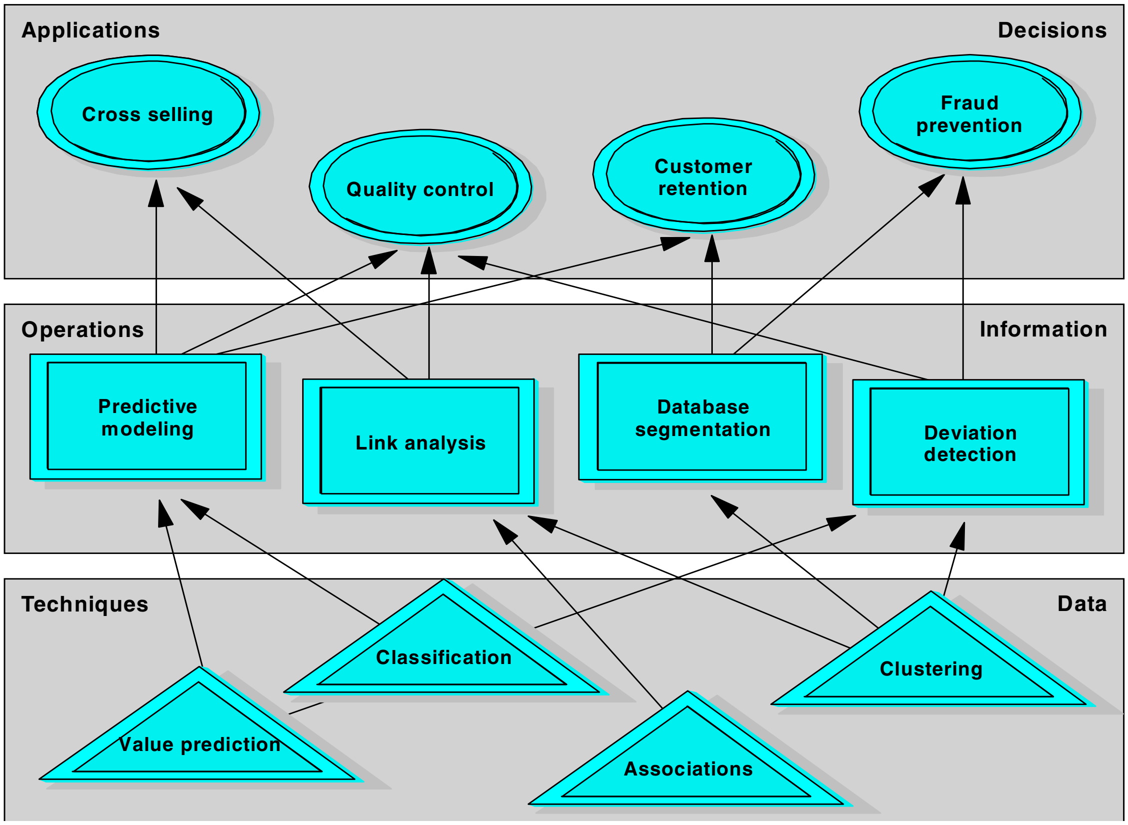


Figure 37. Applications, operations and techniques

The actual application of data mining in your environment depends on your business and partly on your imagination and that of the mining expert. Table 1 on page 116 provides an overview of data mining applications that have been used so far. We have distributed a number of examples over three main categories.

Table 1. Data mining application areas

Market Management	Risk Management	Process Management
Target Marketing	Forecasting	Inventory optimization
Relationship Management	Customer Retention	Quality control
Channel Management	Churn or Attrition Analysis	Demand Forecasting
Market Basket Optimization	Underwriting	Business Scorecards
Cross Selling	Competitive Analysis	
Market Segmentation	Healthcare Fraud	
Web Usage Analysis		

5.7.2 The data mining process

Transforming the contents of a data warehouse into the information that can drive decision-making is a complex process that can be organized into four major steps:

- Data Selection
- Data Transformation
- Data Mining
- Result Interpretation

5.7.2.1 Data selection

A data warehouse contains a variety of diverse data not all of which will be necessary to achieve a data mining goal. The first step in the data mining process is to select the types of data that will be used. For example, marketing data-bases contain data describing customer purchases, demographic data, lifestyle data, census and state financial data, and so on. To identify how to lay out the shelves of a department store, a marketing executive will only need to combine customer purchase data with demographic data. The selected data types may be organized along multiple tables. As part of the data selection step, table joins may need to be performed using products such as IBMs DataJoiner. Furthermore, even after selecting the desired database tables, it is not always necessary to mine the contents of the entire table to identify useful information. Under certain conditions and for certain types of data mining operations, for example, when creating a classification or prediction model, it may be adequate to first sample the table and then mine the sample; usually a less expensive operation.

5.7.2.2 Data transformation

Once the desired database tables have been selected and the data to be mined has been identified, it is usually necessary to perform certain transformations on the data. The type of the transformations is dictated by the type of data mining operation performed and the data mining technique used. Transformations vary from conversions of one type of data to another, for example, converting nominal values into numeric ones so that they can be processed by a neural network, to definition of new attributes, that is, derived attributes. New attributes are defined either by applying mathematical or logical operators on the values of one or more database attributes. For example, taking the natural logarithm of an attribute's values, or establishing the ratio of two attributes. Products such as IBM's Query Management Facility (QMF) or Intelligent Decision Server can be used to transform selected data.

5.7.2.3 Data mining

The transformed data is subsequently mined using one or more techniques in order to try extracting the desired type of information. For example, to develop an accurate, symbolic classification model that predicts whether a magazine subscriber will renew his subscription, one has to first use clustering to segment the subscribers' database, and then apply rule induction to automatically create a classification model for each desired cluster. While mining a particular data set, it may be necessary to access additional data from the warehouse, and/or perform further transformations on the originally selected data.

5.7.2.4 Result interpretation

The extracted information is then analyzed with respect to the end user's decision support goal, and the best information is identified and presented to the decision-maker through the decision support system. Therefore, the purpose of result interpretation is not only to visualize (graphically or logically) the output of the data mining operation, but also to filter the information that will be presented to the decision-maker through the decision support system. For example, if the data mining goal is to develop a classification model, during the result interpretation step the robustness of the extracted model is tested using one of the established test methods, such as cross validation. If the interpreted results are not satisfactory, it may be necessary to repeat the data mining step, or to iterate through the other steps. This is one of the reasons that the information extracted through data mining must be ultimately comprehensible. While performing a particular operation, one often finds that it is necessary to revise data mining operations performed earlier. For example, after displaying the results of a transformation, it may be necessary to select additional data in which case the data selection step is repeated. The data mining process with the appropriate feedback steps between the various data mining operations is shown in Figure 38.

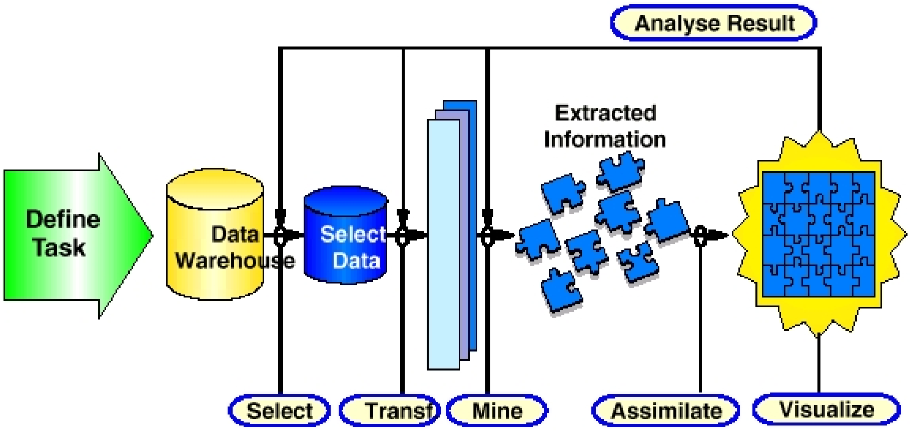


Figure 38. The data mining process

5.7.3 Data mining operations

Four operations are associated with discovery-driven data mining:

5.7.3.1 Predictive model creation

This is the most commonly used operation primarily because of the proliferation of automatic model-development techniques. The goal of this operation is to use the contents of the database, which reflects historical data, that is, data about the past, to automatically generate a model that can predict a future behavior. For example, a financial analyst may be interested in predicting the return of investment of a particular asset so that he can determine whether to include it in a port-folio he is creating. A marketing executive may be interested to predict whether a particular consumer will switch brands of a product of interest. Model creation has been traditionally pursued using statistical techniques. The value added by data mining techniques in this operation is in their ability to generate models that are comprehensible, and explainable, since many data mining modeling techniques express models as sets of if... then... rules.

5.7.3.2 Link analysis

Whereas the goal of the modeling operation is to create a generalized description that characterizes the contents of a database, the goal of link analysis is to establish relations between the records in a database. For example, a merchandising executive is usually interested in determining what items sell together. That is, men's shirts sell together with ties and men's fragrances, so that he can decide what items to buy for the store. He might want to buy ties and fragrances, and he would have to decide how to lay these items out. That is, ties and fragrances must be displayed near the men's shirts section of the store. Link analysis is a relatively new operation, whose large scale application and automation have only become possible through recently developed data mining techniques.

5.7.3.3 Database segmentation

The goal of database segmentation is to partition a database into segments of similar records, that is, records that share a number of properties and so are considered to be homogeneous. In some literature the words *segmentation* and *clustering* are used interchangeably.

As databases grow and are populated with diverse types of data it is often necessary to partition them into collections of related records either as a means of obtaining a summary of each database, or before performing a data mining operation such as model creation, or link analysis. For example, assume a department store maintains a database in which each record describes the items purchased by a customer during a particular visit to the store. The database can then be segmented based on the records that describe sales during the "back to school" period, records that describe sales during the "after Christmas sale" period, and so on. Link analysis can then be performed on the records in the "back to school" segment to identify what items are being bought together. Deviation detection. This operation is the exact opposite of database segmentation. In particular, its goal is to identify outlying points in a particular data set, and explain whether they are due to noise or other impurities being present in the data, or due to causal reasons. It is usually applied in conjunction with database segmentation. It is usually the source of true discovery since outliers express deviation from some previously known expectation and norm. Deviation detection is also a new operation, whose importance is now being recognized and the first algorithms automating it are beginning to appear.

5.7.3.4 Deviation detection

Deviation detection is a operation whose importance is just being recognized. The first data mining algorithms automating the operation are only now beginning to appear. Interestingly, it is often the source of true discovery because outliers express deviation from some previously known expectation and norm.

Today, analysts perform deviation detection, using statistics and visualization techniques or as a by-product of data mining. Linear regression facilitates the identification of outliers in data. Modern visualization techniques available on high-powered computers enable the summarization and graphical representations that make deviations easy to detect. Some data mining operations will tend to show up deviations as a useful by-product of their main analysis. For example, if a database segmentation produces a cluster with only a few records, that cluster is quite likely to hold outliers and therefore requires further investigation.

The business applications that deviation detection supports include fraud detection in the use of credit cards, insurance claims, and telephone cards, quality control, and defects training.

5.7.4 Data mining techniques

While there are only four basic data mining operations, there exist numerous data mining techniques supporting these operations. Predictive model creation is supported by supervised induction techniques, link analysis is supported by association discovery and sequence discovery techniques, database segmentation is supported by clustering techniques, and deviation detection is supported by statistical techniques. To these techniques one has to add various forms of visualization, which even though does not automatically extract information, it facilitates the user in identifying patterns hidden in data, as well as in better comprehending the information extracted by the other techniques.

5.7.4.1 Supervised induction

Supervised induction refers to the process of automatically creating a classification model from a set of records (examples), called the training set. The training set may either be a sample of the database or warehouse being mined, the entire database, or a data warehouse. The records in the training set must belong to a small set of classes that have been predefined by the analyst. The induced model consists of patterns, essentially generalizations over the records, that are useful for distinguishing the classes. Once a model is induced it can be used to automatically predict the class of other unclassified records. Supervised induction methods can be either neural or symbolic. Neural methods, such as backpropagation, represent the model as an architecture of nodes and weighted links. IBM's Neural Network Utility includes a variety of neural supervised induction methods, such as backpropagation, radial basis functions, and so on. Symbolic methods create models that are represented either as decision trees, or as "if ... then ..." rules. Decision trees are generated using algorithms such as id3, and cart.

Rules are generated by algorithms such as IBM's RMINI, the public domain algorithm foil.

For example, credit card analysis is an application for which a supervised induction is well suited. A credit card issuing company may have records about its customers, each record containing a number of descriptors, or attributes. For those customers for which their credit history is known, the customer record may be labeled with a good, medium or poor labels, meaning that the customer has been placed in the corresponding class of good (medium or poor) credit risk. A supervised induction technique producing symbolic classification models may generate the rule stating If the customer's income is over 25,000, and the age bracket is between 45 and 55, and the customer lives in XYZ neighborhood then the customer is good. A supervised induction technique is particularly suitable for data mining if it has three characteristics:

1. It can produce high quality models even when the data in the training set is noisy and incomplete.
2. The resulting models are comprehensible and explainable so that the user can understand how decision are made by the system.
3. It can accept domain knowledge. Such knowledge can expedite the induction task while simultaneously improving the quality of the induced model.

Supervised induction techniques offer several advantages over statistical model-creation methods. In particular, the induced patterns can be based upon local phenomena while many statistical measures check only for conditions that hold across an entire population with well understood distribution. For example, an analyst might want to know if one attribute is useful for predicting another in a population of 10,000 records.

If, in general, the attribute is not predictive, but for a certain range of 100 values it is very predictive, a statistical correlation test will almost certainly indicate that the attributes are completely independent because the subset of the data that is predictive is such a small percentage of the entire population.

5.7.4.2 Association discovery

Given a collection of items and a set of records, each of which contain some number of items from the given collection, an association discovery function is an operation against this set of records which return affinities that exist among the collection of items. These affinities can be expressed by rules such as "72% of all the records that contain items A, B and C also contain items D and E." The specific percentage of occurrences (in this case 72) is called the confidence factor of the association. Also, in this association, A, B and C are said to be on an opposite side of the association to D and E. Association discovery can involve any number of items on either side of the association.

A typical application that can be built using association discovery is Market Basket Analysis. In this application, a retailer will run an association discovery function over the point of sales transaction log. The transaction log contains, among other information, transaction identifiers and product identifiers. The collection of items mentioned above is, in this example, the set of all product descriptors, or SKU's. Typically, this set is of the order of 100,000 or more items. The set of products identifiers listed under the same transaction identifier constitutes a record, as defined above. The output of the association discovery function is, in this case, a list of product affinities. Thus, through association discovery the market basket analysis application can determine affinities such as "20% of the time that a specific brand toaster is sold, customers also buy a set of kitchen gloves and matching cover sets."

Another example of the use of association discovery is in an application that analyzes the claim forms submitted by patients to a medical insurance company. Every claim form contains a set of medical procedures that were performed to the given patient during one visit. By defining the set of items to be the collection of all medical procedures that can be performed on a patient and the records to correspond to each claim form, the application can find, using the association discovery function, relationships among medical procedures that are often performed together.

5.7.4.3 Sequence discovery

In the transaction log discussed above, the identity of the customer that did the purchase is not generally known. If this information exists, an analysis can be made of the collection of related records of the same structure as above (that is, consisting of a number of items drawn from a given collection of items). The records are related by the identity of the customer that did the repeated purchases.

Such a situation is typical of a Direct Mail application. In this case, a catalog merchant has the information, for each customer, of the sets of products that the customer buys in every purchase order. A sequence discovery function will analyze such collections of related records and will detect frequently occurring patterns of products bought over time. A sequence discovery function could also have been used in one of the examples in the previous section to discover the set of purchases that frequently precede the purchase of a microwave oven. Another example of the use of this function could be in the discovery of a rule that states that 68% of the time when Stock X increased its value by at most 10% over a 5-day trading period and Stock Y increased its value between 10% and 20% during the same period, then the value of Stock Z also increased in a subsequent week.

Sequence discovery can be used to detect the set of customers associated with frequent buying patterns. Use of sequence discovery on the set of insurance claims can lead to the identification of frequently occurring medical procedures performed on patients, which in turn can be used to detect cases of medical fraud.

5.7.4.4 Conceptual clustering

Clustering is used to segment a database into subsets, the clusters, with the members of each cluster sharing a number of interesting properties. The results of a clustering operation are used in one of two ways. First, for summarizing the contents of the target database by considering the characteristics of each created cluster rather than those of each record in the database. Second, as an input to other methods, for example, supervised induction. A cluster is a smaller and more manageable data set to the supervised inductive learning component.

Clusters can be created either statistically, or using neural and symbolic unsupervised induction methods. The various neural and symbolic methods are distinguished by:

1. The type of attribute values they allow the records in the target database to take, that is, numeric, nominal, structured objects
2. The way they represent each cluster
3. The way they organize the set of clusters, that is, hierarchically or into flat lists. Once the database has been clustered, the analyst can examine the created clusters to establish the ones that are useful or interesting using a visualization component.

Statistical methods represent a cluster as a collection of instances. It is difficult to decide how to assign a new example to existing clusters since one must define a way for measuring the distance between a new instance and the instances already in the cluster. It is also difficult to predict the attributes of members of a cluster. One must identify the attribute's value by applying a statistical procedure to the entire data set.

Neural clustering methods such as those included in IBM's Neural Network Utility, for example, feature maps, represent a cluster as a prototype with which they associate a subset of the instances in the data set being clustered. Symbolic clustering methods, for example, AQ11, UNIMEM, COBWEB, operate primarily on instances with nominal values. They consider all the attributes that characterize each instance and use artificial intelligence-based search methods to establish the subset of these attributes that will describe each created cluster.

Clustering differs from other data mining techniques in that its objective is generally far less precise. Such techniques are sensitive to redundant and irrelevant features. This problem can be alleviated by permitting the user to direct the clustering component to ignore a subset of the attributes that describe each instance, or by allowing the analyst to assign a weight factor to each attribute;

increasing the weight of an attribute increases the likelihood that the algorithm will cluster according to that attribute. The importance of attributes, especially numeric-valued attributes, can be established using univariate and bivariate statistical methods.

5.7.5 Visualization

Visualization provides analysts with visual summaries of data from a database. It can also be used as a method for understanding the information extracted using other data mining methods. Features that are difficult to detect by scanning rows and columns of numbers in databases, often become obvious when viewed graphically. Data mining necessitates the use of interactive visualization techniques that allow the user to quickly and easily change the type of information displayed, as well as the particular visualization method used (for example, change from a histogram display to a scatter plot display, or to Parallel Coordinates). Visualizations are particularly useful for noticing phenomena that hold for a relatively small subset of the data, and thus are "drowned out" by the rest of the data when statistical tests are used since these tests generally check for global features.

The advantage of using visualization is that the analyst does not have to know what type of phenomenon he is looking for in order to notice something unusual or interesting. For example, with statistical tests the analyst must ask rather specific questions, such as "does the data fit this condition?" Often, the analyst wants to discover something unusual or interesting about a set of instances or an attribute. However, he must ask very directed questions, such as "is the distribution skewed?" or "is this set of values consistent with the Poisson distribution?" No general statistical test can answer the question "is there anything unusual about this set of instances?"; there are only tests for determining if the data is unusual in a particular way. Visualization compensates for this; humans tend to notice phenomena in visually displayed data precisely because they are unusual.

Depending on the skill level of the end user that needs to analyze or interpret a data mining result, the final visualization method needs to be implemented to take this in account. Figure 39 shows an example of a clustering process. It is pretty obvious, that a result like this needs to be presented to a very skilled analyst for further interpretation.

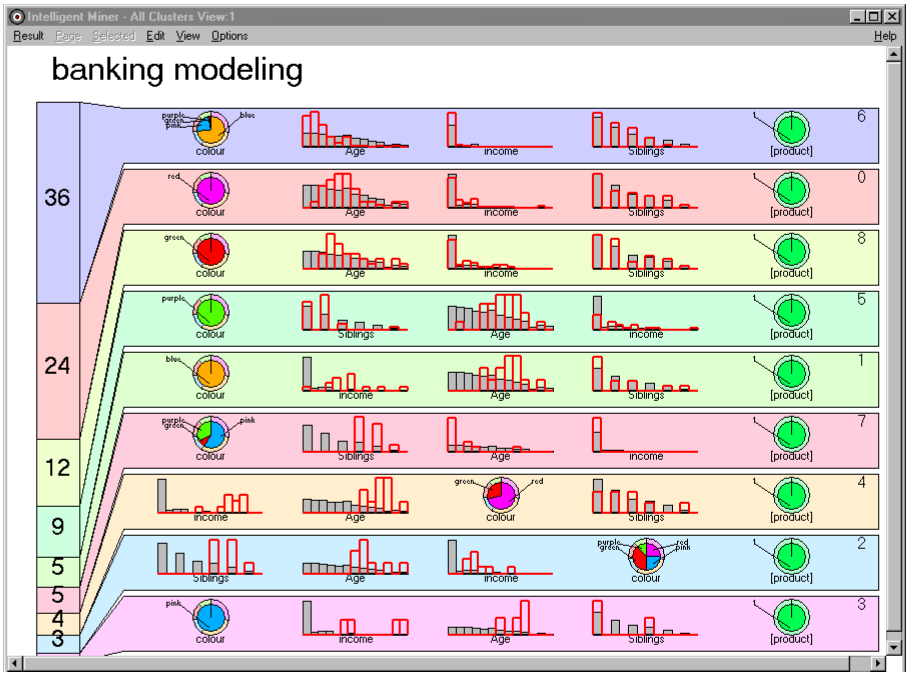
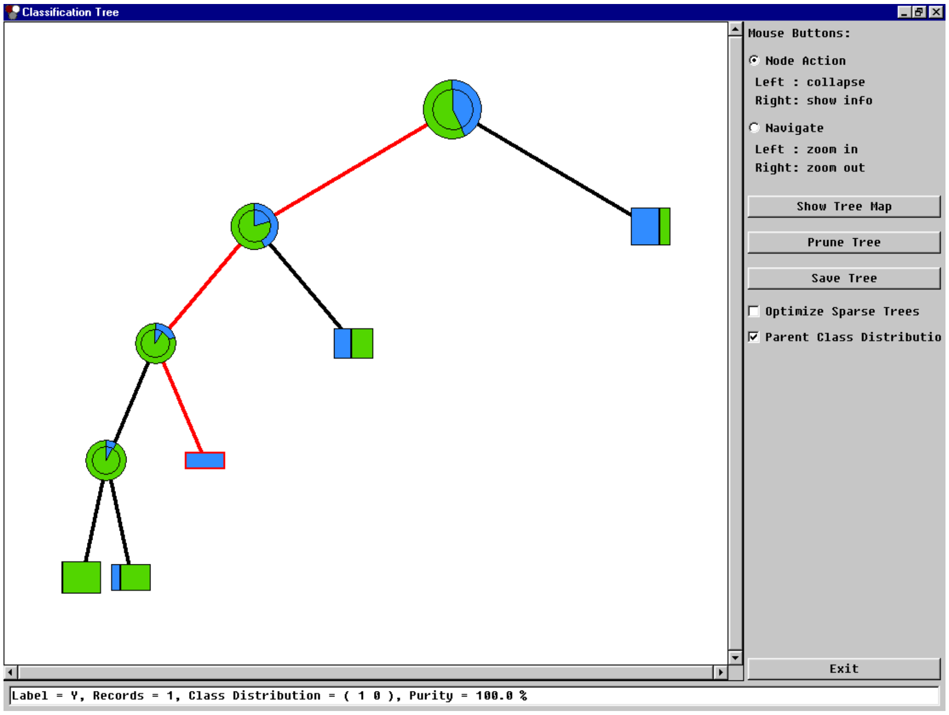


Figure 39. Cluster visualization

The following figure (Figure 40) shows the result of a decision tree. Compared to the clustering result, it is more easy to understand, but still too complex for a quick decision made by an end user.



Mouse Buttons:

● Node Action
Left : collapse
Right: show info

○ Navigate
Left : zoom in
Right: zoom out

□ Optimize Sparse Trees

☑ Parent Class Distributio

```
Label = Y, Records = 1, Class Distribution = ( 1 0 ), Purity = 100.0 %
```

Figure 40. Decision tree

If, for example, the result of a data mining process will be used to make quick decisions, other visualization techniques should be applied. To make a decision based on certain thresholds, a traffic light is a simple to use technique. A green traffic light is used to show that the value is in range, yellow indicates that the value needs to be given some attention, whereas red means that an action needs to be taken.

5.7.6 Data mining applications

Data mining is now being applied in a variety of domains ranging from investment management, to astronomy. Its importance and application potential has been particularly recognized in retail and marketing, banking, insurance, health care, and telecommunications for applications such as market basket analysis for promotion effectiveness, customer vulnerability analysis, customer relationship management for cross-selling, portfolio creation, cellular telephony fraud detection, and so on. In each of these applications, it is usually necessary to perform several data mining operations, in addition to the data warehousing and decision support operations. Furthermore, in certain cases it may be necessary to use several techniques to perform one particular operation in order to fully take advantage of the data characteristics. We can now describe a particular data mining application (customer vulnerability analysis) and detail the operations it supports.

Consumer vulnerability analysis refers to the process of mining various types of consumer data to extract models, called vulnerability models, that predict consumer loyalty levels to a particular brand of product, such as orange juice, or class of products, such as frozen fruit juices. Based on the model's predictions, companies determine how to present products to consumers in stores, and identify the consumers who will be targeted by each of the marketing strategies they employ.

Companies such as packaged goods companies and household electronics companies aim at gaining a share of the market for each new product they introduce, and increasing, or at least maintaining, the market share of each of their existing products. To achieve these two goals, companies must create marketing strategies, and identify the consumers to whom each strategy should be directed. Marketing strategies include media advertising, direct mail campaigns in which consumers may receive free product samples or discount coupons for a particular product, in-store promotions in which a particular product is promoted in a particular store and so forth. For example, a marketing strategy of a frozen fruit juice company aimed at maintaining market share of 12-ounce cans of frozen orange juice in the Midwest during the winter months may involve the mailing of discount coupons to 20% of the households with at least two children during each of these months. The corresponding strategy for urban households who live in the northeastern region of the United States may call for weekend in-store promotion of the same product. The question that a marketing executive has to answer in this case is: which households should be included in the direct mail campaign for the effort to be successful?

The target audience for each marketing strategy is identified through prediction models that are developed by analyzing various types of data, such as point of sale data collected at the supermarket checkout counter, consumer purchase history, demographics, and surveys, obtained from a monitored and well-understood group of consumers. For example, the A.C. Nielsen company records all the supermarket purchases of 15,000 households. Marketing strategies are aimed at consumers who are perceived as vulnerable, that is, they have low brand loyalty for a particular product and are thus likely to switch to a different brand. Such consumers are of interest for two reasons. First, the company attempts to attract as many vulnerable customers of its rivals as it can, thus increasing its market share for a particular product. Second, each company tries to dissuade its vulnerable customers from switching to rival products, thus maintaining its market share for the product. The goal of consumer vulnerability analysis is to develop models that can identify the particular set of consumers who should be targeted by a particular marketing strategy. Figure 41 describes an example of customer vulnerability analysis.

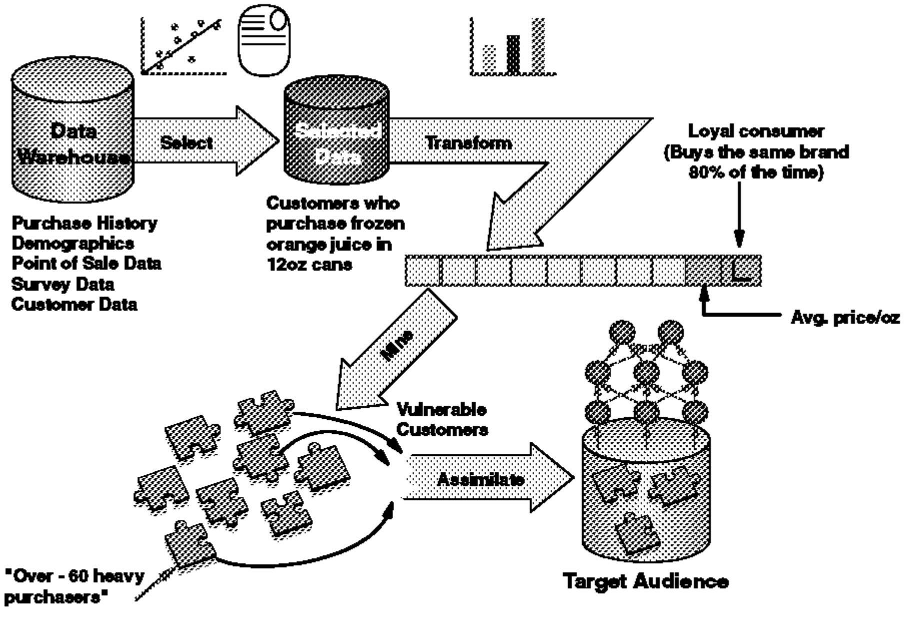


Figure 41. Model development — consumer vulnerability analysis

Vulnerability prediction models are developed using the following five-step process.

1. Identify the types of data that will be used in the analysis, as well as the actual database records of interest. For example, the analyst of a frozen fruit juices company who is in the process of developing vulnerability models for consumers who purchase frozen orange juice in 12-ounce cans, may consider purchase history and demographic data alone, and select all the data-base records that include the purchase of at least one 12-ounce frozen orange juice can.

2. Define the concept of "loyal consumer,'' or conversely the concept of "vulnerable consumer,'' which is to be predicted given other data on a consumer. For example, an analyst may provide the following definition for a brand-loyal frozen orange juice consumer: "a loyal consumer of frozen orange juice is one who buys the same brand more than 80% of the time." The data selected during the first stage is then pre-processed using these definitions and each record is labeled appropriately as "loyal" or "vulnerable." During this step, other derived features which might potentially be useful in characterizing a consumer (such as "average price paid per ounce") are added as new fields to the database. Their presence contributes to the quality and comprehension of the resulting models.
3. Using data mining clustering methods identify statistically important subsets of the data. For example, "heavy frozen orange juice purchasers over the age of 60 living in northeastern states." These subsets may include both loyal and vulnerable consumers. Different segments of the market may have different properties which call for different kinds of marketing strategies.
4. Perform a model creation operation to develop a prediction model for either the database identified during the first step or each subset identified during the third step. For example, a rule discovered by applying supervised rule induction on the "over-60 heavy purchasers" subset may reveal that "if the consumer purchases three of the top five frozen orange juice brands during a one-year period, then the consumer is vulnerable." If the analyst decides not to perform the third step in this process, and the database is too large, he may sample it and create the prediction model from the resulting sample. In this way, the analyst obtains an average case prediction accuracy for the discovery technique used and the data being mined.
5. Apply the model to the database of consumers who will be targeted by the marketing strategy and use it to classify each of the records (consumers) included in this database. The records that are classified by the model as "vulnerable" comprise the target audience of the marketing strategy.

Traditionally, the third and fourth steps of this process have been performed using database queries, visualization, and linear regression. Manually creating effective database queries is a time-consuming process. Furthermore, important subsets of the data are often overlooked. Using data mining clustering techniques this step can be automated and improve the quality of the identified data subsets. While frequently accurate, linear regression models are hard to interpret, and cannot provide explanations of the predictions they make. One can use instead supervised induction techniques, such as neural networks, and rule- and tree-induction.

Appendix A. Special notices

This publication is intended to help IT specialists and consultants to get prepared for the IBM professional certification to become an IBM Certified Solutions Expert - Business Intelligence. The information in this publication is not intended as the specification of any programming interfaces that are provided by the IBM products mentioned in this book. See the PUBLICATIONS section of the IBM Programming Announcement for the IBM products for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
APPN
AS/400
AT
CT
DATABASE 2
DataGuide
DataJoiner
DataPropagator
DB2
DRDA
IBM ®
IMS
Intelligent Miner
Net.Data
Netfinity
NetView
OS/2
OS/390
OS/400
QMF
RACF
RS/6000
S/390
Service Director
SP
System/390
Visual Warehouse
XT
400

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 International Technical Support Organization publications

For information on ordering these ITSO publications see “How to get ITSO redbooks” on page 137.

- *From Multiplatform Operational Data To Data Warehousing and Business Intelligence*, SG24-5174
- *Getting Started with Data Warehouse and Business Intelligence*, SG24-5415
- *My Mother Thinks I’m a DBA! Cross-Platform, Multi-Vendor, Distributed Relational Data Replication with IBM DB2 Data Propagator and IBM DataJoiner Made Easy!*, SG24-5463
- *Managing Multidimensional Data Marts with Visual Warehouse and DB2 OLAP Server*, SG24-5270
- *Intelligent Miner for Data Applications Guide*, SG24-5252
- *Mining Relational and Nonrelational Data with IBM Intelligent Miner for Data Using Oracle, SPSS, and SAS As Sample Data Sources*, SG24-5278
- *Intelligent Miner for Data - Enhance Your Business Intelligence*, SG24-5422

B.2 Redbooks on CD-ROMs

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at http://www.redbooks.ibm.com/ for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	**Collection Kit Number**
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

B.3 Other publications

These publications are also relevant as further information sources:

- *Discovering Data Mining*, ISBN 0-13-743980-6
- *Data Mining,* ISBN 0-201-40380-3
- *Data Warehousing*, ISBN 0-07-041034-8
- *Business Intelligence: The IBM Solution Datawarehousing and OLAP*, ISBN 1-85233-085-6
- *DB2 Universal Database Certification Guide*, ISBN 0-13-079661-1

How to get ITSO redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download, or order hardcopy/CD-ROM redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbook fax order form

Please send me the following:

Title **Order Number** **Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

□ Invoice to customer number

□ Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

A

aggregate. A group of data cells arranged by the dimensions of the data. For example, a spreadsheet exemplifies a two-dimensional array with the data cells arranged in rows and columns, each being a dimension. A three-dimensional array can be visualized as a cube with each dimension forming a side of the cube, including any slice parallel with that side. Higher dimensional arrays have no physical metaphor, but they organize the data in the way users think of their enterprise. Typical enterprise dimensions are time, measures, products, geographical regions, sales channels, etc.

application programming interface (API). A functional interface supplied by the operating system or a separate orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

architecture. The number of processing units in the input, output, and hidden layer of a neural network. The number of units in the input and output layers is calculated from the mining data and input parameters. An intelligent data mining agent calculates the number of hidden layers and the number of processing units in those hidden layers.

attribute. Characteristics or properties that can be controlled, usually to obtain a required appearance. For example, color is an attribute of a line. In object-oriented programming, a data element defined within a class.

C

calculated member. A calculated member is a member of a dimension whose value is determined from other members' values (e.g., by application of a mathematical or logical operation). Calculated members may be part of the OLAP server database or may have been specified by the user during an interactive session. A calculated member is any member that is not an input member.

cell. A single datapoint that occurs at the intersection defined by selecting one member from each dimension in a multi-dimensional array. For example, if the dimensions are measures, time, product and geography, then the dimension members: Sales, January 1994, Candy Bars and United States specify a precise intersection along all dimensions that uniquely identifies a single data cell, which contains the value of candy bar sales in the United States for the month of January 1994.

children. Members of a dimension that are included in a calculation to produce a consolidated total for a parent member. Children may themselves be consolidated levels, which requires that they have children. A member may be a child for more than one parent, and a child's multiple parents may not necessarily be at the same hierarchical level, thereby allowing complex, multiple hierarchical aggregations within any dimension.

consolidate. Multi-dimensional databases generally have hierarchies or formula-based relationships of data within each dimension. Consolidation involves computing all of these data relationships for one or more dimensions, for example, adding up all Departments to get Total Division data. While such relationships are normally summations, any type of computational relationship or formula might be defined. Synonyms: Roll-up, Aggregate

D

DATABASE 2 (DB2). An IBM relational database management system.

database table. A table residing in a database.

database view. An alternative representation of data from one or more database tables. A view can include all or some of the columns contained in the database table or tables on which it is defined.

data field. In a database table, the intersection from table description and table column where the corresponding data is entered.

data format. There are different kinds of data formats, for example, database tables, database views, pipes, or flat files.

data table. A data table, regardless of the data format it contains.

data type. There are different kinds of Intelligent Miner data types, for example, discrete numeric, discrete nonnumeric, binary, or continuous.

dense. A multi-dimensional database is dense if a relatively high percentage of the possible combinations of its dimension members contain data values. This is the opposite of sparse.

derived data. Derived data is produced by applying calculations to input data at the time the request for that data is made, i.e., the data has not been pre-computed and stored on the database. The purpose of using derived data is to save storage space and calculation time, particularly for calculated data that may be infrequently called for or that is susceptible to a high degree of interactive personalization by the user. The tradeoff is slower retrievals.

derived members. Derived members are members whose associated data is derived data.

detail member. A detail member of a dimension is the lowest level number in its hierarchy.

dimension. A dimension is a structural attribute of a cube that is a list of members, all of which are of a similar type in the user's perception of the data. For example, all months, quarters, years, etc., make up a time dimension; likewise all cities, regions, countries, etc., make up a geography dimension. A dimension acts as an index for identifying values within a multi-dimensional array. If one member of the dimension is selected, then the remaining dimensions in which a range of members (or all members) are selected defines a sub-cube. If all but two dimensions have a single member selected, the remaining two dimensions define a spreadsheet (or a "slice" or a "page"). If all dimensions have a single member selected, then a single cell is defined. Dimensions offer a very concise, intuitive way of organizing and selecting data for retrieval, exploration and analysis.

drill down/up. Drilling down or up is a specific analytical technique whereby the user navigates among levels of data ranging from the most summarized (up) to the most detailed (down). The drilling paths may be defined by the hierarchies within dimensions or other relationships that may be dynamic within or between dimensions. For example, when viewing sales data for North America, a drill-down operation in the Region dimension would then display Canada, the eastern United States and the Western United States. A further drill- down on Canada might display Toronto, Vancouver, Montreal, etc.

F

field. A set of one or more related data items grouped for processing. In this document, with regard to database tables and views, *field* is synonymous with *column*.

file. A collection of related data that is stored and retrieved by an assigned name.

file name. (1) A name assigned or declared for a file. (2) The name used by a program to identify a file.

flat file. (1) A one-dimensional or two-dimensional array; a list or table of items. (2) A file that has no hierarchical structure.

formatted information. An arrangement of information into discrete units and structures in a manner that facilitates its access and processing. Contrast with *narrative information*.

function. Any instruction or set of related instructions that perform a specific operation.

I

input data. The metadata of the database table, database view, or flat file containing the data you specified to be mined.

instance. In object-oriented programming, a single, actual occurrence of a particular object. Any level of the object class hierarchy can have instances. An instance can be considered in terms of a copy of the object type frame that is filled in with particular information.

M

metadata. In databases, data that describes data objects.

multi dimensional analysis. The objective of multi-dimensional analysis is for end users to gain insight into the meaning contained in databases. The multi-dimensional approach to analysis aligns the data content with the analyst's mental model, hence reducing confusion and lowering the incidence of erroneous interpretations. It also eases navigating the database, screening for a particular subset of data, asking for the data in a particular orientation and defining analytical calculations. Furthermore, because the data is physically stored in a multi-dimensional structure, the speed of these operations is many times faster and more consistent than is possible in other database structures. This combination of simplicity and speed is one of the key benefits of multi-dimensional analysis.

O

output data. The metadata of the database table, database view, or flat file containing the data being produced or to be produced by a function.

OLAP. On-Line Analytical Processing (OLAP) is a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user.

P

pass. One cycle of processing a body of data.

processing unit. A processing unit in a neural network is used to calculate an output by summing all incoming values multiplied by their respective adaptive connection weights.

R

reach through. Reach through is a means of extending the data accessible to the end user beyond that which is stored in the OLAP server. A reach through is performed when the OLAP server recognizes that it needs additional data and automatically queries and retrieves the data from a data warehouse or OLTP system.

record. A set of one or more related data items grouped for processing. In reference to a database table, *record* is synonymous with *row*.

roll-up. see Consolidate.

rotate. To change the dimensional orientation of a report or page display. For example, rotating may consist of swapping the rows and columns, or moving one of the row dimensions into the column dimension, or swapping an off-spreadsheet dimension with one of the dimensions in the page display (either to become one of the new rows or columns), etc. A specific example of the first case would be taking a report that has Time across (the columns) and Products down (the rows) and rotating it into a report that has Product across and Time down. An example of the second case would be to change a report which has Measures and Products down and Time across into a report with Measures down and Time over Products across. An example of the third case would be taking a report that has Time across and Product down and changing it into a report that has Time across and Geography down.

S

scoping. Restricting the view of database objects to a specified subset. Further operations, such as update or retrieve, will affect only the cells in the specified subset. For example, scoping allows users to retrieve or update only the sales data values for the first quarter in the east region, if that is the only data they wish to receive.

selection. A selection is a process whereby a criterion is evaluated against the data or members of a dimension in order to restrict the set of data retrieved. Examples of selections include the top ten salespersons by revenue, data from the east region only and all products with margins greater than 20 percent.

slice. A slice is a subset of a multi-dimensional array corresponding to a single value for one or more members of the dimensions not in the subset. For example, if the member Actuals is selected from the Scenario dimension, then the sub-cube of all the remaining dimensions is the slice that is specified. The data omitted from this slice would be any data associated with the non-selected members of the Scenario dimension, for example Budget, Variance, Forecast, etc. From an end user perspective, the term slice most often refers to a two- dimensional page selected from the cube.

slice and dice. The user-initiated process of navigating by calling for page displays interactively, through the specification of slices via rotations and drill down/up.

sparse. A multi-dimensional data set is sparse if a relatively high percentage of the possible combinations (intersections) of the members from the data set's dimensions contain missing data. The total possible number of intersections can be computed by multiplying together the number of members in each dimension. Data sets containing one percent, .01 percent, or even smaller percentages of the possible data exist and are quite common.

Structured Query Language (SQL). An established set of statements used to manage information stored in a database. By using these statements, users can add, delete, or update information in a table, request information through a query, and display results in a report.

symbolic name. In a programming language, a unique name used to represent an entity such as a field, file, data structure, or label. In the Intelligent Miner you specify symbolic names, for example, for input data, name mappings, or taxonomies.

T

transaction. A set of items or events that are linked by a common key value, for example, the articles (items) bought by a customer (customer number) on a particular date (transaction identifier). In this example, the customer number represents the key value.

transaction ID. The identifier for a transaction, for example, the date of a transaction.

List of abbreviations

ADK	application development toolkit
ANSI	American National Standards Institute
APPC	advanced program to program communication
API	application programming interface
APPN	advanced peer to peer networking
ASCII	American National Standard Code for Information Interchange
BI	Business Intelligence
CAE	client application enabler
CP	control point
CORBA	Common Object Request Broker Architecture
CPI-C	Common Programming Interface-Communications
DB2	database 2
DBA	Database Administrator
DBMS	database management system
DCL	data control language
DDL	data definition language
DML	data manipulation language
DR	distributed request
DRDA	distributed relational database architecture
DSS	decision support system
DUW	distributed unit of work
DW	data warehouse
EIS	executive information system
FTP	file transfer protocl
GID	group ID
GUI	graphical user interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HLQ	high level qualifier
IBM	International Business Machines Corporation
IDS	intelligent decision support
ISO	International Organization for Standardization
I/O	input/output
IM	Intelligent Miner
IMS	Information Management System
ISDN	integrated services digital network
IT	information technology
ITSO	International Technical Support Organization
JCL	job control language
JDBC	java database connectivity
JDK	java developers kit
JRE	java runtime environment

LAN	local area network
LOB	large object
LU	logical unit
ODBC	Open Database Connectivity
OEM	original equipment manufacturer
OLAP	on-line analytical processing
OLTP	on-line transaction processing
OSA	open systems adapter
OSI	open systems interconnection
POS	Persistent Object Service
QMF	Query Management Facility
RACF	resource access control facility
RAD	rapid application development
RAM	random access memory
RDBMS	relational database management system
ROI	return on investment
RUW	remote unit of work
SDF	Server Definition File
SMP/E	system modification program/enhanced
SNA	shared network architecture
SQL	structured query language
TCP/IP	Transmission Control Protocol/Internet Protocol
TP	transaction program
UDB	universal database
UDF	user-defined function
UDP	user datagram protocol
UDT	user-defined type
VSAM	Virtual Storage Access Method

Index

ITSO redbook evaluation

Business Intelligence Certification Guide
SG24-5747-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction __________

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

SG24-5747-00
Printed in the U.S.A.

Business Intelligence Certification Guide

SG24-5747-00