Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

# Business Intelligence

The text is intended for students of course Business Intelligence on the Faculty of Economics

Jihočeská univerzita v Českých Budějovicích
Branišovská 1645/31a
370 05 České Budějovice
IČ: 60076658

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

# CONTENT

# 1 INTRODUCTION

At present, one of the most important things for both companies and individuals is always to have up-to-date, credible, detailed and available information about their internal and external enviroment. Information serves for decision-making in various rapidly changing conditions. The company may wrongly decide on the basis of a lack or poor quality of information, and this error can have serious consequences (at best, only time and financial costs). The issue of data acquisition and analysis from various sources, both internal, business and external, can be described as Business Intelligence (BI). Business Intelligence can be perceived as the evolution of information systems in enterprises. It provides medium / higher management with the right information and knowledge not only about own business but also about other factors that can influence business processes and business success. Business Intelligence allows to get and glance information that can be overlooked in the course of normal traffic, or can be seemed as not to be important at the first sight. BI in itself is not a guarantee of success, but it is a very powerful tool for internal analysts to help them generate news and forecasts about future developments and changes in business operations as well as changes in the external environment. Herewith, BI allows to management to set short / long-term goals or strategies to ensure the success in business world.

The following text explains the basics of business intelligence, including the design and implementation principles of a data warehouse. The text is intended for students of the Faculty of Economics of the University of South Bohemian in České Budějovice.

# 2 THE ROLE OF BUSINESS INTELLIGENCE IN ENTERPRISES

## 2.1 The concept of BI in a historical context

IBM computer scientist Hans Peter Luhn, who defined it as „the ability to understand the relationships between the facts presented in such a way that an appropriate action is possible to reach desired goals“, mentioned the term „Business Intelligence“ (BI) for the first time in 1958.

In the late 1970s, solutions of managerial and analytical tasks in corporate governance in connection with online data processing started to be used. The first attempts from this area are associated with the company Lockheed. In the '80s the first commercial applications Comshare and Pilot were launched in the United States. We consider them to be the first BI software. They were based on multidimensional data storage and processing. We also include them in the group of the systems that support company strategic decisions called EIS (Executive Information System). The market for these products was rapidly expanding in the 1990s, and since 1993 these products have also begun to appear on the Czech IS/ ICT market. At the same time, data warehouses (Data Warehouse) and data market (Data Market) conceptions appeared. Exponentially increasing data volumes also influenced to establishing data mining (Data Mining) technology.

The term Business Intelligence became known in the public at about 1989. The source was the analyst at Gartner, Inc. Howard Dresner who redefined the concept of Business Intelligence as "a set of concepts and methods designed to improve the company's decision" His definition is much closer to today's concept.

In the mid-eighties, large companies, especially banks, used stand-alone reporting systems quite commonly. With the newly expanding type of IT infrastructure, which also began to represent an interesting market, the need for a description of this new concept arises. In the late eighties, the first article, which described the architecture of the data warehouse by the author William H. Inmon, appeared. William H. Inmon is called therefore the "Father of data warehousing" although data warehouses existed before. A little later (1992), W.H. Inmon published the famous book "Building the Data Warehouse". Books on the subject BI then flooded the book market.

As the reporting abilities of data warehouses developed, new tools also were developed. Their main purpose was to help users with an analysis of existing reports directly above the databases. This was the main reason for creating "On-Line Analytical Processing "(OLAP) tools that allow ad-hoc analysis with acceptable speed and sufficiently low by limiting the scope of possible data analyzes. The emergence of these instruments is related to the realization of multi-dimensional data modeling, that was introduced by Edgar F. Codd in 1993. The same author, in the 1970s, designed and implemented relational data modeling, which is still the basis for the design and the use of relation databases (RDBMS).

Edgar T. Codd also defined the properties of OLAP tools using twelve rules to derive 18 properties of OLAP tools. However, this rather complicated definition has not been used widely. A simplified and comprehensible OLAP definition known as FASMI (Fast Analysis of Shared Multidimensional Information) appeared in the OLAP report (www.olapreport.com) in 1995.

The increasing complexity of data analyzes and their statistical processing has given rise to Data Mining concept. This concept originally meant the search for mutual, yet unknown, dependencies in data. Today, this term is used for procedures and tools used for any statistical processing of data - mainly from data warehouses. These tools allowed a new look at stored data using mainly statistics and vizualization, mainly for the purposes of modeling, planning and anticipating the development of companies according to forward-looking indicators.

The latest, now steeply evolving direction in BI is the increasingly intensive use of data warehouses to communicate more effectively with customers through CRM (Customer Managemet Systems) systems, communication with partners and business vendors through SCM systems (Supply Chain Management) and the use of sources of external information (social networks, Internet source) and more.

The whole development is illustrated on the following picture:

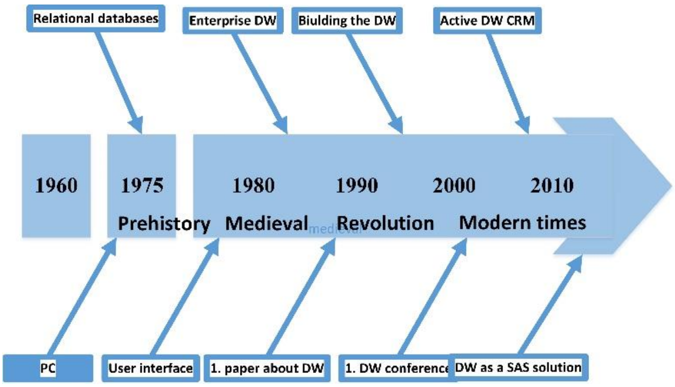


Figure 2.1: BI development timeline

At the present time, there are more definition of the BI. Every author when dealing with BI often introduce his/her definition, and the perception or awareness about what BI is (what it includes, and what is not) is different. In general, however, current BI is "a specific type of business informatics tasks that almost exclusively support analytical, planning and decision-making activities, and are built on the principles that best match these activities."

BI is a "suite of processes, know-how, applications and technologies designed to effectively and efficiently support management activities of organizations at all levels, and in all areas of corporate governance, i.e., sales, purchasing, marketing, financial management, asset management, human resource management, manufacturing, and more. "

BI applications are built on the principles of multidimensional business data views. These are tools using mathematical apparatus and graphical interpretations processing data from ERP (or other) systems (ERP – Enterpeise Resource Planning) so that managers get a faster and better insight into the company's performance on these data outputs and can make the right decisions based on a large amount of data.

## 2.2 Definition of the notion Business Intelligence

**Our definition is one of the more possible definitions:**

**Business intelligence** = processes, technology, and tools needed to transform data into information, information into knowledge and knowledge into plans that enable actions to be taken to support the organization's primary goals.

Let’s explain the BI approach on a bank example. Let’s consider a bank that has three basic resource systems for simplicity: banking system (bank transaction), sales system (sales records at individual branches), and credit approval system (work-flow system with implemented process of approval). Each system uses its own database.

Data from all three systems is integrated and consolidated into a special big database, bank's data warehouse. Integration and storage of data in one place allows users to easily obtain information such as: how many new and how many existing clients the bank has sold its products over the past week, how many existing clients are applying for a loan, and which stages of processing these credits are in (i.e., data → information).

Advanced users (analyst) are able to determine trends in the development (for example, seasonal fluctuations in the sales of a particular banking product), or patterns of client behavior (if a man between 35 and 45 years buys product A, he also purchases product B within 3 months). There is a transformation of information into knowledge (i.e., information → knowledge).

Knowledge gained from information allows us to prepare rules and plans. Such a plan can look like this: if there are more than 100 requests for credit in a processing stage X at one time, the declared processing time will not be met with a high probability. If such a situation occurs, the credit department must be strengthened by more employees immediately. Other rules are set based on the success rate of sales. They can then trigger local marketing campaigns. Knowledge has been transformed into concrete plans (knowledge → plan).

The plans are implemented through actions to support the bank's business (events). The results of events are then projected into data in source systems that integrate into the data warehouse to generate new information, knowledge, etc.

## 2.3 Relationship BI, ECM and KM

Enterprise content management, so-called ECM (Enterprise Content Management is a technology that provides the means to create, manage, store, publish, search, personalize, and present all digital content. The main goal of the ECM concept is to provide relevant real-time information to those who need it for their decisions with an increasing volume of data. Knowledge management, KM (Knowledge Management) aims to bring together those who know with those who need to know at the moment, or to "transform the knowledge of individuals into an organization's knowledge." KM is based on the fact that useful data loaded with redundancy (i.e, the noise that needs to be filtered out) is flowing into the company and its surroundings and transferred to the information or data that have a meaning derived from the context. Information can be interpreted and classified. Knowledge is value-added information that makes it possible to make decisions, it depends on experience and ability to understand. The wisdom, on the other side, it cannot be shared as knowledge because it is associated with the process of individual learning, understanding and comprehension, and it has therefore one of the top positions (Figure 2.2).

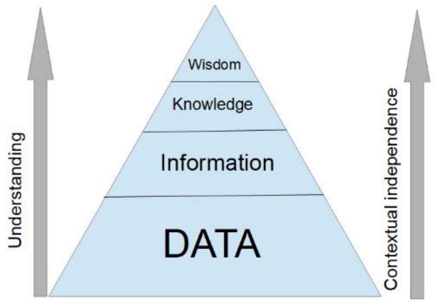


Figure 2:2: Pyramid of knowledge management (Custom Processing, 2014)

ECM provides accessibility, addressability, uniqueness and security of content. BI tools are geared towards transforming content into knowledge, i.e., transforming data into knowledge. These are then by KM managed throughout the lifecycle. KM and BI and ECM systems work closely together to create new content that is more usable by the organization.

## 2.4 Who is BI determined for?

In large enterprises, BI is applied for a longer time and more often than in small and media enterprises (SMEs). Each manager is in charge of specific tasks to ensure company operations (Figure 2.3) BI applications are intended primarily for high and medium managers, analysts and planners. BI allows for a more precise focus on giving every employee of the enterprise the same "version of truth". BI products can improve the quality of management decisions and thus increase business competitiveness.

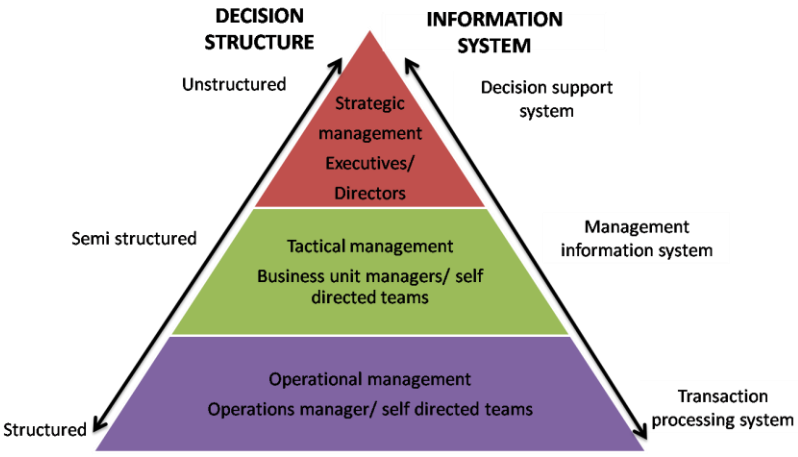

DECISION STRUCTURE
INFORMATION SYSTEM
Unstructured
Strategic management Executives/ Directors
Decision support system
Semi structured
Tactical management
Business unit managers/ self directed teams
Management information system
Operational management
Operations manager/ self directed teams
Structured
Transaction processing system

Figure 2.3: Levels of management and decisions

The need and possibilities of using BI in SMEs (Table 2.1), where the limited business scope, the complexity of their partner structure, and overall management style (mostly management style where managerial experience, informal relationships and business sense are decisive) is prevailing, are increasingly being considered. New BI tools for this SME segment have an impact on this decision.

Table 2.1: Category of SME according the EU Commission (http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_cs)

| Company category | Staff headcount | Turoover | Balance sheet total |
|---|---|---|---|
| **Medium-sized** | < 250 | ≤ € 50 m | **≤ € 43 m** |
| **Small** | < 50 | ≤ € 10 m | **≤ € 10 m** |
| **Micro** | **< 10** | **≤ € 2 m** | **≤ € 2 m** |

## 2.5 The position of BI in enterprise information systems

BI presents a wide range of tools and applications. BI applications are built on the use of data generated elsewhere, mostly in ERP (Enterprise Resource Planning) applications (Figure 2.4). If these applications and their databases are poor, then the quality of BI solutions is also greatly affected. In addition, this lack of performance is much more striking than in so-called transactional applications, while reducing user confidence in BI. In other words, the quality of the BI solution depends on the quality of the production data provided from the individual databases.

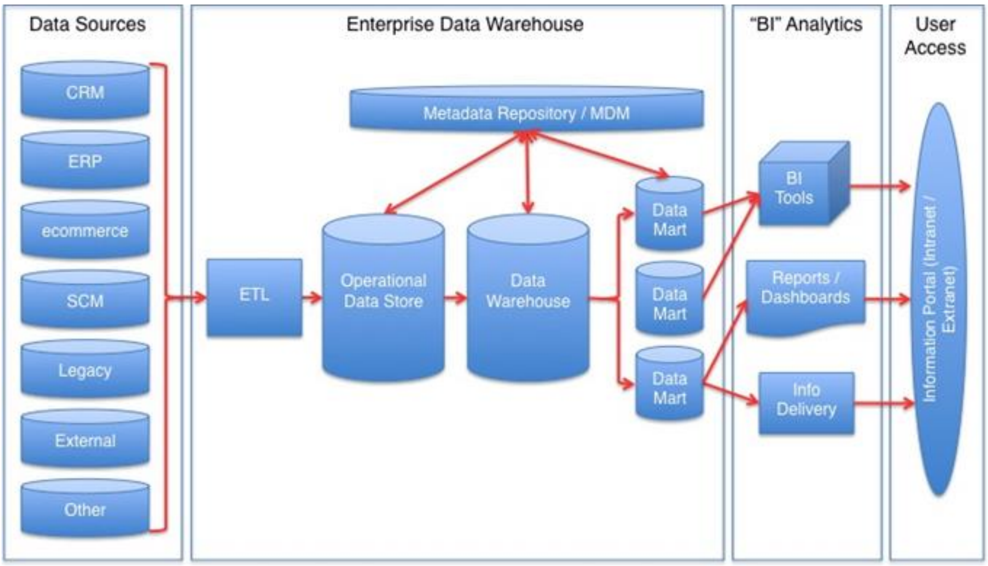


Figure 2.4: Data sources for BI analytics

The BI status of enterprise information systems and related processes is shown in Figure 2.5 BI utilization belongs to the EIS and DSS level.

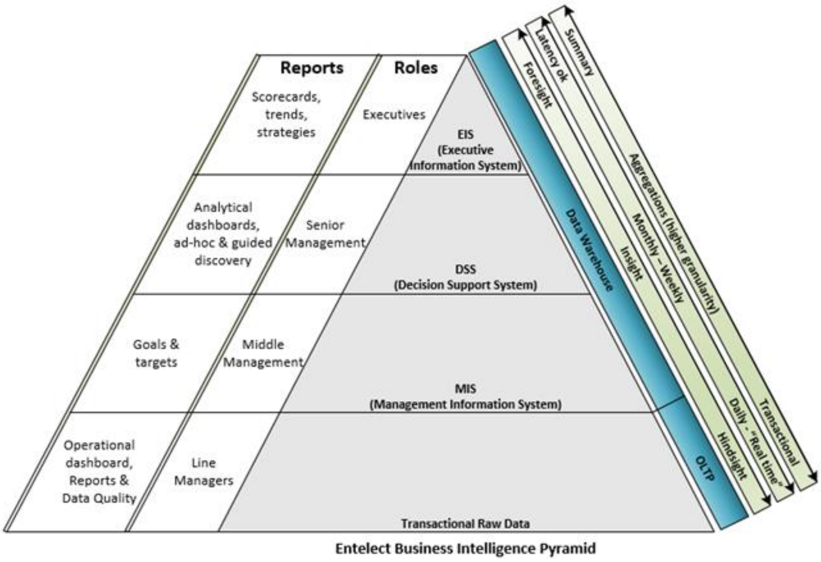

Reports
Roles
Scorecards, trends, strategies
Executives
EIS (Executive Information System)
Analytical dashboards, ad-hoc & guided discovery
Senior Management
DSS (Decision Support System)
Goals & targets
Middle Management
MIS (Management Information System)
Operational dashboard, Reports & Data Quality
Line Managers
Transactional Raw Data
Data Warehouse
OLTP
Foresight
Insight
Hindsight
Latency ok
Monthly – Weekly
Daily – "Real time"
Summary
Aggregations (higher granularity)
Transactional
Entelect Business Intelligence Pyramid

Figure 2.5: BI in the context of enterprise information systems hierarchy (Source: http://source.entelect.co.za/bi-instant-expert)

# 3 GENERAL BI ARCHITECTURE

Business Intelligence (BI) is a concept that covers concepts such as data warehouse, data mining, OLAP Online Analytical Processing), data transformation, and many more. The concepts of BI architecture and related methods have stabilized. In this chapter we will deal with the main architectures of the BI solution and explain the main concepts.

In order to get an idea of what areas of BI is involved in, and what principles BI is based on, we will first introduce a simple business information model that will help us graphically illustrate the influence and the place of BI in the company.

## 3.1 Information model of a company

The basis of the information model is the fact that each firm or institution always deals with four basic activities:

1. It produces the object of its trade (the material reason for its own existence).

2. Interact with the outside world.

3. Monitor the events inside, the intensity and quality of the external interaction.

4. It monitors and measures the financial result of its own activity.

Taking into account the information flows between these parts of the company, we will get the following figure:

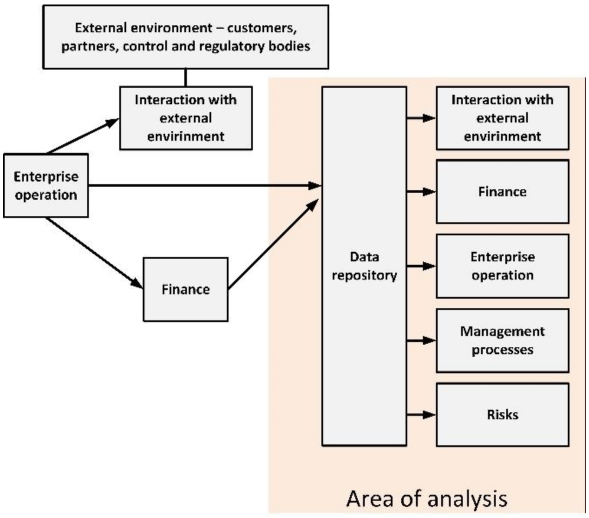

External environment – customers, partners, control and regulatory bodies
Interaction with external envirinment
Enterprise operation
Finance
Data repository
Interaction with external envirinment
Finance
Enterprise operation
Management processes
Risks
Area of analysis

Figure 3.1: Information model of a company

BI system incorporates information from production, financial systems, operation systems, management processes information, risks information etc. into an analytical field including business external environment. The results of analyzes also have an impact on production systems - for example in banking, a data warehouse is often used to calculate a client score, which is then imported back into operating banking systems (see aforementioned example). BI system is by its scope and importance to the largest and most important company structures.

## 3.2 BI Architecture

Bi architecture can be divided into the following parts:
1. Transformation environment - extraction and transformation of data.
2. Data environment - Data repository architecture.
3. Reporting Environment - Reporting Tools, OLAP, Data Mining ...

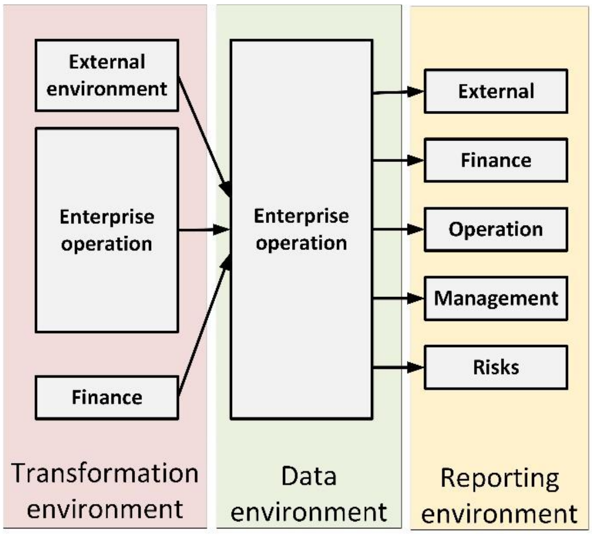


Figure 3.2. Information model of an organization

Within the transformation environment, the data transfer to the data repository target data - data transformation is realized.

The first step is always to extract data from operating systems. In this case, the rule (based on experience with many projects) applies - generating extracts should be ensured as far as possible by the production system itself. The main reason is the maximum possible limitation of the load of operating systems by extraction of data. Only administrators of these systems know when and how these extracts can be generated.
At present, the data transformation process is done in two ways:
1. Extraction - Transformation - Load (import) (ETL).
2. Extraction - Load - Transformation (ELT).

### ETL

In this process, data is transformed into target structures on a transform server that is not part of the target data repository system. Saving data into temporary structures for final reprocessing is called as "staging" and the appropriate directories is called "stage" database. In addition to extracted data, the transformation server includes software tools for data transformation, including implemented scripts.
Reworked data is then added to target structures. The process is schematically illustrated in the following figure:

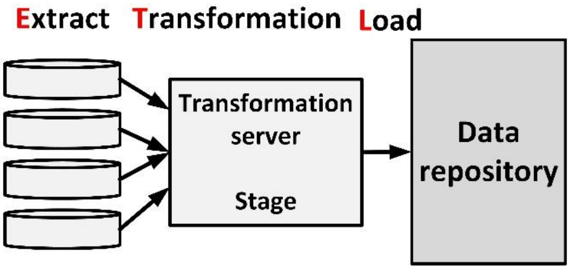


Figure 3.3. ETL Architecture

The benefits of this procedure are as follows:
- The data transformation takes place outside of the target data repositories. Data repositories are therefore available for a longer period of time for user needs.
- Some complicated algorithms can be better implemented within this architecture.

This process also has its disadvantages - in particular:
- Data transformations are often very demanding to perform.
- Some ETL tools require for data transformation data not only from production systems but also from the target repository. The whole process of data transformation is then more complicated.

### ELT

This approach is based on the fact that the performance of the database system of the target data repository is also used for the data transformation. "Stage" is one of the databases of this data environment. The target data repositories are used over working hours for users and at night for data updating. ETL server is used for storing extracted data and controlling the whole data transformation but this server does not work on transformation. The architecture of such a solution is presented on the following figure:

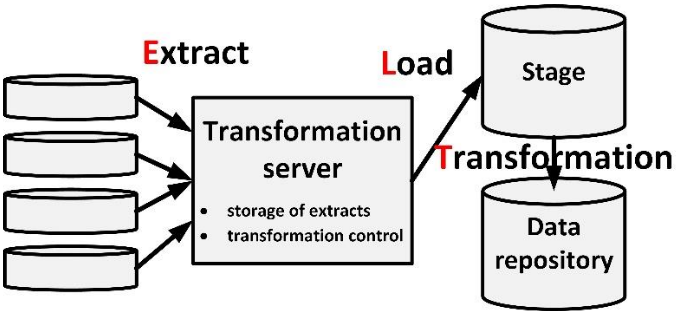


Figure 3.4 ELT Architecture

**The benefits of this architecture are:**

- The enhance and the speed up the data transformation by enhancing the performance of the target data repository environment.
- ETL server performance requirements are very low.

**Unfortunately, this solution has drawbacks, of which the most important are:**

- Data transformation requires a majority of resources for itself, so the system is not accessible to users during the transformation process.
- The ELT tools must be well integrated with the database technology used for target data repositories.

**The choice of architecture and tools for data transformation**

Both of these described approaches to data transformation have their pros and cons. The right choice always depends on a specific situation. When selecting architecture, at least the following aspects should always be considered:

- Existing technology environment.
- Compatibility of the ETL / ELT tools with the target database.
- Availability of the ELT or ETL supplier on the local market. Existence and quality of implementation services - it is best to verify the references.

One of the main advantages of ETL / ELT tools is cost savings made possible by reducing the number of developers and facilitating implementation through the graphical interface. However, given the prices of these products, it is always necessary to calculate this intention carefully. The main advantage of all tools of this type that we always use is the interface for monitoring and managing of transformation processes and metadata support. These two aspects need to be examined with particular attention.

## 3.3 Data repositories

Data warehouse, data-mart, operational data warehouse and concepts related to the organization of data are linked to the data repository architecture - dimensional data modeling and entity relational model in the third normal form. We will explain these when describing the data repository architectures.

**Date Mart**

Data mart is a thematic-oriented database serving one specific purpose - mostly serving the information needs of one unit. Its data structure is subject to the required outputs. Data marts exists mostly because each business unit needs its own information environment, and data mart intended for example for marketing does not suit the department of traffic.

Depending on whether directly operating (transaction) systems or a data warehouse are the data source, we recognize the independent and dependent data marts. Usually, data marts are performed as separate databases, sometimes on separate hardware. There is also the ability to create the necessary dependent mart data using a view as a logical layer above the data warehouse.

The logical diagram of architecture with independent data mart is shown in Figure 3.5.

This architecture is "seductive" by the fact that the individual dates of the marty are simple, cheap and fast to load and the first results can be seen soon.

Unfortunately, after a certain time, it is usually found out that the number of data marts is too high, the data in them and the reports created above do not match and the maintenance costs grow rapidly, not to mention the burden of operating systems that have to extract many data many times into various data marts. Therefore, today's architecture is abandoned, and most companies try to either consolidate the existing data of data marts or go from the beginning through the implementing of a central data warehouse.

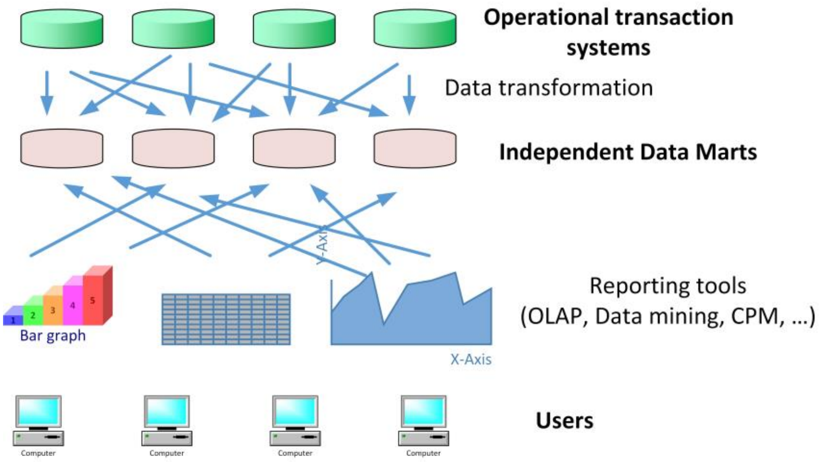


Figure 3.5: Data repository architecture of independent data marts

Data warehouse

- Data warehouse ïs considered to be a full-fledged / central / consolidated data warehouse (also Enterprise Data Warehouse (EDW)). It is characterized by the following features:
- Data warehouse is object-oriented. The data warehouse objects are a set of all data that is extracted from the operating applications on the subject. An example of the subject may be "CUSTOMER." This subject may include customer data and its history.
- Data structure is not dependent on specific report tools or applications. In other words, the data structure is application-independent.
- Contains detailed data from many production systems in historical ranges. The amount of aggregate data is minimal.
- By integrating of operating system data into one consolidated data model, a "one version of the "truth" is used to make consistent reporting.
- It serves to reporting, to data-mining and to data analysis needs across the company.
- Must be, if required, directly accessible to end users.
- Can provide simultaneous access to a large number of applications, reporting platforms and end-users.
- Must be able to respond to ad-hoc queries without compromising accessibility for other users.
- DW architecture is scalable - both in terms of data organization and in terms of performance parameters and a number of parallel approaches.
- It is updated regularly, the update frequency may range from near-real time updates (order minutes) to daily or weekly updates.

**Organization of data in DW**

The data must be organized in relational databases according to the subject areas whose data are stored in the DW. This is a concept different from the data modeling of the operating system structures where the primary data modeling is based on the processes handled by the system. For this subjective modeling, as the experience shows, the most appropriate entity-relational model is in the third normal form. This simply means that each attribute (data column) occurs in the model only once except those attributes (columns) that are used for relationships between entities (tables). The exact definition of the third normal form as well as other normal forms - defined by at least five - can be found on the web [1], [2] or in the classical literature - for example, book 13 is very nice.

This approach to data warehousing was founded by the "father" of data warehousing, by William Irtmon, and his book [4] is one of the best and most readable in this field.

It is worth mentioning that Ralph Kimball, also one of the most important figures of the warehousing date, holds the contradictory position published in his excellent book [5J. In essence, he argues that the only rational way of modeling data in data warehouses is dimensional modeling (see, for example, [5j). However, practice shows that for DW, Inmon's approach is more appropriate. His opponent's views, however, are not mistaken, but they are useful at the implementation of data marts (that have already been mentioned).

Detailed versus aggregated data

The data warehouses usually store detailed data in historical rows. However, the storing of detailed data requires a very powerful database engine that can calculate the required aggregation at a user-friendly time. If it is not possible to ensure such performance, either because of the database machine itself, or just the budget is not enough for a powerful enough hardware, it makes sense to store aggregated data, but aggregations hide one danger: they suppose some way of data processing which is not ussualy universal. The result can be an inconsistent change of aggregated data, which must be supplemented by detailed data, in this case in the operative data warehouse.

Operational Data Warehouse (ODS)

An operational data warehouse is an entity of the BI architecture, containing detailed data updated with high frequency - it can be even minutes or fewer. The history of detailed data is usually very short - usually one or several days. In addition to these operational data, it also contains copies of some aggregates from the data warehouse related to the purpose of the ODS. Unlike DW, the ODS contains data about a precisely defined goal - application or operational reporting. That is why the number of ODS is usually more.

Note: remember that the need for operational data warehouses is the result of non-compliance with the basic DW properties, which is often forced by the deficiency of the DW technologies selected. Typical examples are the lack of detailed data in data warehouse structures, ie an inappropriate data model, and the fact that the platform does not support frequent and on-line data updating. Another option is that an application that uses this data is not compatible with the DW platform.

An example of an ODS architecture is shown on the following schema:

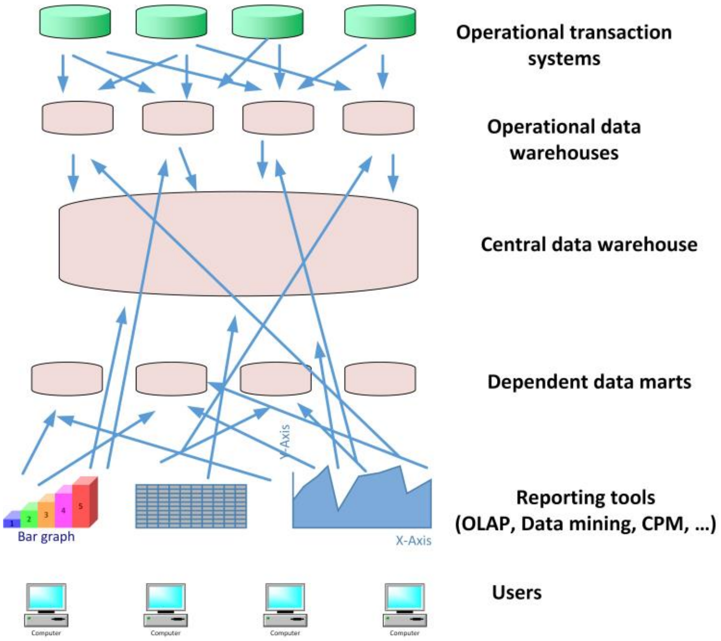

Operational transaction systems
Operational data warehouses
Central data warehouse
Dependent data marts
Reporting tools (OLAP, Data mining, CPM, ...)
Users
Bar graph
Y-Axis
X-Axis
Computer
Computer
Computer
Computer

Figure 3.6. Data repository architecture composed of central DW, ODS, and dependent data marts

If possible, it is necessary to build the BI architecture with as few data repositories as possible. The larger the amount of data repositories, the more interfaces and transformation algorithms need to be maintained. The cost of managing and expanding complex architecture is high; the ability to respond to changes in user requirements is low. It is appropriate always to create an architecture with a central data warehouse and a minimum of dependent data marts

## 3.4 Reporting environment

This part of the BI architecture is visible to end users. It is created from tools that present data stored in a data repository. At first, tools whose main purpose was to help users analyze existing reports directly on the database - i.e. "on-line" were created. These tools are denoted as on-line analytic processing (OLAP) tools that allowed ad hoc analysis at acceptable speed and sufficient small limitation of the scope of possible data analysis.

OLAP tools are divided according to the way they access the data repository. The first variant uses a data warehouse (or some data mart) and the OLAP tool generates the necessary SQL queries and formats the output. Tools with this "on-line" approach are called Relational OLAP, abbreviated as ROLAP. The disadvantage of this process is the extraordinary burden of the data repository, which raises the risk of long-term responses for end users, so ROLAP tools are equipped with smart cache to minimize the operation of database engine On the other hand, the user always gets an answer obtained directly from the most up-to-date data.

Responding to ROLAP Response Time Response Problems offers an alternative called MOLAP (Multidimensional OLAP). This architecture is based on a pre-made computation of data from a data repository into proprietary files (often referred to as "multidimensional cubes"), so the response to the user request is almost immediate, and the data repositories are loaded only during data cube filling, which, however, may be lower than the frequency of the data repository updates.

The main difference between ROLAP and MOLAP is that ROLAP only stores cube metadata, while MOLAP stores physically pre-calculated cubes.

Some vendors combine benefits of both approaches, and mark their products with the name of HOLAP-Hybrid OLAP.

Architecture BI is nothing complicated, but it is an extensive structure, the simple points for successful architecture building can be summarized as follows:

- Decisions about the BI architecture have a major impact on the future ability to respond to user requirements, and thus to the entire lifetime of BI's business environment.
- Design of the ETL architecture has a very significant impact on the total cost of the BI architecture. Often it is up to 50% of all costs. At the same time, it will significantly affect the scalability of the whole solution.

The main consideration here is the compatibility of this tool with existing production systems, the platform chosen for data repositories and the functionality of the interface for managing and administration of the warehouse and data warehouse updates.

- It is necessary to keep the architecture as simple as possible - especially with the data repository architecture.
- Claims on the BI architecture are significantly different from the requirements for operating systems. It is necessary to not to be afraid to think about new technologies and platforms.

# 4 BUILDING OF BI

BI solution is not a SW package that it is possible to buy in a box at a merchant, to install it according to the manual and to start operating. Building a BI solution is a long-lasting (virtually endless) process of constant change as well as constantly evolving the needs and processes of organization.
In order to build a BL solution, a specific HW infrastructure and SW resources are required, arranged in the right architecture. But the solution is far from being a technological matter. As the name of the area itself suggests, the main focus of the solution is to support decision-making of management and of company processes. Therefore, the building any of the solutions belonging to category BI can not be successfully separated from the business context. Analytical preparation The BI solution should reflect a strategic view of the company and its goals: it is not about doing the same things we do now but using the potential of the newly developed solution for improvement and optimization. According to Gartner, we can expect a tendency toward "process-driven BI, where reports and analyzes are directly embedded in the business process flow." Building a BI solution should therefore be part of implementing of strategic visions.

## 4.1 The process of BI bulding

Building a business solution is a long-term and complex task. It should be implemented in the form of a project(s). The necessary first step is to develop an overall concept of a solution (feasibility study) following the global strategy of the company. A feasibility study should provide answers to these basics questions:

- Which areas (and which strategic objectives) will the solution be primarily covered / supported by the solution and how?
- Who is the top management to be the "sponsor" of the project, who owns the individual parts of the solution and who acts as a gestor and promoter within the company?
- What will be the architecture of the solution and what technological components will be optimal for the company in terms of performance / price ratio (also with regard to the expected future development)?
- How will the total subject matter be divided into the implementation phases (increments)?
- »What timetable and gross budget will the individual increments?
- What implementation mode will be chosen (own resources, suppliers, outsourcing, combinations), or who will be the supplier of the individual components of the solution?
- What risks are associated with implementation?

Very often, the feasibility study is developed with the participation of an external contractor. Such a process helps to minimize the impact of a certain "operational blindness" of own staff, and at the same time makes it possible to use effective use of experience from other similar solutions.
The intended target range of the BI solution is always useful to divide into partial sub-areas realized in incremental increments, taking into account the priorities. Each incremental increment must fit into the overall concept of the BI solution. Experience has shown that, given the size of the organization and the material extent of the initial stages, the first results can be achieved within 3 months.
The partial steps of building the BI solution, resp. one of its increments, may be specific depending on which particular technology components are chosen for implementation. To some extent, however, it is possible to formulate a generally valid procedure ("travel" map) as shown in the following figure.

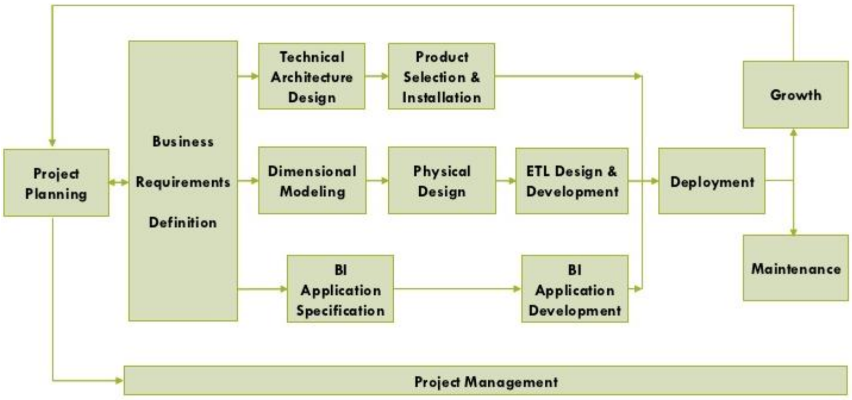

Project Planning
Business Requirements Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
ETL Design & Development
BI Application Specification
BI Application Development
Deployment
Growth
Maintenance
Project Management

Figure 4.1 The road map of the BI project

The activities carried out in the individual stages and their standard outputs necessary for further progress or serve as the documentation of the developed solution:
In the phase of business justification, the so-called business case (BC), the main subject of which is a structured proposal for the required change, arises. It should be a standard precondition for opening a larger project and is explicitly required by many project management methodologies. BC on a coarse level addresses the business needs to be addressed by the project. It includes the rationale for the project, the expected benefits of use, the options considered (including their pros and cons), the expected costs of implementation and operation and the risks assumed. Almost always should be included! "do nothing" option and should be compared with implementation options. BC is the basis for the decision to start or stop the project. During the project BC should be periodically verified to ensure that "BC is still valid, i.e., there is still a business need and the project is still aimed at meeting the business needs.
As a result of a regular review, the entire project may be terminated (possibly some of its phases are canceled) or, alternatively, extended to another area of the solution.

Project planning is an essential component of project management. The input is specifying the scope of the project and determining the methods and procedures used to implement it. Then, the individual steps and tasks, their work breakdown structure (WBS) are defined. Definitions and dependencies of individual tasks are defined, which makes it possible to identify the critical way of realization.

Each task is assigned the necessary resources and the project costs are determined beforehand. The plan can be optimized in order to balance the use of resources with the duration of the project. During the project, the plan regularly compares the current status and development of project activities.

In the initial phase of the project, only a framework plan is compiled, and the detailed implementation plan is useful to compile only after specifying the real scope in the business analysis phase.

Business (Conceptual) Analysis (BA) focuses on identifying / analyzing requirements to help companies implement strategic goals through internal changes in organization, culture, processes, and systems. Part of this phase is mapping existing processes, systems / functionalities / data and organization, comparing with requirements, finding differences and designing a solution concept.

The components of the design are:

- Concept of relevant business processes in the form of a process model.
- Conceptual model of content (data) and functions, rough drafting of reports.
- Concept of data flow architecture.
- Concept of IT architecture and infrastructure design including inclusion in the overall IS/ICT architecture of the organization.
- Identification of sub-areas, definition of their priorities and grouping in individual implementation increments, definition of interdependencies, assumptions, etc.

- Detailed implementation plan.

BA takes place in close cooperation with key business representatives, often through workshops, interviews that serve to find out information, verify mutual understanding, and agree on proposed solutions.

The main objective of the detailed analysis phase is the detailed elaboration of the conceptual design in all its components (relevant for a particular increment). In the eventual subsequent increments, the analysis of the impacts of the planned changes on the already implemented solutions and the possible modification of the conceptual framework defined in the business analysis phase are also included. Detailed analysis and design are summarized in a set of documents and models in accordance with the established methodology.

Detailed analysis and design undergoes a thorough acceptance process involving business users (gesturs) and key IT architects. The main goal of the approval phase is to verify whether the proposed solution meets the business requirements. This must be done before starting of own implementation.

During the implementation phase, the development team (in close cooperation with the analytical team) creates and integrates elements according to the detailed analysis (database, ETL process components, applications, reports). Part of the implementation phase includes internal tests conducted by the development team.

The main activities of the testing and deployment phase are:

- Migration of all developed components and relevant data into the test environment.
- Preparation and implementation of user acceptance tests, troubleshooting and user training.
- Migration to production environment (including initial load).
- Implementation of changes caused by deployment of BI solutions (process changes, methodologies, etc.). One of the key related processes is the quality management of the source data used in the BI solution.

The final action is the final acceptance of the solution developed. During all phases, quallity assurance is required .

The procedure of implementation of each additional increment or change of the BI solution is basically the same as in the initial phase of the entire activity. However, it is clear that at various stages of the "maturity" BI solution of the company it is no longer necessary to carry out some steps of the general procedure, for example, a partial increment of Implementation of BI application without changing the existing data base.

The natural requirement of the user is that the implemented solution fulfills the intended goals in all respects. It is not just about the functionality but also about the content of the presented information. For example, it is necessary, but not sufficient, that the data base administrator of a BI solution (Data Warehouse, DWH, or Data Warehouse, ODS) knew that the data transformation was successful. The end-user of the information contained in the data base must know it almost at the same time. Likewise, the user must be able to find out:

- What data (what quantities) does Bi solution offer?
- How these values are generated, respectively. how do they relate to primary data from operating systems (so-called transformation rules)?
- What business rules are applied In extensions to BI applications?
- What is the validity range of available data?
- etc.

Sufficient end-user documentation, i.e., created using business terminology, is often a neglected part of the BI solution. Information is trustworthy and meaningful only if it is obvious not only their content and meaning, but also their origin. A very important part of BI solution is therefore the documentation and especially the so-called metadata (metadata = data / information about data). User metadata can be divided, in principle, into two categories:

- Descriptive - Description of data structures, the importance of individual objects and their quantities, description of transformation and business rules applied in the process of filling these structures and in BI applications. Administrators of this type of metadata must have sufficient knowledge of the company's business, because only this way they are able to ensure their comprehensibility for end-users.
- Operational - Information on the availability of data / reports for a given date, about errors detected in data (fatal or automatically corrected), etc.

## 4.2 Methodological framework

A number of standard methodologies exist for the implementation of BI projects. These methodologies summarize and generalize knowledge from already implemented BI solutions and provide a library of templates covering the entire process of management and implementation. BI vendors generally use their own methodology based on a general standard, enriched by their own experience.

The BI solution is constantly evolving. In order not to compromise the consistency and the existing functionality during the changes, it is useful to formulate a general binding methodology for the initial and the change implementation in the initial stages of building BI solutions.

Part of the methodology is usually:
• Procedures and output patterns for each phase.
• Techniques and best practices for selected activities.
* Sample data and process models for a given business area (banking, telecommunications, etc.).
* Model architectures of BI solutions for a given business area or technology.

A separate chapter is the management methodology of a BI project. The methodology includes, in particular, the rules and support for the activities of the project director, founder and project bodies (such as steering committee, change committee, acceptance team, quality team) in the various phases of the project, ie preparation, implementation, change management, quality verification, operation, subsequent support and further development. The methodology covers all areas of project management (scope, time, cost, quality, integration, human resources, communication, risks).

For the bi-solution building project, it is necessary to consider all the generally valid critical factors of the implementation of IT projects. As a decisive factor in the success of the BI solution, the following assumptions are perceived.
The mission of BI solution is to support the company's business processes and the implementation of BI can lead to significant changes. An in-depth analysis of business needs must be made at the beginning of each increment. That is why constant participation of key business representatives is needed, not only at the end of the whole development cycle, when testing and acceptance of the finished solution is already under way. The key is their participation at the beginning of the development cycle, i.e, when business needsvare defined in detail, priorities are set and verifying that the proposed solution concept meets the expected goals is performed. In order to push for the necessary changes, it is necessary to support the project from the top management of the company.

Building of a BI solution is therefore not just a technological matter for the IT team. For this reason, it is even recommended that the IT department of the company does not primarily manage the BI project. Even though his role in the project is irreplaceable. Successful achievement of the expected benefits can only be achieved with the intensive synergy of key business representatives who are able to formulate not just the current but, above all, future needs of their business areas.

Based on practical experience with BI projects, a number of risks have been identified, the treatment of which needs to be given maximum attention before and during the start of the project. E.g.:

- Organizational security - Missing / inappropriately chosen project leader for BI solutions; not ensuring the necessary project roles.
- Range control - too large a matter scope solved at one time; non-reflection of agreed priorities in solution creation; extending the scope of the solution during the implementation.
- Expectation management - Incorrect expectation that BI solution will be a universal remedy for the inadequate functionality of operating systems or malfunctioning business processes,
- Integration of BI solution - reluctance to modify existing business processes and workflows if the implementation of the BI solution requires such modifications.

# 5 MASTER DATA MANAGEMENT

The first attempt at an overall view of the organization was brought by data warehouses that had data from the entire organization (at least for the support of analytical tasks) stored in one place, independent of the individual operational systems. Due to centralization, consolidation of data was to be performed to a certain extent and a layer of enterprise-wide definitions of individual objects should be built. Practice is unfortunately - almost without exception - another. Data warehouses originated mostly as individual increments of overall solutions according to particular departments / processes. Data warehouse requirements were formulated more or less in isolation, only taking into account the existing content of the solution. Unlike transactional systems, the data warehouse certainly brings a greater degree of insight into and independence from individual business areas, but overall, the effect of consolidating information and the "unified version of truth" is less than originally expected.

The main motto of master data nagement (MDM) is a taking care of key (master) data of the organization. Under this, we understand the management process from the acquisition of key data through maintenance to delivery and deployment. MDM goals can be defined as follows:

- Key data in the organization is unified not only in content but also in understanding.
- Key data in the organization is trusted and correct, respectively. quality-built in trusted and stable sources and managed and provided by trusted, stable and reliable systems.
- Key data is available in the organization whenever it is needed, and to all who need it and are entitled to work with it.

Why are there only key data here? Let us realize that we need to be constantly behaving economically. The MDM solution is difficult to implement, and it is advisable to first focus on places with the greatest effect. These are just the key dates because they are critical to the organization's operation.

What are key (maste) data? Key (master) data are usually those that are used in the organization as dials, or reference data. Typical examples of key data are customer data, in its entirety (identification, address / contact, demographic, behavioral, etc.). Furthermore, the data on products, organizational structure, property of the organization and, of course, all relevant links within and between these groups.

In the case of key data, it is not only about its own data but also about its definition. Emphasis on business entity definitions is one of the important features of MDM.

Key (master) data must be first to be derived from existing data scattered throughout the organization. The process of defining these data is not trivial, and in itself requires comprehensive support. If we can get this data (which means its extraction, cleanup, consolidation and then reliable storage - all within a single understanding of the target object, source objects and necessary transformations) we have not yet won. This data awaits a lot of traps in the organization in terms of reducing their quality, misunderstanding of their content, misuse, and more. For this reason, it is necessary to take care of master data very carefully and to establish a process and organization of this administration.

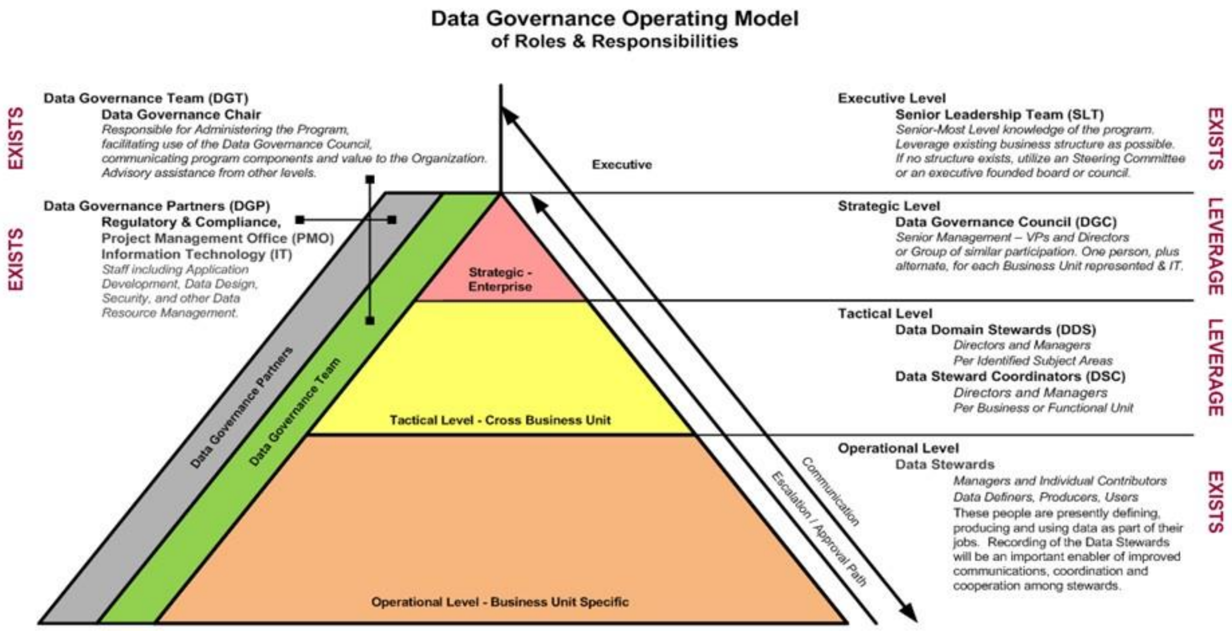
Obrázek 12: Celkové řešení MDM

In order to support the entire lifecycle, MDM has two main and inseparable components

- Technical section.
- Business section.

The specific MDM solution may look like this:
Its process heart is the data governance program. It consists of organizational and procedural measures, and primarily ensures that all key business data have a unified and uniformly comprehended business definition to ensure data quality, consistency and consolidation of all key data, and last but not least, build data and their quality to a prominent place in

the organization's priorities. The data governance program employs several roles - Data Stewards, Data Quality Manager, Data Governance Steering Committee, and others.


Data Governance Operating Model
of Roles & Responsibilities
EXISTS
Data Governance Team (DGT)
Data Governance Chair
Responsible for Administering the Program, facilitating use of the Data Governance Council, communicating program components and value to the Organization. Advisory assistance from other levels.
EXISTS
Data Governance Partners (DGP)
Regulatory & Compliance,
Project Management Office (PMO)
Information Technology (IT)
Staff including Application Development, Data Design, Security, and other Data Resource Management.
Executive
Strategic - Enterprise
Tactical Level - Cross Business Unit
Operational Level - Business Unit Specific
Data Governance Partners
Data Governance Team
Communication
Escalation / Approval Path
Executive Level
Senior Leadership Team (SLT)
Senior-Most Level knowledge of the program. Leverage existing business structure as possible. If no structure exists, utilize an Steering Committee or an executive founded board or council.
EXISTS
Strategic Level
Data Governance Council (DGC)
Senior Management – VPs and Directors or Group of similar participation. One person, plus alternate, for each Business Unit represented & IT.
LEVERAGE
Tactical Level
Data Domain Stewards (DDS)
Directors and Managers
Per Identified Subject Areas
Data Steward Coordinators (DSC)
Directors and Managers
Per Business or Functional Unit
LEVERAGE
Operational Level
Data Stewards
Managers and Individual Contributors
Data Definers, Producers, Users
These people are presently defining, producing and using data as part of their jobs. Recording of the Data Stewards will be an important enabler of improved communications, coordination and cooperation among stewards.
EXISTS

*Developed by KIK Consulting – Last Updated 09/04/2013*

Obrázek 13

The technological heart of the solution is the so-called MDM hub. MDM hub is basically a combination of technologies that allows manual acquisition, automatic extraction of key data from the vicinity of the MDM solution (both batch and real time), consolidation and data cleaning, storage and subsequent delivery (either directly to the end-user or the surrounding systems).
The MDM hub thus consists of:

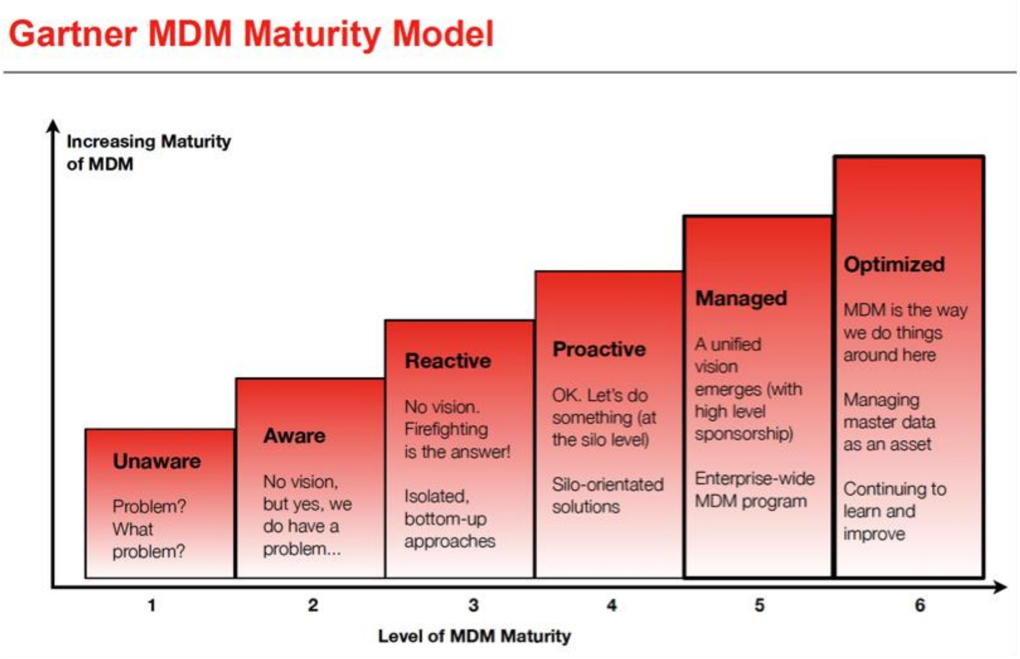
- Database for data storage, including a data model typically market-specific.
- Store all necessary types of definitions - metadata.
- Data transfer / integration tools - batch or real-time.
- Tools and processes for consolidating and ensuring the quality of managed data.
- End user access tools:
    - As a regular data user.
    - In the role of administrator / administrator of data.

Obrázek 14: Technologické komponenty MDM řešení

How does MDM differ from other IS / ICT solutions?

1. Unlike other IS / ICT systems, an integral and crucial part of MDM business processes is the process.
2. For MDM, real-time data transmission and processing technologies (milliseconds) are used for the first time.
3. Unlike other approaches to data quality solutions, MDM uses a semantic method - when checking data quality, it is not necessary to first describe the type of data that the data quality management tool is working on. The tool itself determines the object (and the probability) it is based on the content - semantics - of the data object, and chooses the appropriate action based on these findings (and defined business rules).
4. Data quality MDM solutions not only corrects bad data but also enriches the appropriate data available in other (internal and external) systems.

It should be noted that the solution thus described constitutes the ideal and most advanced situation. As in all other areas and within the MDM, we can identify different developmental stages differing in approach, complexity of solutions and outcomes. The diagram below shows four MDM development stages and their differences.


Gartner MDM Maturity Model
Increasing Maturity of MDM
Unaware
Problem? What problem?
Aware
No vision, but yes, we do have a problem...
Reactive
No vision. Firefighting is the answer!
Isolated, bottom-up approaches
Proactive
OK. Let's do something (at the silo level)
Silo-orientated solutions
Managed
A unified vision emerges (with high level sponsorship)
Enterprise-wide MDM program
Optimized
MDM is the way we do things around here
Managing master data as an asset
Continuing to learn and improve
1
2
3
4
5
6
Level of MDM Maturity

Obrázek 15: Vývojové úrovně MDM

Master Data Management can never achieve the desired results and will never be effective if only one of the two key folders is implemented. Technological solutions without business processes and management will only bring short-term results, as they will not eliminate the causes of the problems, but only their consequences. Conversely, only business changes resolve some of the causes, but they do not consolidate and clear own key data.

From the above, it is clear that understanding key data, centralizing, consolidating and improving data quality leads to a significant shift in the use of existing information systems. This is why MDM is one of the main directions of developmentas of Business Intelligence.

# 6 CORPORATE PERFORMANCE MANAGEMENT

Corporate Performance Management (CPM) aims to link BI to strategic goals of the organization and activities of individual employees. The CPM allows to integrate separate activities, processes, and tools in a single whole. CPM systems help employees use BI to change their behavior or perform activities aimed at meeting the organization's primary goals.

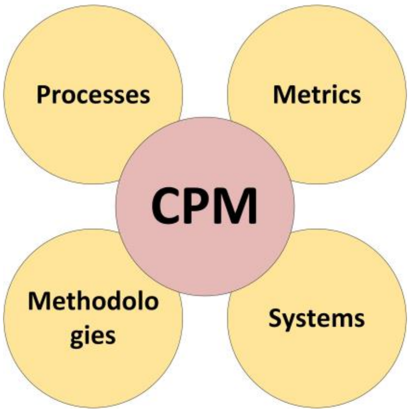


Fig. 6.1 The four CPM components

The Gartner Group defines the CPM as follows: **CPM is the concept of umbrella processes, metrics, methodologies, and systems used to monitor and manage the company's performance**.

CPM (Corporate Performance Management) is not the only term used. We can meet the concepts of BPM (Business Process Management) and EPM (Enterprise Performance Management), which are actually synonyms, but some vendors are sometimes using these term in narrowed meaning in conjunction with specific technologies.

There are four major CPM components:

- Processes - main processes of CPM main steps - strategy formulation, planning, monitoring and analysis, corrections.
- Metrics - key metrics (KPI - Key Performance Indicators). These metrics measure organization performance.
- Methods - Balance Scorecard, Six Sigma, ABC / M (Activity-Based Costing / Management) are typical examples of methodologies that are part of CPM.
- Systems - CPM includes planning systems, systems for calculating and displaying of defined KPIs in the form of dashboards and scorecards

BS provide a way of the assessing of organization performance and the documenting of a strategy by the measurement in four balanced dimensions: financial, customer, internal processes and organization development.

Basic steps of CPM

The basic principles of CPM functioning transforming the organization's strategy into action can be described in 4 steps:

1. Formulation of strategy
2. Planning
3. Monitoring and analysis
4. Correction

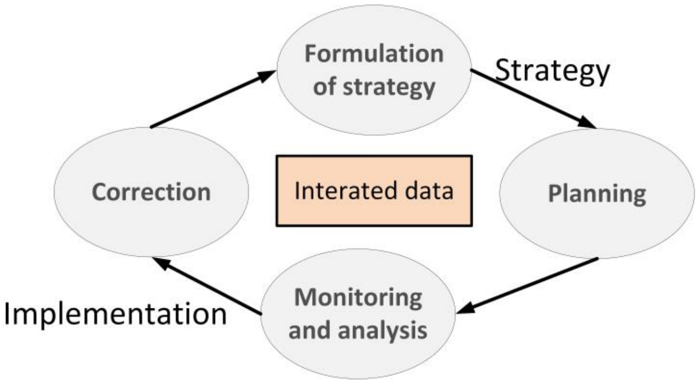


Fig. 6.2 Four steps of CPM

The individual steps are closely interconnected and form a closed cycle. The first two cycle steps - Strategy formulation and Planning support the development of the strategy. The remaining two (Monitoring and analysis and Correction) support strategy implementation.
Various methodologies, techniques and technologies are used in every step. All steps are supported by an integrated and consolidated database.

More detailed description of these steps:

Strategy formulation

What is our strategy for this year, in the medium to long term? Task standing before the top management of the organization. If it is not possible to measure, it is not possible to manage well. Measurement is a real managerial need, so it is necessary to define a set of metrics (KPI) to measure the fulfillment of the defined strategy and set their target values.
Just selecting and defining metrics is one of the most critical part of CPMs. Metrics (their calculation, monitoring, and analysis) rely on the entire CPM concept. In addition to the "classical" financial metrics, it is necessary to include non-financial metrics (such as the number of employees with a given qualification, the number of branches, sales volumes in units.) Examples of techniques and methodologies used in this step are Strategic Maps and Balanced Scorecards etc.
Organizations often fall into the trap of a large number of monitored metrics.
Best practices concerning metrics:
Optimum is je 12 – 25metrics;
Every metric must have its owner;
Metrics must be actionable, i.e., organizations must be able to influence all metrics by their activity.

Planning
The aim of this step is to create a plan of resources and activities (modification of existing processes, of new projects) to achieve a defined strategy. In line with the previous requirement to include non-financial metrics, the planning process must also include non-financial planning.
Planning is a collective activity in which the largest number of employees in the organization needs to be efficiently involved. In addition, this process is continuous where the plan responds to current developments and trends.
An important side effect of planning processes is communication over strategic goals of the organization and individual targets of individual employees. The result is the sharing of values and goals and the direction of the organization in one target.
However, the current practice is different for a number of organizations. The planning process is a bureaucratic exercise that brings little value to the organization. In addition, it is conceived as a one-time event, the approved plan is seldom changed, regardless of the real situation and further developments.
The use of scheduling tools can make the entire planning process much more effective. They can automate past manual tasks associated with refining, editing, and consolidating spreadsheet plans. The result is a reduction in the workflow of the entire planning process and more space for truly creative planning activity. They significantly reduce errors through defined control mechanisms. They allow fast "what-if" analysis.

Monitoring and analysis
How do we stand in comparison with the plan? What are the causes of the current state? What is the development trend? We look at the past, compare the plan with reality, analyze the causes, we estimate further developments. We are talking about the step that is within the CPM the most typical BI field. We work with data, acquire information about metrics, and analyze information in context.
To do this, we need BI infrastructure, BI tools, and especially people who can work with BI tools.
The success of the CPM concept depends most on people, at different levels, with differently defined goals. Establishing a transparent incentive system that the organization's employees are delivering on the set goals is one of the critical assumptions for the success of the CPM.

Correction
What needs to be done to meet the strategy and set goals? We are talking about the most important step in the entire CPM cycle. If we come to the stage when we know that the situation does not develop according to our ideas and when we understand the causes of this state, we must act.
The scale of activities is broad at this stage, ranging from immediate operational measures to reprogramming - for example in the context of developments in international markets or the fall in exchange rates. Plan corrections, in most cases, relate to the operational level, interventions at the strategic level are rather an exception.
The ability to close the entire CPM cycle and the ability to revise and prepare a further plan represents a competitive advantage that supports organizational success within changing external and internal conditions.

# 7 CORPORATE REPORTING

From the perspective of every modern successful company, key concepts are productivity, profit, quality of service, customer satisfaction, product success, market share growth, and so on. However, the difficult task is to quantify and evaluate these concepts in specific situations. But the quality measurement of company success is an inevitable prerequisite for its competent management.

Corporate Reporting covers all areas of information interpretation needed to support decision making within corporate communications and communication infrastructure. This may include:

- Visualization of measurement of achievement of operational and strategic objectives of the company;
- Presentation of key indicators and trends;
- An overview of the success of products and services;
- Corporate and in-house financial reporting.

## 7.1 The need for reporting systems

The need for reporting systems stems from the specific needs of the company. We preset here a few examples of cases where it is advisable to think about implementing the reporting system:

Example 1: Employees spend a lot of time manually creating analyzes and reports for company management. Instead of devoting their expertise to activities linked to the use of information and knowledge already acquired, they always repeat the same manual - and often frustrating - activities related to the collection, processing and formatting of data into the resulting form.

Example Two: Due to excessive time and effort, it is not possible to produce the more sophisticated or more detailed reports.

Example Three: Creating reports and analyzes and using them so far is inefficient and hinders optimization of business management.

Example Four: An existing information system does not allow a historical view of information, it is not possible to compare individual periods, to determine trends and to evaluate the state existing at a given date.

A or otherwise: A typical example can be a business company report that evaluates the success of product sales. In one report you can see data on the number of pieces sold, the turnover, the amount of inventory, the goods delivered, the complaints and the turnover, the category of products over their subcategories up to the level of the specific products. All this for the specified period and divided by region of sale. Additionally, with growth / decline information over the previous period. This is a basic reporting need. Such a report is manually configurable with difficulty or is almost impossible, but it is one of the standard tasks for the reporting system.

## 7.2 The description of the term reporting

The term reporting is a visualization of information. We can also say that this is a process of transforming data into information and knowledge. This process may be a simple task according to a particular situation, but it can also be a complex solution. In the case of comprehensive corporate reporting implementation, it may be the use of reports starting with simple product sales reports and ending with comprehensive IFRS or GAAP international IFRS statements. Individual types of reporting are characteristic of the way, how they are created, utilized, and added. We will introduce the main ones.

Static reporting

This type of reporting is particularly useful for visualizing information of a standard structure and appearance with almost invariable input parameters (basically, only the parameterization of the cuts or filters - e.g., a report for a selected period) can be used. It is well suited for financial reporting (e.g., consolidated quarterly reports), product sales reports, or automatically sent by emails (for example, the top 100 most successful products for the past week sent as an excel document) etc. The condition is that a defined structure report was accepted by everyone who would use these reports

Dynamic reporting

Unlike static reporting, user of a dynamic report can influence the content and form of a report by entering input parameters, or even alter the entire structure of a report by selecting different dimensions. This type is suitable for reviewing pre-unknown time periods, product and customer categories, and, if necessary, partially influencing the design of the report itself.

Ad hoc reporting

Users who do not even have one of the types of reports mentioned above have the opportunity to create their own report that will match their specific needs at an instant - an ad hoc report. Ad hoc reports fit in situations where it is difficult to determine in advance what content and form the report should meet, if any, this information is not yet known. After creating an ad hoc report, the user can decide whether to complete or delete their purpose, or store the report definition for later reuse, creating a static or dynamic report from an ad hoc report.

The main advantages of abovementioned types of reporting:

- Static Reporting - The user gets one-click information.
- Dynamic reporting - the customizing of the report to the needs of a particular user, browsing and viewing of information from different angles.
- Ad hoc reporting - independence from developer of reporting systems (the user can create the report himself) and the ability to create a report as soon as the need arises, and all the information needed to define the report will be known.

Decentralized reporting

Decentralized reporting is characterized by the fact that it is based on data structures that do not necessarily have to comply with data integrity standards. The advantage of such reporting is the relative speed of its implementation. There are also many disadvantages of this type of reporting that can overcome the benefits over time and with the growth of the company.

The main drawbacks are:

- Problems with a non-uniform version of the truth (different reports show different information, which can result in a loss of user confidence in the system and thus its failure.
- The growing demand for maintenance of such reporting solutions.
- Problems in extending solutions with new features and modules.
- Lack of clarity of reports for users.

Decentralized reporting is typically characteristic of emerging and small companies that need to quickly implement less complex solutions directly above production systems.

Centralized reporting

Centralized reporting is characterized by differences in several areas over decentralized reporting:

- Access to reporting is understood strategically with clearly defined goals and procedures.
- Input data structures are integrated into (corporate) data repositories.
- Reports are consolidated at a corporate (or at least a divisional) level.
- The reporting solution is planned and phased.

The benefits of centralized reporting are:

- One version of truth of information through a central solution.
- Comprehensive functionality.
- System development and maintenance.
- Stability and credibility of the solution.

## 7.3 Reporting systems – design a implementation

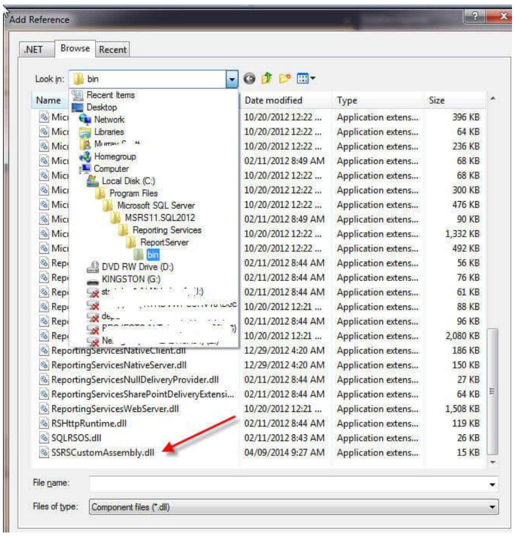
A company that feels the need to implement a reporting system should first define key points:

- What are the objectives of the reporting system and what are its expected benefits.
- Which areas should be covered by the system.
- Who will be the internal owner of the solution and who will be manager.
- What character (centralized / decentralized) will a solution have.
- What is the required / acceptable timeframe for implementation of the solution.
- What is the budget for implementation and maintenance of the system.
- Whether implementation will take place only with internal resources or whether some form of outsourcing (through development to complete outsourcing) is selected.

The very methodology of designing and creating reporting solutions depends on the definition of the key parameters that the company needs to know about the value of its performance. In general, however, we can define the crucial phase of creating such systems.

In the first stage, this is, of course, an analysis and specification of the information requirements. Next,  a specific design of technical architecture follows, which is often relied upon to select a specific software platform and products for solution implementation.

After approval by the customer, the technological work involves the physical design of data models, data streams and their connection to processes. The resulting implementation and pilot operation of the system only hinders the development of standard (static) reports and templates for dynamic reports.



Look in: bin

| Name | Date modified | Type | Size |
|---|---|---|---|
| Mic | 10/20/2012 12:22 ... | Application extens... | 396 KB |
| Mic | 10/20/2012 12:22 ... | Application extens... | 64 KB |
| Mic | 10/20/2012 12:22 ... | Application extens... | 236 KB |
| Mic | 02/11/2012 8:49 AM | Application extens... | 68 KB |
| Mic | 10/20/2012 12:22 ... | Application extens... | 68 KB |
| Mic | 10/20/2012 12:22 ... | Application extens... | 300 KB |
| Mic | 10/20/2012 12:22 ... | Application extens... | 476 KB |
| Mic | 02/11/2012 8:49 AM | Application extens... | 90 KB |
| Mic | 10/20/2012 12:22 ... | Application extens... | 1,332 KB |
| Mic | 10/20/2012 12:22 ... | Application extens... | 492 KB |
| Rep | 02/11/2012 8:44 AM | Application extens... | 56 KB |
| Rep | 02/11/2012 8:44 AM | Application extens... | 76 KB |
| Rep | 02/11/2012 8:44 AM | Application extens... | 61 KB |
| Rep | 10/20/2012 12:21 ... | Application extens... | 88 KB |
| Rep | 02/11/2012 8:44 AM | Application extens... | 96 KB |
| Rep | 10/20/2012 12:21 ... | Application extens... | 2,080 KB |
| ReportingServicesNativeClient.dll | 12/29/2012 4:20 AM | Application extens... | 186 KB |
| ReportingServicesNativeServer.dll | 12/29/2012 4:20 AM | Application extens... | 150 KB |
| ReportingServicesNullDeliveryProvider.dll | 02/11/2012 8:44 AM | Application extens... | 27 KB |
| ReportingServicesSharePointDeliveryExtensi... | 02/11/2012 8:44 AM | Application extens... | 64 KB |
| ReportingServicesWebServer.dll | 10/20/2012 12:21 ... | Application extens... | 1,508 KB |
| RSHttpRuntime.dll | 02/11/2012 8:44 AM | Application extens... | 119 KB |
| SQLRSOS.dll | 02/11/2012 8:43 AM | Application extens... | 26 KB |
| SSRSCustomAssembly.dll | 04/09/2014 9:27 AM | Application extens... | 15 KB |

File name:

Files of type: Component files (*.dll)

Fig. 7.1 SQL Server Reporting Services

Part of the design of the reporting system should also be a security policy. Thus, the exact definition of who has access to reports, who can, and can not, create their own reports, at what time, how often, and in what form, the reports / information for that user should be accessed.

## 7.4 The functionality of reporting system

Reporting systems have gone through modernization in connection with gradual developments in this area, with new requirements being placed on it in connection with the development of BI. Systems have given users more and more benefits. Some reporting requirements remained, others arised at the same time with innovations in business, industry and IT.

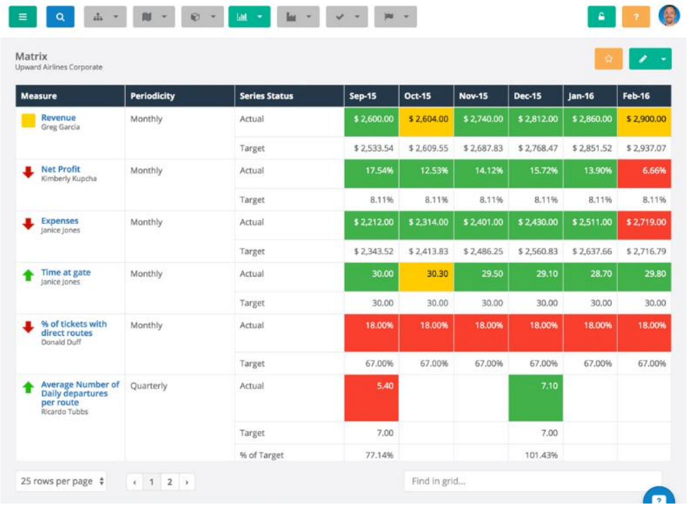
The current fundamental reporting needs can be characterized as:

- Centralized reporting solutions (corporate intranets / web portals).
- Efficient reporting solution (fast development at low cost and high added value).
- Integration of sophisticated solutions into reporting (reporting over data in OLAP databases, interaction with datamining applications).
- Integrating reporting with regular office applications (reporting within corporate portals, spreadsheets, and text editors).
- Access to reports using a simple user interface of an internal browser from anywhere in the world.
- Export the report to standard applications for further processing and sharing.
- Managing access to reports and protecting sensitive information that reports may contain.

As it can be seen in the list of current needs of reporting solutions, it is a comprehensive portfolio of functionalities, a more detailed description of which would be sufficient for a separate publication. Therefore, we have chosen one of the aspects of modern reporting on which we can define current needs. This aspect is the integration of reporting and data in OLAP databases.

OLAP (On Line Analytical Processing) databases (also referred to as multidimensional databases or cubes) offer special functionality compared to standard relational databases. This functionality means the following aspects:

- A comprehensive, unified data model that works with concepts such as the number of pieces sold. Net turnover, Product, Product Category, Month, Last Year, Customer Region, Supplier Name. Campaign name, etc.
- Pre-defined unified business logic, such as the method of calculating the price of a product without VAT, the average turnover of inventories.
- Pre-aggregated indicators, such as the sum of turnovers for the Ceske Budejovice region in February 2018 by individual customers.
- Additional predefined logic and functionality, such as Key Performance Indicators (KPIs), multi-language support, definition of follow-up actions, and more.



Matrix
Upward Airlines Corporate

| Measure | Periodicity | Series Status | Sep-15 | Oct-15 | Nov-15 | Dec-15 | Jan-16 | Feb-16 |
|---|---|---|---|---|---|---|---|---|
| 🟨 Revenue<br>Greg Garcia | Monthly | Actual | $ 2,600.00 | $ 2,604.00 | $ 2,740.00 | $ 2,812.00 | $ 2,860.00 | $ 2,900.00 |
| | | Target | $ 2,533.54 | $ 2,609.55 | $ 2,687.83 | $ 2,768.47 | $ 2,851.52 | $ 2,937.07 |
| ↓ Net Profit<br>Kimberly Kupcha | Monthly | Actual | 17.54% | 12.53% | 14.12% | 15.72% | 13.90% | 6.66% |
| | | Target | 8.11% | 8.11% | 8.11% | 8.11% | 8.11% | 8.11% |
| ↓ Expenses<br>Janice Jones | Monthly | Actual | $ 2,212.00 | $ 2,314.00 | $ 2,401.00 | $ 2,430.00 | $ 2,511.00 | $ 2,719.00 |
| | | Target | $ 2,343.52 | $ 2,413.83 | $ 2,486.25 | $ 2,560.83 | $ 2,637.66 | $ 2,716.79 |
| ↑ Time at gate<br>Janice Jones | Monthly | Actual | 30.00 | 30.30 | 29.50 | 29.10 | 28.70 | 29.80 |
| | | Target | 30.00 | 30.00 | 30.00 | 30.00 | 30.00 | 30.00 |
| ↓ % of tickets with direct routes<br>Donald Duff | Monthly | Actual | 18.00% | 18.00% | 18.00% | 18.00% | 18.00% | 18.00% |
| | | Target | 67.00% | 67.00% | 67.00% | 67.00% | 67.00% | 67.00% |
| ↑ Average Number of Daily departures per route<br>Ricardo Tubbs | Quarterly | Actual | 5.40 | | | 7.10 | | |
| | | Target | 7.00 | | | 7.00 | | |
| | | % of Target | 77.14% | | | 101.43% | | |

Fig. 7.2 An example of binding of report with scorecard results

The benefits of reporting over OLAP databases are:

- The business users understand data model, they can therefore generate reports themselves.
- Even complicated reports, which, for their calculation, need to add tens of millions of records, are a matter of seconds.
- Visualization of KPIs and trends in reports, drill-down, setting up other applications on a report (for example, clicking the vendor's name opens a map with its location) and more.

# 8 BI ORGANIZATION

The Business Intelligence Competence Center (BICC) is the name of the organizational unit providing internal services in the BI.This chapter is dedicated to the BICC. Gartner Group introduced the term BICC a few years ago, and the concept quickly took hold in BI area.

If proper organization of BI is missing, then the following issues may occur:

- Absence of centralized management and solution BI
  There is no clear long-term and often also medium-term development of strategy for BI. The solution of operational problems and requirements dominates. It is unclear both from the technology point of view and from the point of view of realizatin of requirements the way in which an organization will solve these problems.
  The frequent manifestation is that various organizational units provide solutions to their requirements individually, using different technologies (mostly for reporting) and different procedures (there is no or no uniform methodology). This then leads to a frequent situation where there is no overview of the reports used, their purpose, the exact interpretation and the way they are used.
- Missing capacities to meet user requirements
  A frequent picture - the user is unhappy because he just learned that his request for a new report will be solved sometime in 4-5 months. That the marketing campaign should be ready at that time? What if there are no spare capacities?
  Even worse case - the user does not even know who to turn to. It is not defined who is responsible for preparing the reports.
- Insufficient BI methodology
  BI methodology is often degraded to just a request form for new requirements. The entire lifecycle of the implementation of BI requests and projects is not covered. In particular, there is no definition of how to acknowledge receipt of the request from the contracting entity or the BI deployment procedures. If the BI methodology already exists, there is a lack of management and systematic development of the methodology.
- Problems in communication and weak knowledge management
  In many organizations, we can meet a lot of technocratic BI concepts. The department responsible for BI is integrating new data sources, developing ETLs, creating a report based on precise assignments, and managing the technical infrastructure. However, there is insufficient communication with advanced users. Such a department is then perceived as a IT department or its part.
  There is often insufficient knowledge sharing within the organization. There is no system in place where information is stored, there is no environment for forwarding and sharing information. Missing regular BI education.
- Missing access control for data quality
  Data quality management is the most common problem. Usually, it is not clearly defined who is responsible for this area. No system of data quality management is implemented.

These five typical problems, respectively. some of them, are at the beginning of the initiatives leading to the creation of the BICC. These problems are impulses that drive management to interest in the BI area and adopt a solution that implies a qualitative shift.
The long period of implementation of new user requirements and unclear competence are in practice one of the most common and powerful arguments for the introduction of BICC.

A proven model for implementing user requirements in BI is a two-level model. The BICC department (2nd level) ensures the implementation of demanding requirements and provides expert consultations. An advanced user group (1st level) provides services (reporting, basic analysis) to its departments.
Advanced BI users (Advanced / Power Users Group) are representatives of the various departments in the organization with the necessary knowledge and skills. In comparison to ordinary users, they have above-standard equipment from the perspective of BI tools (reporting and analytics tools) and take part in specialized BI training.

## 8.1 BI organization design

However, the "simple" response to the problems mentioned above is not the main reason for the creation of specialized organizational departments, the main argument is hidden in the very essence of BI. Efficiently functioning BI represents a combination of three worlds:

1. Knowledge of the organization's business area (business)
2. Knowledge of analytical procedures and techniques (analysis).
3. Knowledge of the necessary IT tools and technologies (IT / IS).

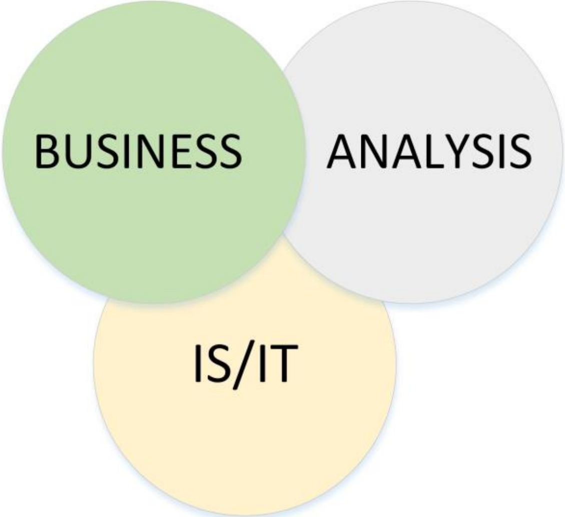


Fig. 8.1. BI connects three areas

In a common organization, this knowledge is distributed and there is no single place to connect these three areas (see Fig. 8.1). We have analysts who handle excellent analytical tools they need, but have trouble negotiating with a business people. We have business analysts who do not have the necessary knowledge of tools and of used data structures. The result is the lack of full potential BI, i.e., business does not get as much from BI as it could get.

The idea behind the BICC concept is simple - to achieve a significant synergy effect by linking the three areas to a single organizational unit.
In addition to solving or reducing the size of the problems mentioned in the previous paragraphs, it is possible to summarize the main benefits of introducing BICC into the following five points:

- Quality support for users through knowledge of business, data, analytical procedures and tools.
- Protection and maximization of the value of investment in technology (technical infrastructure, database, development environment, reporting and analytical tools).
- Integration and consolidation of all processes and activities that allow analytical processing of data needed to support business.
- Total reduction of risks associated with the implementation and implementation of projects in the BI.
- Promoting knowledge management for BI within the organization (building, maintaining, and sharing knowledge about procedures, data, technologies).

## 8.2 BICC workload

The BICC workload should include responsibility for BI strategy and responsibility for BI procedures and methodology, for the implementation of user requirements, for training and support of users and advanced users. Workload of a BICC can be divided into six basic areas:

Obrázek 27: Funkční oblasti BICC

More detailed description of single functional ereas (F1 – F6).

(F1) Strategy

- BICC is responsible for the BI strategy in the organization. The strategy builds on the overall strategy of the organization and on trends and recognized best practices in the BI field.
- BICC coordinates all BI activities in the organization and cooperates with similar specialist departments within the parent company.
- The BICC remuneration model should be based on KPIs (Key Performance Indicators) tied closely to business results (meeting primary organization goals).

(F2) Methodology

- BICC is responsible for preparing, developing, enforcing and ensuring compliance with the BI methodology (methodology, standards, techniques and procedures).
- BI methodology covers the entire basic implementation of BI process: Requirement, Preliminary Analysis and Prioritization, Detailed Analysis, Implementation, Testing, Acceptance, Deployment and Publication, Support and Operation, Cancellation (for example, of report).
- BICC is responsible for monitoring and managing Service Level Agreements (SLAs) for the BI field (e.g., guaranteed time to open the data warehouse for users, availability of defined reports).

(F3) Communication and training

- BICC ensures systematic work and communication with business users (regular meetings, satisfaction surveys) and advanced users (regular professional workshops, training, active involvement in knowledge management).
- BICC provides organization and implementation of professional training; either by itself or externally.
- BICC is an internal PR for the BI area and a regular publishing of B1 information (intranet, company periodical).

(F4) Implementation of requirements
The collecting, the prioritizing and the analyzing of user requirements in collaboration with business representatives is one of the core activities of BiCC:

- BICC participates in the preparation of tenders, initiates and implements projects in BL
- BICC provides analytical support for top management, demanding analytical tasks, and generally outputs with high added value.
- BICC ensures the implementation of user requirements (ad-hoc reporting and analysis) or with the support of advanced users.
- BICC also can co-operate with external suppliers to meet the requirements.

Místem častého Jiskření" mezi BICC a uživateli je stanovování priorit realizace jednotlivých uživatelských požadavků. Je to místo, kde se nelze zcela vyhnout zavedené firemní politice (různě nastavené vztahy, různá síia manažerů). Manažer zodpovědný za činnost BICC by měl ve vlastním zájmu co nejdříve iniciovat tým (BI Steering Committee) složený ze zástupců managementu a businessu, jenž bude rozhodovat o konfliktních situacích při stanovování priorit Manažer BICC bude tým pravidelně informovat o činnosti BICC (plnění KPIs).

(FS) Data management

- BICC secures the acquisition of new data sources and their implementation and integration into existing data structures (DWH, data marts). In some cases, it takes over the management of selected data sources.
- BICC is responsible for data quality management of Bi solutions. It ensures the definition and development of data quality management methodology, the selection and implementation of data quality support tools.
- BICC is in charge of managing metadata, including managing user and technical metadata, and managing metadata technology infrastructure.

(F6) Support

- BICC provides technical and user support for BI solution and reporting (2nd level support).

The BICC does not necessarily have to be built as a line structure, but it can be conceived as a virtual organizational unit with a matrix structure. Apart from the close management of the BICC, other BICC representatives are integrated into some line departments where they also carry out activities within the BICC's competence.
BICC is mostly part of the organization's static linear structure.
The BICC department is dedicated to data integration and ETL development - Data warehousing. Separate part is the Data Quality Department. While this department is often the weakest, its significance is crucial to the success of BI.
The Reporting and Analysis Department is a "end-to-end" user interface that is the department that communicates with users most intensively and ensures that user requirements are met. Users evaluate the benefits of BI according the functioning of this department.

The following figure shows one of the possible organizational arrangements of BICC:

Figure 8.2 Example of an organizational BICC arrangement

Up to 20 different roles can be counted in the BICC. There are, however, several of them that are key to BICC's performance and they can not miss. The following table lists the role, department, main role roles, and brief characteristics of requirements to the role.

| Role | Department | Focus | Main requirements |
|---|---|---|---|
| BiCC Manager BI | | BI Management Strategy | • Conceptual Thinking<br>• Knowledge of principles, procedures and methodologies of BI<br>• Business know-how (processes, services, products, concepts)<br>• Basic knowledge of used technologies and data structures |
| Analyst<br>(Business Analyst) | Reporting and Analysis | User Requirements, Analysis and Reporting | • Analytical capabilities Knowledge of analytical and reporting tools<br>• Detailed knowledge of data structures (data warehouse, marty data, data structures for reporting)<br>• Data modeling, time |

| | | | |
|---|---|---|---|
| | | | management tools, SQL<br>• Excellent business know-how (processes, services, products, concepts) |
| Data analyst | Data warehouse | Data warehouse, data integration, design | • Deep knowledge of data model and warehouse architecture<br>• Good knowledge of (selected) data sources<br>• Data modeling, time management tools, SQL<br>• Analytical capabilities |
| ETL Developer | Data warehouse | ETL design and development | • Knowledge of ETL and development tools<br>• Knowledge of used database technology Detailed knowledge of selected data structures (data warehouse, data marts) |
| Data Quality Specialist | Data quality | Data quality | • Organizational skills<br>• Basic knowledge of process control<br>• Knowledge of principles and concepts of cleaning / data quality control<br>• Basic business know-how (processes, services, products, concepts)<br>• Knowledge of used technologies and data structures |

Table 8.1 The characteristics of main roles in BICC

# 9 DATA WAREHOUSE

Let us summarize the facts that have been mentioned so far, leading ultimately to the creation of a new type of databases - data warehouses (DS).

- Various enterprise data is gathered in the databases of different information systems,
- This data data is archived, poorly utilized,
- Most databases are corporate, the development of their IS is basically closed,
- There is a need to find the ways to manage companies better, objectively to optimize enterprise management,
- Statistical methods that can be used for data analysis in databases were developed,
- New economic instruments, methods for management and optimization are available,
- Methods for gaining knowledge from data were developed.

The idea was to use data from both enterprise and external databases, from archives and other sources for analysis and other processing, i.e., for long-term evaluation of past performance, prediction, support strategic decision making of marketing, management.

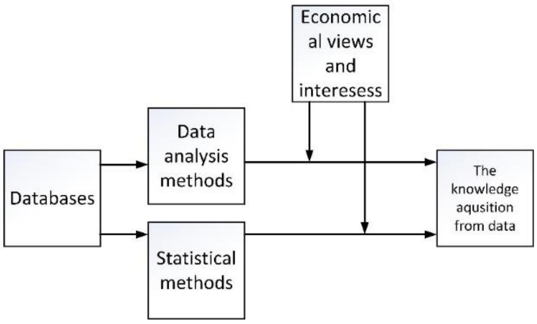

Economical views and interesess
Data analysis methods
Databases
Statistical methods
The knowledge aqusition from data

Figure 10.1 The schema of a process of data acquisition

Initially, these analyzes were naturally carried out as a further functional structure above operational databases and its information system. Soon, however, a number of drawbacks of this solution have emerged and idea was to separate data for analyzes from the operational databases. The idea was to realize a kind of parallel between material production and information production. Material products are made from the source (raw) material, transferred to the warehouse, where they are packed, and then they are sold. This schema is:

Production - Warehouse - Sales

Similarly, source data from databases (~ source material) is processed into comprehensive and global information (~ products) and stored in a separate database, called data warehouse. Then it is tailored to the needs of customers (they are packed in an appropriate, comprehensible form) and sold. The individual stages can be marked as:

OLTP - DW - OLAP

OLTP (On Line Transaction Processing). Usually, data in enterprise information systems (e.g., accounting or logistics systems) are routinely performed with daily transactions. These data are called operational or transactional data. Respective data processes in relevant information systems are also called the On-Line Transaction Processes (OLTP). OLTP information systems serve primarily to perform routine work.

DW (Date Warehouse) is a data warehouse. A database that contains data from all enterprise OLTP information systems, a place where calculations also are made and pre-calculated results are stored.

OLAP (On Line Analytical Processing) is a tool for the performing analyzes, selecting and presenting new information, seeking useful knowledge.

Likewise, classic operational databases, a data warehouse (what is also a database) works with its transactions and proper software systems.

## 9.1 The comparison of operational databases and data warehouse

Operational databases (OLTP database) are designed optimally for maintenance large amounts of data and for selective retrieval of information. That means without redundancy, with the helpof clearly defined rules. The rules are called normal forms, and design algorithms are known as a design of a well-structured database. Database with all relations (tables) at least in the Third normal form is called a standardized database (see the course Database systems).

The reasons for normalization are also known. Normalized tables do not contain data redundancy and therefore no inconsistency with possibly other accompanying phenomena as anomalies at insertion or anomalies when deleting the data. Operational standardized databases serve to everyday enterprise transactions (accounting, logistic operation, etc.). More specifically, OLTP operating systems use these operating databases. Mainly administrative staff works with these databases through information systems (the tasks like to find unpaid invoices, find material with current quantity less than required minimum, etc.). Data placement is therefore usually optimized according to the needs of operational departments. Data from operational databases is regularly archived, however this archived data is not used on regular basis.

### Using OLTP to support decision making

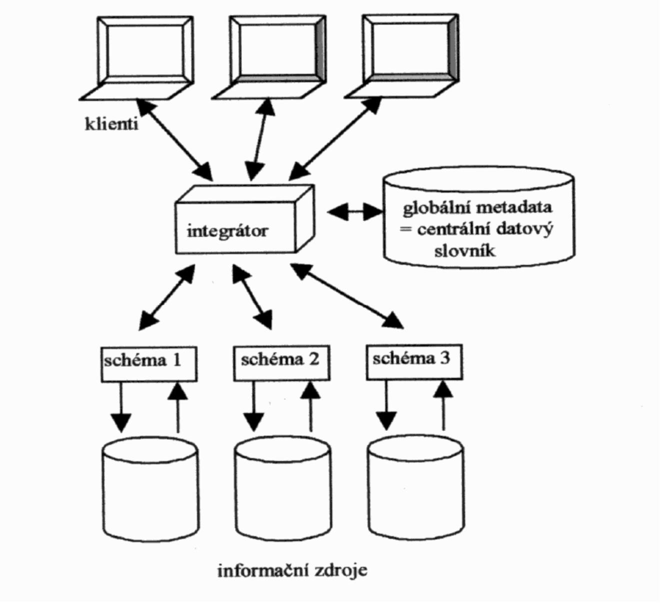
The use of the data from operational database (also from multiple sources) and from databases with archive data to support decision-making brings some problems.

In a traditional integration, it is possible to keep partial database schemas and using them to create one global database scheme. This is an approach known from the theory of distributed databases (see Figure 10.2). This approach is driven by user queries. In order to meet the query, appropriate information sources are found in the global scheme, they are processed then, and transmitted to the user. It's slow and difficult process, it means complex data collection through several sub-schemes for each query. It is also inefficient because these actions are repeated for queries of the same type. This requires large time and has big capacity requests for operational database.

Management requires a different type of queries than ordinary users in operating databases. Management is usually interested not to get report about individual records but is interested in global information. Management requires intensive questions (how many computers we have delivered to the Prague region in the last quarter, was the profit bigger compared to the same period of the last year, how much bigger about). This information is not directly contained in a database. It is necessary calculate it from the source data. Additionally, these questions are not predefined (they are very immediate varied and very changing according to management needs) and therefore their processing is not directly built into respective information system.

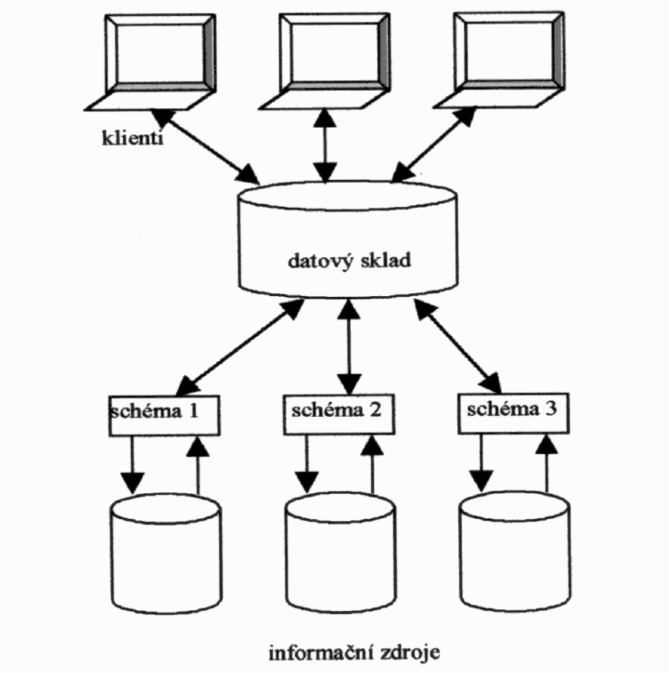
Even when it is possible to formulate any SQL queries over operating database, some pieces of information data are unachievable through SQL queries in transactional systems. Sometimes it is necessary to manually combine the results of SQL queries from various databases for achieving the desired results. Hence, the following issues arise:

- Data to be used in analyses must be flawless - consistent, complete, often transformed, to different levels aggregated, selected from different sources - also from archive and external data,
- In addition to its own data, it is also necessary to record their definition, structure and other information about the data (the so-called metadata)
- Application programs must have a completely different character than the operational algorithms,
- Algorithms performed over data are time-consuming. They thus burden the operation of operating information systems, they cause long delays in making complicated queries, they compete for computer resources between transactional queries and decision support questions,
- Data should be stored on a medium in other way than operational data are stored
- On the other hand, it is not necessary for the data to be current, the nature of the analysis is usually in searching for generally valid facts from longer-term data,
- Previous access to data is complex, it is not a user-friendly interface to the database SW (for requesting either an existing application or SQL Query Language, i.e., it is suitable only for operational work outputs; there is no automatic formatting, graphical visualization means for unexpected queries).



## Characteristics of data warehouses

These aforementioned reasons have led to a new solution, to define a new stand-alone database of a higher level - the data warehouse and the decision support system (DSS) over it. Data from the operational database and from archives and other sources are stored in this data warehouse. Data warehouse (central database) is so pre-engineered to be optimally available for OLAP and Data Mining analyzing algorithms. Preprocessing includes checking for consistency and completeness of data, transformation (into the same measurement units, same data formats, etc.), aggregation (temporal, spatial summation, etc.), metadata definition, appropriate organization of data storage on the medium.



From the database point of view, the data warehouse (DW) is created by integrating heterogeneous data, hence its definition.

**Definition:**

Data warehouse is an integrated, entity-oriented, permanent and time-resolved collection of data, organized to support management needs.

### Basic characteristics of data storage in data warehouse

- Storage of data is separate so that their demanding processing does not interfere with the operation of operating systems.
- Data are not only from the operational database, but also from archives, other external sources and reality (not only custom data, but general data converted into strategic information).
- Data is integrated one time and stored in a DW with its own metadata and with own organization of data. An advantage is a faster response when querying, without loading operational databases.
- Data are pre-prepared - transformed, unified, checked, aggregated, in format suitable for decision support systems.
- Another option is (batch) data refill, summary, historical data record.
- Data is not categorized according to the requirements of individual departments, as for operational databases for day-to-day transactions, but are concentrated (by type) monitored objects and management needs at different levels. Their structure, format and semantics and physical storage are optimized for the needs of the entire business and for OLAP tools.
- Specific analyzes and their presentations can be now performed over DW. DW are optimized for direct querying by selected types of questions (for OLAP).
- DW with its tools (DSS) has a ability to respond to questions ad hoc = "brought by life" in a short time. It is not necessary to prepare them in advance, pre-program them, but these questions can be important for decision-making business strategists. It is said that the information that comes late is no longer information.
- DWs are used to present data, hypothesis testing, and discovering new knowledge (data mining) through specialized software.

### Differences between IS and DW design

Process of analysis, design and implementation of data warehouses and systems for subsequent analysis and presentation their results are significantly different from the analysis and implementation of operational ISs.

In contrast to traditional IS, other technologies need to be used

**for DW design (so-called multidimensional modeling)**

- Data is organized differently than in the classic databases (DB), it is integrated (unified from different sources), arranged in homogeneous views for subsequent evaluation. Because data is used for analyzes in different time periods, it has to create time series. For the needs of management, aggregated data is important, not detailed operational data.

**for preprocessing data**

- Transferring data from a base database to a DS is called a data pump. New data is added into DW in regular intervals, for example, outside working hours in order not to burden the operation of the operational IS, or to meet the immediate need on some analysis based on the use of DW.
- Integration, filtering, and transformation are part of the data addition. After data is complete, they are underdone to conversions in order to get proper derived and aggregated data.

**to save the data**

- Operational, detailed data is stored, as well as various aggregations into dimensions (time, geographical dimension, commodity, etc.). The data here is redundant, non-standardized in contrast to relational standardized data storage in standardized enterprise operating system databases;
- The basic difference is in the implementation: ROLAP (Relation OnLine Analytical Process) - the basis is a relational model without normalization; POLAP (Proprietary OnLine Analytical Process) - physical implementation of multidimensional database as a multidimensional cube with special access paths; MOLAP (Multidimensional Cube) - another name for POLAP

**for data processing**

- Some classic database operations are not used, e.g., insert, update, delete, data are uploaded on regular basis from other data sources, data is constant.
- The user interface is optimized for the non-professional user, user-friendly friendly querying, statistical analysis, data mining.

**for presenting the results**

- DW has tools for graphical, table and word presentation and data visualization, pre-processed aggregates and data mining data for reporting.

### Data marts

In terms of the DW implementation, the architecture may not be single physically centralized data enterprise-wide. DW can be either physically centralized or logically centralized and physically decentralized or finally physically distributed.

In addition to a complex DS stored in a single huge database, partially decentralized Data Marts (DM) - data markets or Operational Stores - can be implemented.

They are mostly used for special and more detailed analyzes, focusing on some aspects, for example marketing. Data markets or operational stores contain fairly fresh data and are used online analyzes where it is necessary to respond to real - time trends (e.g., preparation of raw materials for preparing meals at the facility depending on current demand to maximize time customer clearance).

### Presentation layer of DW (OLAP)

The data presentation is the third step. The data is processed, summed up so that the view of the data and the derived knowledge is understandable to the ordinary user. Statistical methods, modeling of trends, etc., are used to make information become transparent and understandable for users in terms of their queries.

Systems are optimized for fast data selection, summation, analytical processing of large volumes of data. The goal is to provide valuable information quickly from large volumes of data. The DW structure is presented to the user when querying, the user can query the information at different levels of aggregation:

- For detailed data,
- On summation tables.

Aggregate values can be pre-calculated at different levels of detail (sums of amount of sold goods per day, per week, per month, year, average sales for this period, average revenue for shop, district, county, state, etc.).

Of course, analyses must be easy and fast to switch between hierarchical levels - up i down, or displaying time or dimensional rows (the course of earnings for more months, for all shops, etc.). Horizontal and vertical procedures are called

- Drill-down
- Roll-up
- Slicing & dicing

### Brief description of processes and data flows

The following figure summarizes the entire process of creating, uploading, storing and using data.

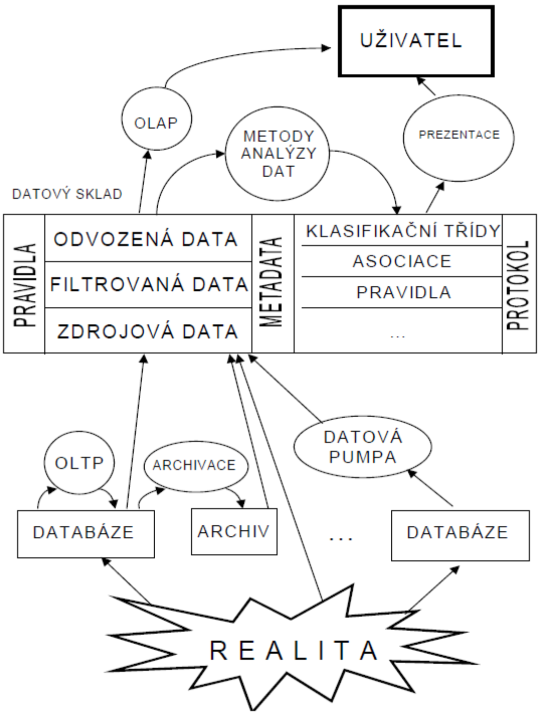


Figure 10.3. Schema of data flows within data warehouse and its surroundings

Source data is always taken from the real world. They are usually stored for operational reasons into databases usually controlled by the relevant information system. However, they may exist in others formats - in excel spreadsheets, text tables, just "on paper", etc. Data already inactive are usually stored in archives. If data in database is supported by an information system, we call this whole OLTP process. Such databases can be many, even if the data owner is the same entity - a company, organization, individual (the company usually operates several IS - human resources system, stock records system, logistic system, accounting system, etc.). If the owner of these data decides to use them further for example to management purposes, it is appropriate to build an integrated data warehouse.

## 9.2 Design of data warehouses

### 9.2.1 Phase of a design of data warehouses

The design of the database in the building of the information system includes 3 phases:

- Conceptual, design logical structure using ERD or a class diagram,
- Database, implementation of conceptual schema in a DBMS specific environment,
- Physical, database implementation.

On the other hand, design of a data warehouse inclused 4 phases:

- Conceptual
- Multidimensional
- Database
- Physical

We have already indicated that the structure of the data warehouse is different than the operational database. Especially in that it comprises a series předpočítaných (ie, redundant) data. Precomputation based mainly in calculating totals, averages counts many data (generally aggregated values), including several levels of detail, that is a kind of hierarchy of aggregation. These data need to store somewhere, suggest for them, the new data structure. It is the task of multidimensional modeling.

### 9.2.2 Conceptual modeling of DW

Conceptual modeling role

In the conceptual modeling of the data warehouse, the starting point is not the assignment of the user regarding his/her needs of data recording as in the IS. The situation is different and can be characterized by the following points:

- DW is based on existing database schemas, conceptual models of source information systems are being input for DW. The task is to create an integrated schema of the future DW based on many sources (database schemas). This is not just a union of source schemes, but the task of conceptual modeling is to decide which data will be part of DW and which is not.
- After selecting the attributes, it is necessary to define the rules for the data pump: how each attribute will be filtered, integrated, etc. when the selected attributes are transferred.
- Another task is the conceptual decision whether to create a single schema for single DW or one for each mart data (DM).

This process is labeled as ETL (Extraction, Transformation, Loading) process.

**Selection of attributes for the data warehouse (Extraction)**

The first task of conceptual modeling is to decide what all the data (which attributes) will be stored in data warehouse. Existing database database structures and, if appropriate, other appropriate data and assumed analytical tasks are the basis for this decision making.

Selecting attributes of DB for the DW means:

- To collect a list of source DB with their schemes (i.e., current operational DB, archives and other data sources which can be in text format, can be an excel table, and other files containing data obtained from other external databases, from the Internet, etc., if they can usefully complement and enhance DW);
- To select information from these data sources (i.e., tables, attributes) suitable for DW;
- To describe the atomic structure of the data, especially data copied from individual records source databases, but already integrated and transformed.

**An example:**

A company TRW selles fashion accessories. It has a databases twenty years old. It archives sales data monthly and it keeps them over 2 years (after the warranty expires). Over those 20 years, the company and its databases and information systems developed: the original system in MS FoxPro was replaced after 10 years by a system with another database MySQL. New software HELIOS with MS SQL database has been bought 4 years ago. Each system has slightly other database structure, uses other data types, and other units of measure for certain goods. The first database was developed on request. The design of the database has been changes several time, new attributes were added. Therefore, neither an archive table from different times in the same database do not have identical structure. The first task is to get the structure of these source databases, to compare them and to propose for them the single (integrated) structure so that all these resources can be used in the DW (an atomic data structure).

Solution:

The solution will be based on current operational structure of the actual database, because the data these will continue increase.

- Attributes that did not occur previously and are also important for the DS, will be included in the structure and older records will be marked by pump data missing.
- Attributes that previously occurred and now they do not accur will analyzed and it must be decided about their importance and if they will be include into DW.
- Some attributes can be excluded from the DW (e.g., supplier delivery note number - probably artificial key of this

supplier in database unimportant for analyses in DW, bank account number of supplier - probably a constant value for the DW also unimportant, etc.).

- Attributes with different units of measurement (for example, initially in pieces, now in 4-Pack, or originally in grams, now in kg etc.) will be marked. This proble will be solved in the data pump.

### Data pump (Transformation)

We have the atomic structure of a selected data DW, we chosed which attributes will be stored in DW. Further object is to define rules for the transmission, filtering and transform the source data into the DW or DM, i.e., to define data pump function. We assign to each source data an integration function to the DW. The simplest case is a mere copy values of data source into DW, eventually it is necessary to perform some of these operations: check, modification and transformation.

Functions: cleaning, integrating, transformation, it means that for each attribute is being performed:

- Check the accuracy of the data and eventually data cleaning, i.e., correction of errors: if the data in the source database is well controlled, this function is not required; if there there are erroneous data, are now two repair choices: either to detect the error and remove it with the help of data pump in the DW, or to cooperate with the administrator of the source DB and to supplement controls in the source IS; however, it always is necessary to fix erroneous historical or current data;
- The dealing with missing values - again it is necessary to decide between several options: unfilled data can be ignored, not to include it into DW; if this is an important information for the DW, we have possibility to add it from another source; where data is not completely empty and attribute is important for the DW, it is necessary to mark it with a special value.

  Example:

  The value -1 is used for missing data for the attribute that has non-negative values. This value is used in the analyses over the DW; The user is then informed enough about how much data is involved in the analysis because their averages, sums and other aggregated values would be distorted by different numbers used values.

- Solution of data with a constant value - if they have some attributes with a constant value, we will not use them in DW because they can not bring anything interesting. If the attribute is almost constant, we must consider the scope and the importance of its different values and decide about its inclusion in the DS.

  Example:

  Attribute country of origin of suppliers of goods from a total of 125 suppliers has only 4 with different country of origin than the Czech Republic; we will consider whether we are interested or will be interested in the future from where the goods and for how much money is imported and how this information will be evolving. As a result, we will decide for example that we will not to include this attribute to DW.

- Solution of inconsistent data values - the same values can be recorded by a user in various ways; the most common incorrectly entered data are textual data (name, address, names, atributes such as gender, marital status, etc.), but other types of data can also be incorrectly entered; the task is to unify the same reality data with the same DW data; it is usually necessary:
  - to define a convention for textual data such as names, names, addresses, ... (sequence, size font etc., see rules for entry in the data dictionary); if data transmitted to DW do not conform to convention, it is necassaryto modify their value.

    Example:
    The name of a representative of the company can registered as NOVAK Josef but sometimes as Josef Novak and also Ing. Josef Novak or Novak Josef, ing. or J. Novak, etc. Similarly, the company name is reffered as ABC spol. s r.o, or sometimes as ABC s.r.o., or sometimes as only ABC, etc.).

    Sometimes, it is necessary manual intervention in deciding whether it the same value.

    Similarly, it is necessaru to define the formats of some other data not only textual, but also of data in date or numeric format that can be entered in different databases in different ways, or possibly written by different users in different ways.

Example:

Typical example is the date of birth: in the format dd.mm.yyyy or mm-dd-yyyy or dd / mm / yyyy etc.

To define unit of measure for numerical data and, if necessary, propose conversion rates for older or external source data - units of measure can be different for each source;

- The solution of duplicate records: if the source database has not a control over duplicates then it is necessary to detect duplicates and to remove them, or to recommend further control in the future;
- Derived data: Some source data will be useful to derive additional data and store them, even if these data are redundant; with large volumes of data in DW, it would be a great delay first to always derive these derived data and then make the above them queries.

  Example:
  Typical cases are data derived date attributes - day, month, year, decade, quarter, day of week, etc .;

  Another example is a period between the time data, such as patient age at the time of hospitalization = number of years between the date of birth and date of hospitalization, etc.
- Timestamp of data is the last step of described process; each entry data has a time stamp of when it was upload into DW.

### Uploading data to DW (Loading)

Analysis of the data can be divided according to data sources into 2 parts:

- One-time data transfer from archives and other older or external data sources - data is integrated, transformed and transferred to DW. The situation may be complicated. It is necessary to integrate each source separately.
- From the current operational databases, new data increments will be transfered repeatedly - either a period is proposed, how often and when the data will be automatically transfered automatically, or the data transfer will always be executed according to the decision of a DW administrator. The period depends on the size of the increment and the importance of having DW data up to date.

  **Example:**

  With sales data, it will probably be important to track daily and immediate interest in individual goods;

  For medical data intended for research, it will be enough to add data 1-2 times a year; however, it will be appropriate to monitor data on a daily basis in the case of flu epidemic.

Altogether, the function of a data pump has three phases. In both cases (data from archives or data from the current databases), new data:

1. New data are filtered, integrateed, transformed into atomic data warehouse data; atomic data is a unified, integrated copy of source data with eventual exclusion of data unneeded in DW;
2. Integrated data is copied from atomic data to a new structure of a data warehouse;
3. Designed aggregations are calculated above the base data and are stored in the data warehouse.

Atomic data can remain in the DW or they are deleted throughout the process as intermediate result. We already know that queries over the DW is usually not related to individual records but to sum values.

**Data warehouses and data market (Loading)**

The part of the data transfer plan is the conceptual decision about the deployment of data in a data warehouse. It is not always necessary or expedient to store data in a single large database. It is also possible to create certain parts of DW that will serve to users from various departments:

Several possible solutions exit. There may be only one data warehouse, independent datamarts, or a combination thereof.

Among the data warehouse and data mart may be several types of interconnections:

1. Source data are loaded (integrated, filtered, transformed) separately into all data marts in which this information is needed. In this case, we have only separate marts, there is no integrated data warehouse. Advantages are in smaller dimensions of individual markets and faster response to users. Disadvantages are in redundant preprocessing and redundant data in multiple places, hence maintenance and consistency issues. If data replenishment is not uniformly controlled, but each marketplace controls replenishment itself, data may be diverted over time.

2. The source data is loaded into the data warehouse, where they are integrated. Their parts are then transferred to separate data marts. Benefits are in one-time and unified data processing, minor maintenance problems and quick access of individual users to smaller data. Disadvantages are the redundancy of the same information in many places (data marts). The filling of the central DW is called the primary process, the filling of the data marts the secondary process.

3. There is only a comprehensive central data warehouse without data marts accessed by all users accessing the same data. The advantage is a single pre-processing without further redundancy in data and central maintenance, a disadvantage is a huge data warehouse and thus lower performance and flexibility, and a slower response to many users.

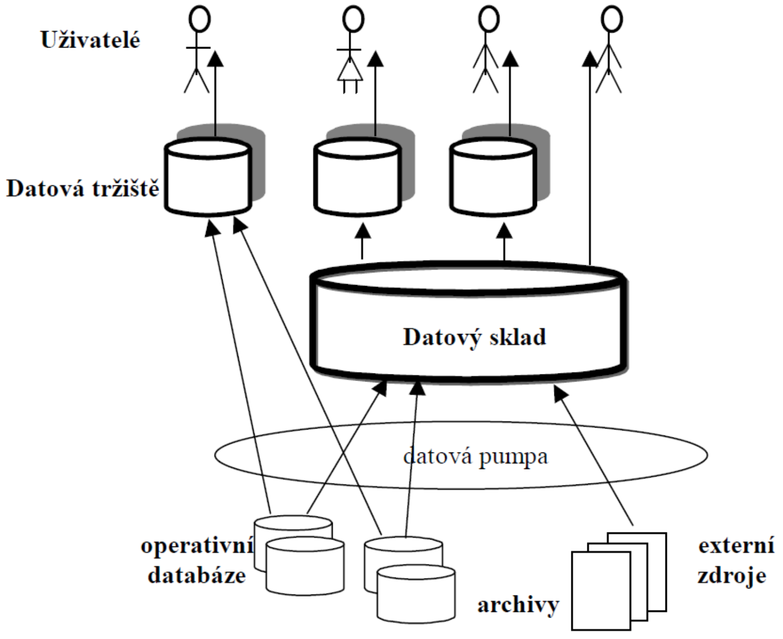


Figure 10.2 Possible solutions include data warehouse and data marts

## 9.3 Multidimensional analysis of a data warehouse

Tasks of multidimensional analysis

- The division of DW attributes by their roles in DW to facts, dimensions and other attributes,
- The design of the DW structure - defining the relationships between the dimensions forming the basis of the facts table,
- The definition of a hierarchy of dimensions,
- The determination of the additivity of facts and the definition of constrains of queries.

Dimensions, facts, attributes

Queries in the IS over an operational database are mostly at entity level. On the other hand, queries in DW are almost exclusively on the summed data, mostly on the values of some attributes aggregated by other attributes, on their so-called dimensional series and on their hierarchy.

Therefore, we must divide the source attributes into those that will be aggregated (summed, averaged, etc.) and those according to which they will aggregate. Because we will often use this partition, we are introducing new attribute attribute names for their different roles. When choosing attributes into DW, we also must consider these future roles:

- Dimensions are attributes according them we can sum and aggregate other attributes (i.e., dimensions are for example: time (daily, weekly monthly sums), seller (sum for the store, for the whole company), commodity (sum for individual goods, for type of goods), place (customer from municipality, district, county, state, etc.). These dimensions often form a hierarchy (in vertical division), and temporal or dimensional series (in horizontal divisions). We can analyze aggregated values (facts) using these dimensions (dimension hierarchies).
- Facts are those attributes that are the main reason for the record, and whose values are of interest to us in decision-making (quantity, price, profit, ...), usually not in individual source entities, but aggregated by dimension; aggregation is a key concept for analysis and the main reason for DW creation;
- Attributes are other attributes that do not belong to dimensions or to facts (name, business name, invoice number, ...), they are only relevant for source entities, usually not from DW derived results. It can also be descriptions of dimensions or their hierarchies or description of metadata.

Example:

A company's database contains following tables for stock records of purchased and sold goods:

Stock (store_card, product_name, id_supplier, price, amount_stock)

Purchase (store_card, purchase_date, amount_purchases, purchase_price)

Sale (store_card, date_of_sale, amount_sale, single_sale_price, id_buyer)

Customer (id_customer, company_name, name_of_representative, town, street, psc, state, phone)

Attributes that summed values will be interested to us are for example:

all quantities - amount_stock, amount_purchases and amount_sale; total prices for purchased and sold goods, i.e., derived data: purchase_price = amount_purchases * purchase_price, sale_price = amount_sale * single_sale_price; profit from sale = sale_price - purchase_price.

Dimensions will be the basic attributes: store_card, id_supplier, purchase_date, date_of_sale, id_customer, town or psc, state and possibly other derived attributes of date attributes: day, month, year, day of week or other. According to these attributes, data concerning quantity, price or profit can be summed up.

Attributes company_name, name_of_representative, phone specify only the information about dimensions. They serve as additional information. We can consider if we include them into data warehouse.

Sometimes the distribution of attributes is not unambiguous. For different purposes the same attribute can be either fact attribute or dimension attribute.

Example:

The attribute age can be fact (makes sense for to count min … avg) in some dimension (the average age of customers, the minimum and maximum age of customers who buy specific good) etc.

Other times, age may be in the role of a dimension (ages performance of primary school pupils, i.e., average performance by age, …). In other case, the derived attribute categorized age of adults called age_categ can acquire values 1 = age 18-23, 2 = age 24-29, 3 = age 30 – 34, etc. This attribute is dimension attribute.

Example

We can distinguish between facts attributes and dimension attributes when we write SQL Query – with the clause SELECT GROUP BY:

Facts are the attributes that can be aggregated (count, averaging, …), i.e., it makes sense to use for these attributes aggregate functions with the SELECT command, e.g.

SELECT SUM (…), COUNT (…), AVG (…), MIN (…), MAX (…)

Dimensions are the attributes that we use after the GROUP BY clause, fact dimensions are grouped and aggregated according to these dimensions.

Other attributes are those which can supplement our information.

Example:

We have a relational schema

Test (id_subject, id_teacher, date, attempt, points, mark )

It makes sense to compile SQL querries about the number of tests or counting of marks and points and averaging them by subjects (average mark, average number of points per subject), by teachers, or by date = test date or by order of the test = 1. term, 2. term, 3. term.

```
SELECT COUNT (*), AVG (points), AVG (mark), MIN (points), …, MAX (points) …

FROM Test

GROUP BY id_subject

Id_teacher

Id_subject, attemp

…
```

Dimension attributes are id_subject, id_teacher, etc. Facts are attributes points, mark, etc.

Note that it makes no sense to calculate the sum = total sum of marks or points for any dimension.

**The data structure of the data warehouse**

After dividing attributes into dimensions and facts, the structure of the data warehouse is modeled, table structures are designed. The task is to create a data structure so that all data, especially their sums according to different dimensions, is as simple as possible.

Each dimension (dimensional attributes and its optional attributes) consists of one table, called a dimensional table or DM - table (DM). The dimension table must have a single-attribute key; if there is no natural key, an artificial must be defined. There are no functional dependencies between the dimension attributes.

The link between the dimensions is information about the facts: the relevant facts are attached to the n-dimensional vector of dimensions. In the ERD, therefore, it is a set of dimensional tables with kardinality of the relationship between M : N : K. Connection is made using a multi-attribute key (one key attribute for each dimension) and facts form the other attributes of that connection. Such a table is called a fact table or a FT - table (FT).

Cardinality is the only value we have to model, others are guaranteed by a source database and a data pump.

Example:

Part of the conceptual schema of the carpet dealer's source database has tables:

Carpet (type, size, material, color, description)

Seller (seller-id, company, contact, address)

Sale (type, seller-id, quantity, price)

We want to register individual sellers in DW, type of carpet, material, color, quantity and price of carpets sold, and finally revenue and sales revenue.

Therefore, there are dimensions Seller (= shop), Carpet type, Carpet color (note that the separate D-table will be the Type (Carpet) from this attribute, which becomes a dimension.) These dimension attributes will each be in one dimensional table.

Individual sales form the link between dimensional tables with other attributes-facts, such as the quantity sold, the cost of sales, and other calculated attributes-facts, yields, and sales profits. The relevant entity relational diagram looks like this:

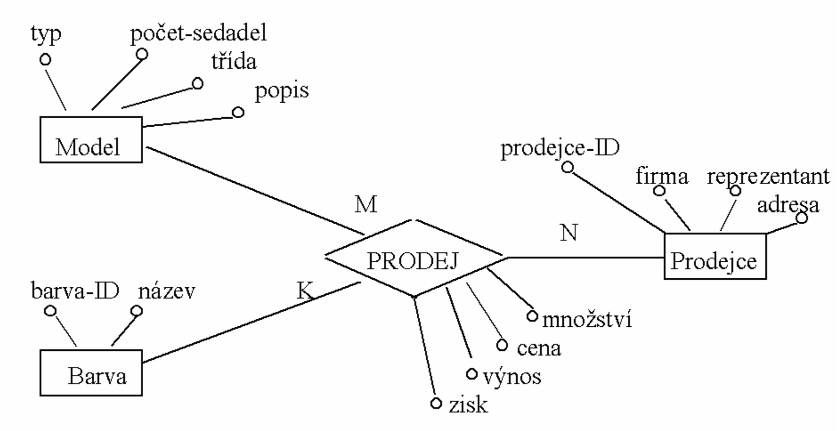


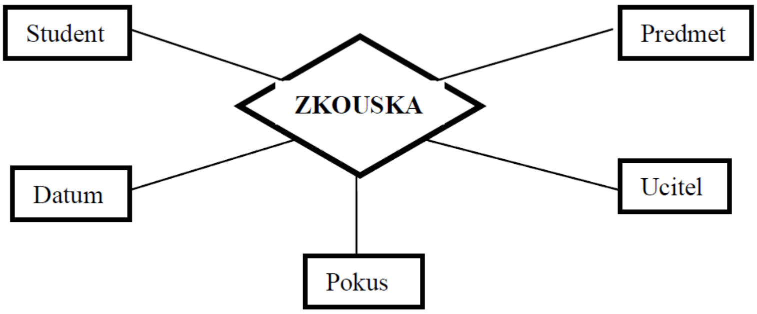
Figure 10.3 ERD of a part of data warehouse of a carpet dealer company

The displayed connection can be implemented by a link table between D-tables where each dimension is represented by its key, and the facts are other attributes of this connection:

Sale (seller-id, type, color-id, quantity, price, yield, profit)

Example:

Let's have relational schema of the source database Exam (id-subject, id-teacher, date, attemps, points, mark) where points and mark are facts and id-object, id-teacher, date, attemp are dimensions. We have only one table, and relevant ER diagram is:



Here, we have here 6 tables, 5 D-tables for each of the five dimensions and one F-table with five key attributes and two fact attributes - marks and points.

### Two approaches to the process of aggregating data

Users are interested in DW primarily data summed or otherwise aggregated. We consider aggregation are classical aggregation functions for example from SQL language, i.e., COUNT (counts), SUM (sums), AVG (averages), MIN (minimum) and MAX (maximum).

Another problem to solve in data warehouses is how and when the aggregate data required will be calculated.

1. The first approach is to leave only atomic data in 3NF sessions, and to perform the required aggregation and processing over the data at each query of a user. This approach has the following advantages and disadvantages:
   - If data volumes are large, then processing is unbearably slow and demanding on HW even in stand-alone DW;
   - The same aggregate data and sums are counted each time when similar queries repeat;
   - Only basic data is stored in the DS, so this approach has lower demand on memory capacity consumption.
2. The second approach is to calculate data and preserve all possible aggregations for all defined dimensions and their hierarchical degrees. Advantages and disadvantages of this approach:
   - For queries, the required data is not counted, only searched for and displayed or conveniently visualized, for example, by graphs;
   - This procedure is more demanding for a preliminary analysis of future possible requirements, so it requires dimensional modeling that no longer complies with 3NF because it contains data derived;
   - The data contains dual-type redundancy: derived atomic data so that it does not have to be re-counted for repeated queries and summed data for the same reason;
   - Optimization is required in data access due to new types of analyzes, so another type of DW management is needed than with the operational information system with the database.

The second approach is more convenient, faster in operation and is therefore used in data warehouse technology.

### Hierarchy of dimensions

Dimensional attributes are sometimes not simple but can form an entire hierarchy. Either the atomic data form the lowest level of the hierarchy or as other accompanying attributes have their higher levels, or higher levels can be deduced (derived) from the basic data.

Example:

Data is a very common and useful hierarchy. In the previous example, the date of sale with the basic D-table Date (id-date, date) may be one of the other dimensions. Apart from the amount of carpets sold for individual days, sums per week, month, year, maybe other - quarter, etc. will certainly be interesting. These additional dimensions can be derived from the dimension Date. As in this example, similar data are calculated in advance according to the DW rules and stored in the D-tables. In this example, the corresponding hierarchy can be represented as follows: day - week - month - quarter - year.

Example:

Another example is the often-occurring and useful hierarchy of territorial layout. In the previous example, we divide the attribute address into street attributes, number, municipality, psc, state and we will have a hierarchy: street - psc - municipality - district - county - state where district and county can be obtained from appropriate codes and psc. Postcode corresponds to the post office. Sumy for municipality, district, county, state will certainly be interesting outside the sums of carpets sold for individual sellers. All derived data will be the other attributes of the D-table Seller.

Additivity of attributes

We have dimensions and facts defined, and we expect queries about sums and other aggregated facts by individual dimensions or their combinations (for example, the sum of carpets sold and their price, ... by month in each region). If we have several dimensions, then we can have a great amount of simple aggregated values, and if we add all combinations of these dimensions, the amount of values is even more numerous, and visualizations of these simple aggregated values are unclear.

The DW is based on the assumption that any meaningful queries are expected for any possible aggregate value - even if the user did not request it in advance (ad hoc queries = business related queries). Thus, in theory, we can assume queries of all facts by all dimensions and all their combinations by all aggregation functions COUNT, SUM, AVG, MIN, MAX.

We have already seen in the school example that some of values do not make sense in reality (for example, the sum of the marks in tests). Therefore, it is necessary to define another integrity constraint that will limit the set of possible queries over DW – additivity of attributes.

**Definition**

Additivity of fact means the meaningfulness of aggregating its values with respect to all dimensions.

It is valid:

- Facts are numeric values and theoretically can be summed up for row groups.
- For simplicity, this property is assumed in the fact table.
- Any exception is explicitly specified in the schema.
- Semi-additive and non-additive facts are sometimes distinguished.

**Definition:**

The attribute in the fact table is called semi-additive if it is not additive with one or more dimensions. Non-additive attributes are not additive due to any dimension.

Thus, the division of facts can be divided according to the adivitity to:

- Additive - aggregation without limitation, all types of aggregation for all dimensions are possible;
- Semi-additive - partially additive, meaning only some aggregations for some dimensions are meaningful;
- Non-additive - it makes no sense to aggregate them according to any dimension.

Part of the definition of attribute-fact additions will include so-called limitations on queries.

## 9.4 Database modeling

ROLAP and MOLAP

DW is used to predict and to storage aggregate data. This is another redundancy in the DW, but due to its use, this redundancy does not create inconsistencies and, on the contrary, it speeds up the responses to querries of users - it only selects the pre-calculated data and need not to calculate them. However, this approach is not easy, there are new problems with how to model aggregated data and its hierarchies at different levels.

At the database level, there are practically two basic approaches and some of their variants:

- ROLAP (Relative OLAP) means DW implementation using relational tables (dimension tables and fact tables) organized into star schemas.
- MOLAP (Multidimensional OLAP) implements DW using hyper cubes or multidimensional cubes. Also called POLAP (proprietary OLAP). This technology must be embedded in the DBMS.
- HOLAP (Hybrid Access to ROLAP and MOLAP)
- DOLAP (Desktop OLAP, data warehouse on client computer)

It is not difficult to design the implementation of the schema in the relational data model. Unfortunately, for multidimensional analysis, the resulting standardized tables would be too cumbersome. Aggregated values tables should be counted on every change in the baseline data. The number of dimensions of the hierarchy of dimensions is given from the analysis and it is not easy to dynamically change the display of the "up and down" sum according to these hierarchies.

It is more efficient to save a database using a multidimensional cube. The cube has as many dimensions as you need to record attributes = dimensions, each field in this cube contains the corresponding values of attributes = facts. At first glance, it is a multidimensional array. However, there will be many combinations of empty dimension values, the field will be sparse, and therefore special implementation techniques are used to implement dimensional dice. The dice principle then leads to a simple approach when calculating different aggregations.

### The model of data warehouse in ROLAP

### Star schema

As we foresee from the previous interpretation, special assembly of relations (tables) called Star is used to implementation a data warehouse using the DBMS. Necessary formalization is as follows.

**Definition:**

Star schema S is a triplet <D, F, CC> where D is the set of dimensional tables, F is a fact table and the CC is set cardinalities.

The star depicts the reality from ERD modeled with the help of an association table (referred to as a F-table here), a set of other n tables (dimension D) and a set of cardinalities CC. The selected attribute from each dimensional table D is the key and is denoted KD. The key of table F is the union $\cup_{i=1..n} KD_i$. Non-key attributes from F are facts.

**Definition:**

Cardinality $CC_i$ is defined for F and dimension $D_i$ follows: if F* is a fact table and D* dimensional table, then the $CC_i$ is satisfied if for each row *u* from F* there is only one row *v* in D* such that $u.KD_i = v.KD_i$.

Thus, the rows of the fact table and rows of each dimension are in relationship N : 1. KD is in F a foreign key, so it must not be NULL. In other words, compulsory membership N: 1 (just 1) is on the side of fact.

**Definition:**

Let S is a star schema. Then, we denote a set of tables $D^*_{i\ ,\ i\ =\ 1..n}$ and F* which satisfy the cardinality CC a **multidimensional database**.

The following applies:

- Each dimension $D_i$ has one key $D_i$ $KD_i$ (e.g., artificial) and is represented by a dimensional table $D_i$ ;
- There is one fact table F, the key of this table is formed by the foreign keys of $KD_i$ of all related dimensions $D_i$, its other attributes are facts f1, f2, ... ;
- The whole structure F and {Di} is called a multidimensional database.

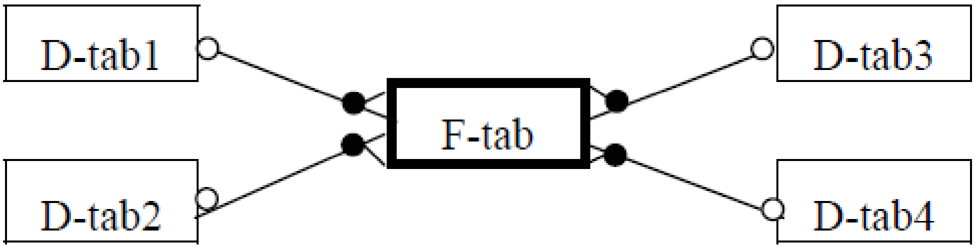


Figure 9.4 General star schema

Example:

Our example of a data warehouse of carpet shop could be modeled using the association table = F-table and a few D-tables as follows:

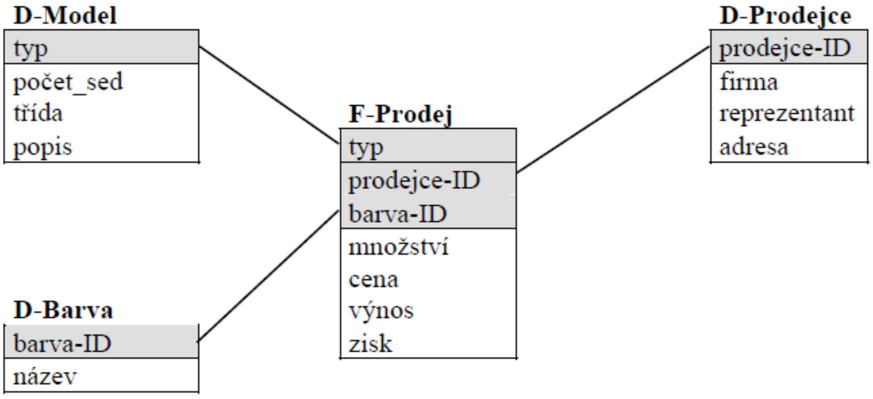


Figure 9.5 Example of a star scheme

### Fact constellation schema

Data warehouse with a single star scheme is simple, but usually not enough for all storage needs. It is often necessary in a DW to record multiple fact tables. However, some dimensions may be identical for various F-table, some may be different. Generally therefore, it is possible to extend the principle of a star to a fact constellation diagram .

**Definition:**

Fact constellation scheme is a triplet <D, F, CC > where D is the set of dimensional tables, F is a set of fact tables and CC is set of cardinalities. For each F ∈ F there is a subset D' ⊆ D and CC' ⊆ CC such that <D', F, CC' > is a star scheme.

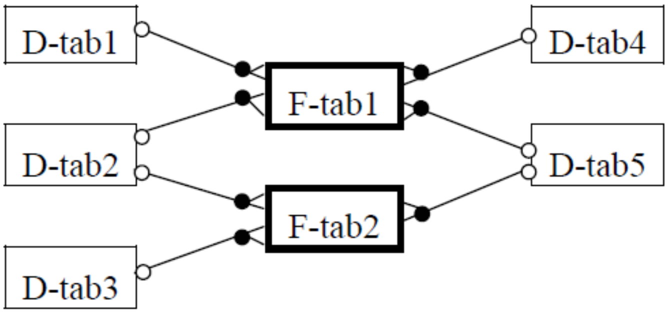


Figure 9.6 General scheme of fact constellation

Example:

The example of a data warehouse of a shop with carpets is simplified. We can imagine that the user will be interested in tracking not only the course of sales, but also the purchase of carpets, delivery of carpets from suppliers and other phenomena.

The following diagram shows the fact constellation with F-tables Sales and Orders. We can see, they use some common dimensions.

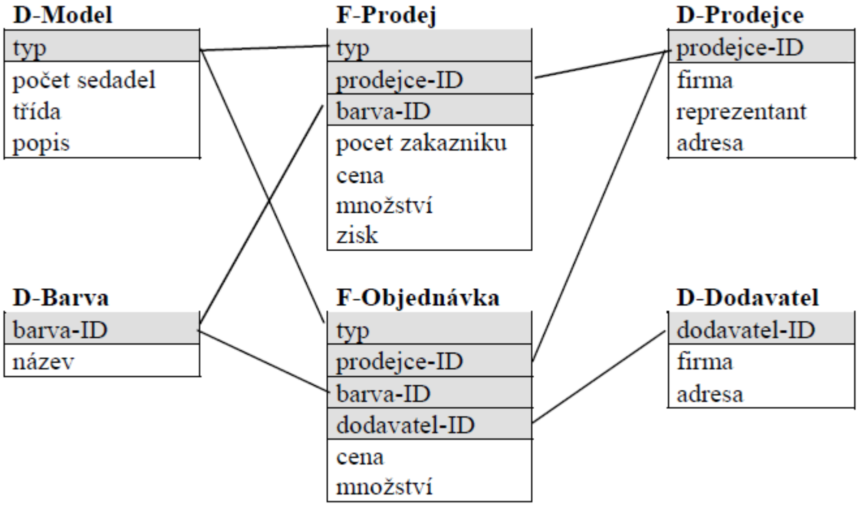


Figure 9.7 Example of fact constellation tables Sales and Orders

### Restrictions on queries

We generally assume in DW the use of all meaningful aggregations of facts with respect to all dimensions and their combinations, according to all aggregation functions COUNT, SUM, AVG, MIN, MAX. However, some aggregations do not have a real meaning, although they can be counted (for example, in the example Exam, we can count the sum of the marks, but the meaning does not have that number).

Most of the aggregated values have meaning and, in accordance with the data warehouse objectives, all these aggregations will be counted, even if the sponsor did not specify it directly. So DW is ready for all unexpected and unforeseen queries (ad hoc queries). Those aggregations that do not make sense in reality will be set as new integrity constraints. Formally, we will call them limitations to queries, and we extend defined star scheme with these constrains.

**Definition:**

The set of restrictions on querries DO over the fact constellation diagram <D, F, CC> is a set of triples (F, D, Agg), where F is a fact from F, and D is the dimension, and Agg is the set aggregation functions for which it does not make sense to aggregate F due to dimension D.

DO = {(Fi, Dj, AG1, AG2, ...), (Fk, D1, AG1, AG3, ...), ...}

Interpretation of each triplet is therefore negative information that the function of Agg can not be used on attribute A due to the dimension D. The fact constellation diagram is quaternion <D, F, CC, DO> with already known meaning.

Example:

We have scheme EXAM (id_subject, id_teacher, date, attemp, points, mark).

Consider the meaning of all aggregated data in a query of the following types ( be careful: the task is not to consider the what the user will or will not ask, but what value does not make sense as a number):

SELECT COUNT (*), SUM (points), SUM (mark), AVG (points), AVG (mark) ... MIN, MAX ...

FROM EXAM

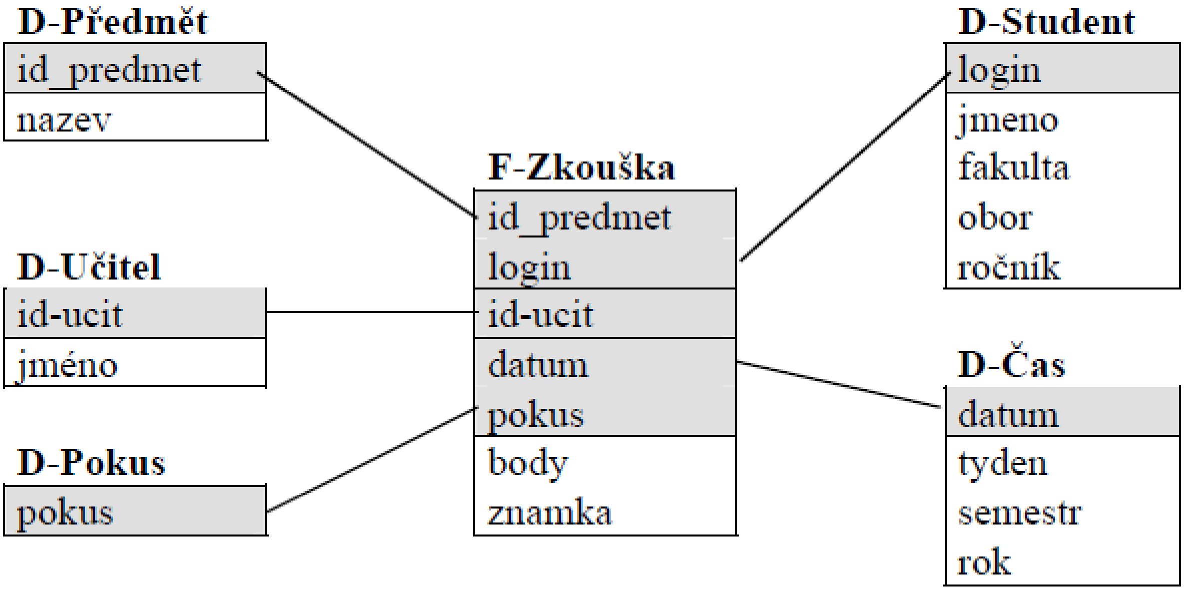
GROUP BY object-id

id-teacher

id-subject, attempt ...

It seems to make sense: the number of attemps per subject, teacher, student, date, attempt average, min, max points and marks for all dimensions.

But it does not make sense to count: the sum of marks for any dimension and the sum of points for some dimensions.



Conclusion: The attributes of points and marks are semiadditive, restrictions on the questions will be:

DO = {(Mark, Subject, SUM) (Mark, Teacher, SUM), ...}

**Hierarchy dimensions in ROLAP**

Modeling of dimensional hierarchies

Dimensions are described using their key and descriptive attributes (name, description, ...). In some dimensions, dimensional hierarchies may be hidden, hierarchy attributes are called members of the hierarchy. More generally, entities with their own attributes may be members of the hierarchy.

Example:

The DW SALES has dimension Type with attributes type (key), unit_weight, color, description.

Dimensional attributes of the Type, date, seller of the database SALES have hierarchy:

Type: goods - color - everything

Some dimensions may even have more than one hierarchy, a typical example is

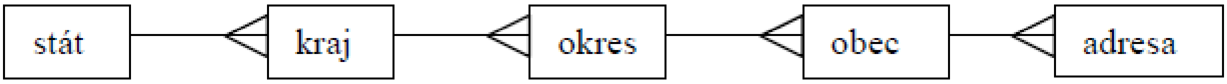
Date: day - month - quarter - year - all

day – week ------------------- all

day - the day of the week ------------ all

Seller: address - the town - district - region - state - everything

At a conceptual level, hierarchy forms a chain of entity types, where the cardinality of two neighbors is 1: N.

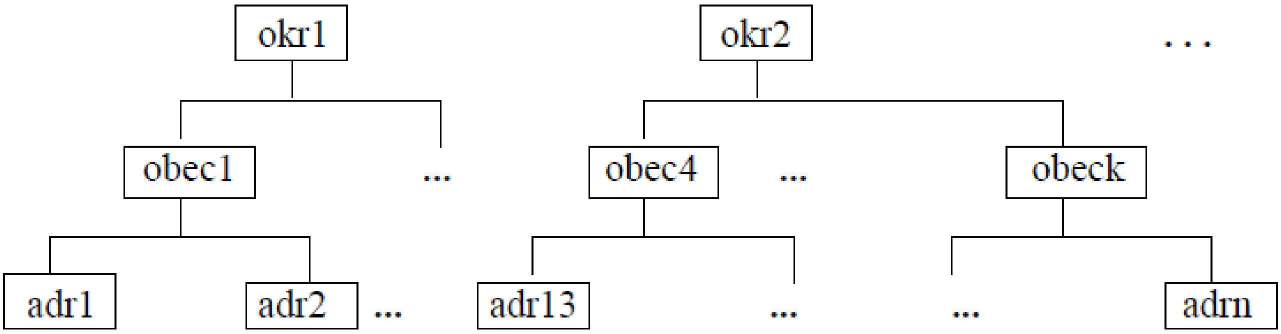


Aggregation are the typical use for the dimensional hierarchy. User can go in the hierarchy one level up (roll up) or one level down (drill down) or to go horizontally along dimensional row. Time series are one case of dimensional lines which is very often - aggregated values of facts for each day or week or month etc.

Extension of the hierarchy may be a set of trees, where each tree leaf contains data about individual entities.

Example:

Part of the hierarchical tree dimension address - town – district; dimensional series are illustrated horizontally.



When modeling hierarchies of dimension, a new problem arises, as how to store hierarchical steps in DW.

The following techniques are used in the RDM:

### Hierarchy of dimensions in a fact table

The easiest way in terms of the number of tables used is as follows:

1F = one fact table for all hierarchical levels and 1D = 1 table for each dimension

The smallest space for data consumes

- Placing the entire hierarchy of a dimension into one D-table (unnormalized with great redundancy and artificial identification);
- Whole hierarchy is perceived as a single domain in one D-table.

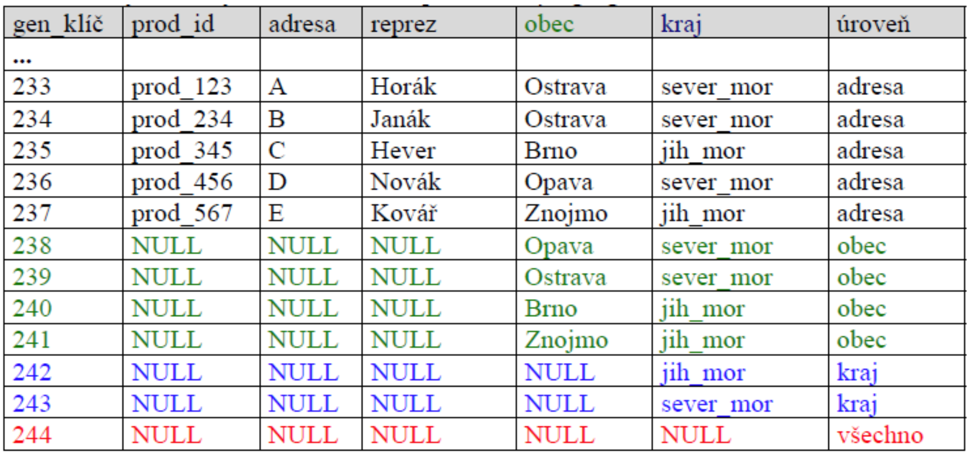
The entire DW is implemented as a star or fact constellation with one F-table and with the source and aggregated data on all levels of the hierarchy.

To differentiate the values of dimensions and their hierarchies are used two methods:

A. record of each dimension and its hierarchical grades have a generated key

Example:

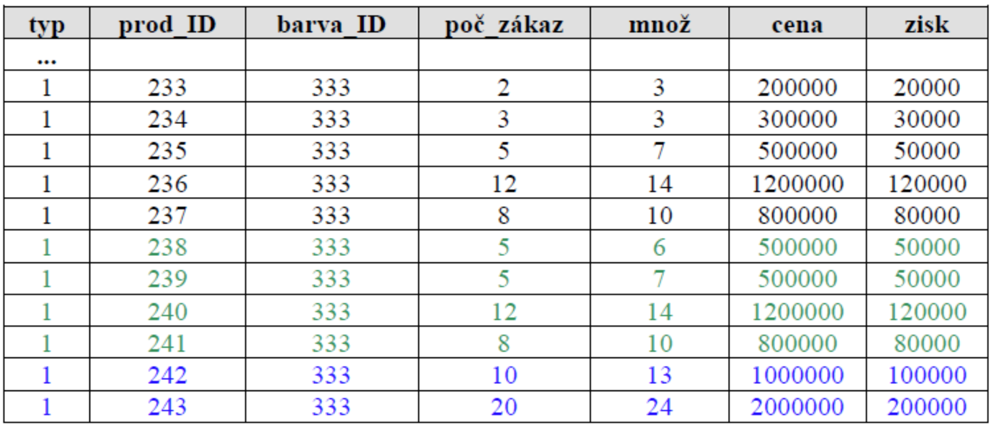
D-table Seller, hierarchical grades are color-coded, artificial key is automatically generated not only at the lowest level but also for all aggregation levels.



| gen_klíč | prod_id | adresa | reprez | obec | kraj | úroveň |
|---|---|---|---|---|---|---|
| ... | | | | | | |
| 233 | prod_123 | A | Horák | Ostrava | sever_mor | adresa |
| 234 | prod_234 | B | Janák | Ostrava | sever_mor | adresa |
| 235 | prod_345 | C | Hever | Brno | jih_mor | adresa |
| 236 | prod_456 | D | Novák | Opava | sever_mor | adresa |
| 237 | prod_567 | E | Kovář | Znojmo | jih_mor | adresa |
| 238 | NULL | NULL | NULL | Opava | sever_mor | obec |
| 239 | NULL | NULL | NULL | Ostrava | sever_mor | obec |
| 240 | NULL | NULL | NULL | Brno | jih_mor | obec |
| 241 | NULL | NULL | NULL | Znojmo | jih_mor | obec |
| 242 | NULL | NULL | NULL | NULL | jih_mor | kraj |
| 243 | NULL | NULL | NULL | NULL | sever_mor | kraj |
| 244 | NULL | NULL | NULL | NULL | NULL | všechno |

D-Vendor table with hidden hierarchies and the generated key

Similarly, dimension color and type have generated a key. Then, a fact table contains all grades of hierarchy of all dimensions and their combinations:



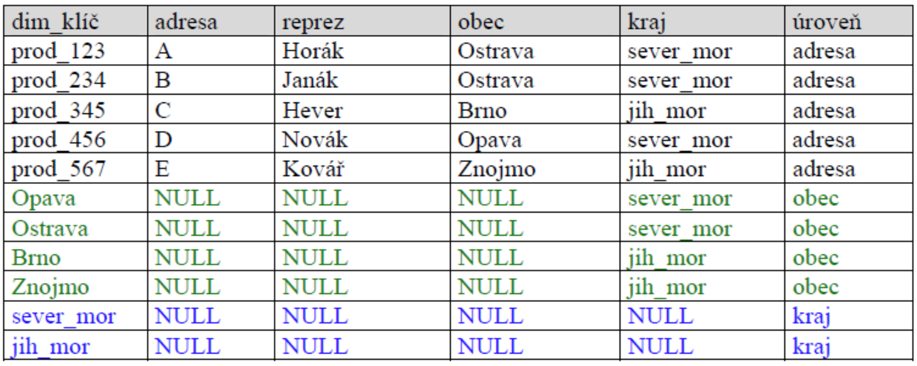
| typ | prod_ID | barva_ID | poč_zákaz | množ | cena | zisk |
|---|---|---|---|---|---|---|
| ... | | | | | | |
| 1 | 233 | 333 | 2 | 3 | 200000 | 20000 |
| 1 | 234 | 333 | 3 | 3 | 300000 | 30000 |
| 1 | 235 | 333 | 5 | 7 | 500000 | 50000 |
| 1 | 236 | 333 | 12 | 14 | 1200000 | 120000 |
| 1 | 237 | 333 | 8 | 10 | 800000 | 80000 |
| 1 | 238 | 333 | 5 | 6 | 500000 | 50000 |
| 1 | 239 | 333 | 5 | 7 | 500000 | 50000 |
| 1 | 240 | 333 | 12 | 14 | 1200000 | 120000 |
| 1 | 241 | 333 | 8 | 10 | 800000 | 80000 |
| 1 | 242 | 333 | 10 | 13 | 1000000 | 100000 |
| 1 | 243 | 333 | 20 | 24 | 2000000 | 200000 |

F-table Sale hierarchy dimension along Seller

B. records of each dimension have a self-identifiable key

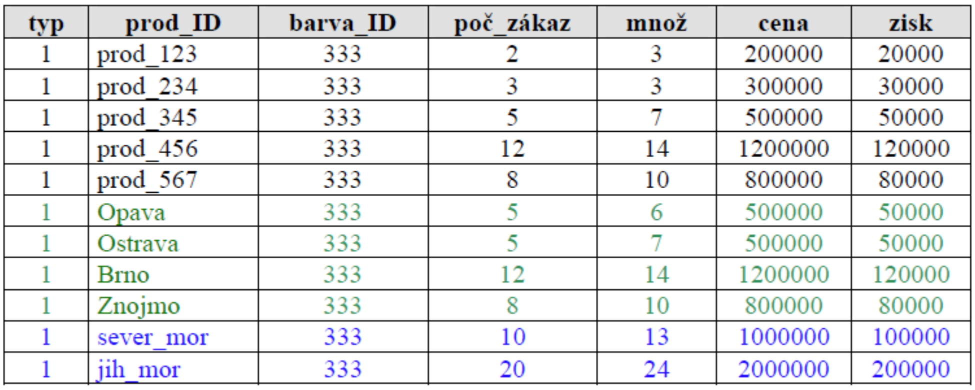
Example:

Another variant uses inhomogeneous key that simultaneously defines the hierarchical level.



| dim_klíč | adresa | reprez | obec | kraj | úroveň |
|---|---|---|---|---|---|
| prod_123 | A | Horák | Ostrava | sever_mor | adresa |
| prod_234 | B | Janák | Ostrava | sever_mor | adresa |
| prod_345 | C | Hever | Brno | jih_mor | adresa |
| prod_456 | D | Novák | Opava | sever_mor | adresa |
| prod_567 | E | Kovář | Znojmo | jih_mor | adresa |
| Opava | NULL | NULL | NULL | sever_mor | obec |
| Ostrava | NULL | NULL | NULL | sever_mor | obec |
| Brno | NULL | NULL | NULL | jih_mor | obec |
| Znojmo | NULL | NULL | NULL | jih_mor | obec |
| sever_mor | NULL | NULL | NULL | NULL | kraj |
| jih_mor | NULL | NULL | NULL | NULL | kraj |

Dimensional table Seller with self-identifiable dimensional keys



| typ | prod_ID | barva_ID | poč_zákaz | množ | cena | zisk |
|---|---|---|---|---|---|---|
| 1 | prod_123 | 333 | 2 | 3 | 200000 | 20000 |
| 1 | prod_234 | 333 | 3 | 3 | 300000 | 30000 |
| 1 | prod_345 | 333 | 5 | 7 | 500000 | 50000 |
| 1 | prod_456 | 333 | 12 | 14 | 1200000 | 120000 |
| 1 | prod_567 | 333 | 8 | 10 | 800000 | 80000 |
| 1 | Opava | 333 | 5 | 6 | 500000 | 50000 |
| 1 | Ostrava | 333 | 5 | 7 | 500000 | 50000 |
| 1 | Brno | 333 | 12 | 14 | 1200000 | 120000 |
| 1 | Znojmo | 333 | 8 | 10 | 800000 | 80000 |
| 1 | sever_mor | 333 | 10 | 13 | 1000000 | 100000 |
| 1 | jih_mor | 333 | 20 | 24 | 2000000 | 200000 |

Fact table Sales with hierarchies along the dimension seller

The advantages of this implementation are

- A small number of tables: there is only one fact table with source facts and also aggregated values; they are distinguished by the key values of the dimensions,

- F-table may include not only aggregation by the simple dimensions, but also by their any combination.

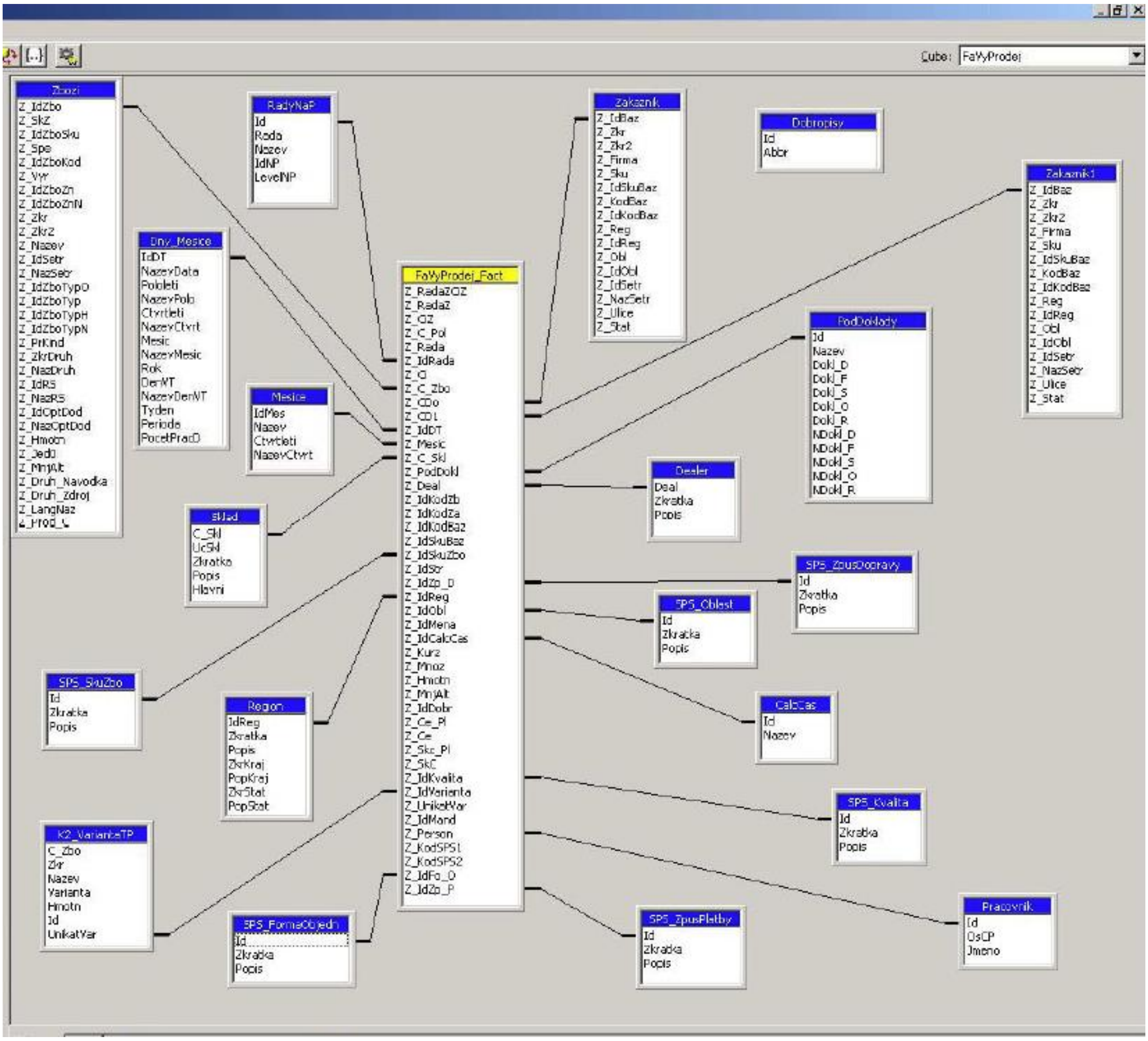
Drawbacks

- The fact table is very large and a searching of a corresponding set of rows is thus slowed down.

Example:

The following figure shows an example of a star scheme of a real warehouse. We see that the number of dimensions can be high. This warehouse is implemented in MS SQL Server 2008 and the scheme is also from this environment.

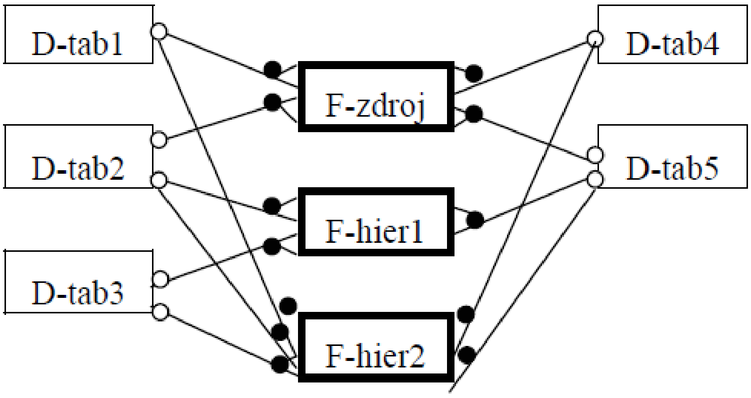
Note that dimensional tables have dimension hierarchies. Typically, for example, at time data Day_month or at Goods.



## Hierarchy dimensions using Constellation of F-tables

The second option is a division of the fact table - F-table is not one, but the original table is horizontally "chopped up", is new for each level of the hierarchy. This means that F-tables form for each dimension a string of tables. Dimensional tables include (as above) also the hierarchy of dimension.

General scheme



General scheme for the hierarchy constellation of facts (for one basic F-table)

The fact constellation scheme is used here in a different sense than above - for the hierarchical strings of fact tables. This solution create many new fact tables, what is challenging for DW designer - it is necessary to chose from right tables when realizing the querry of a user. However, the data of the entire table is mostly used, and it is not necessary to search for them. This and the following solution contain derived (aggregated) fact tables with redundant information to accelerate the response to user queries.

Example:

Again, the SALES schema, but the pre-calculated aggregated (sumed) data on number of customers, price, sales volume and profit are stored in separate fact tables. For each dimension level or combination of dimensions, there is one new fact table.

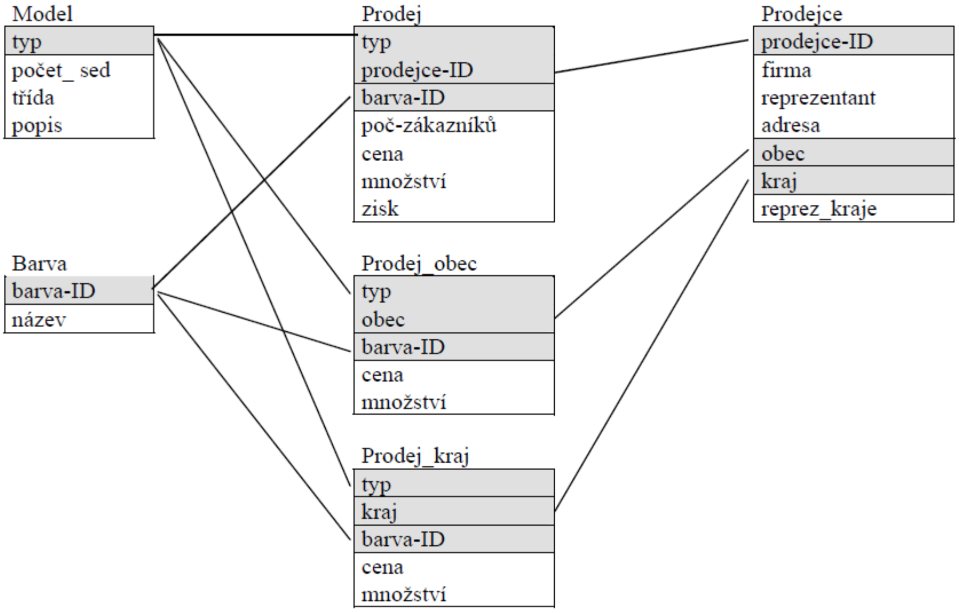
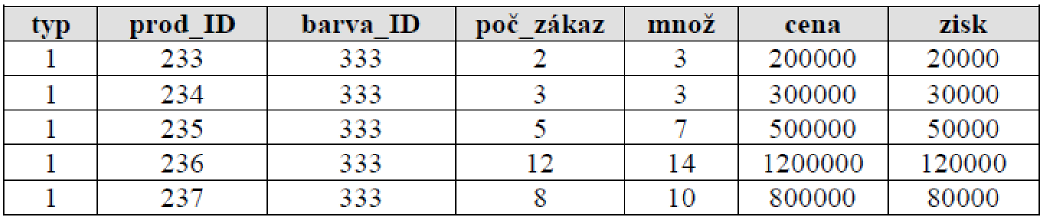


Figure 9.10 Example dimensions using constellation hierarchy fact tables

Example:

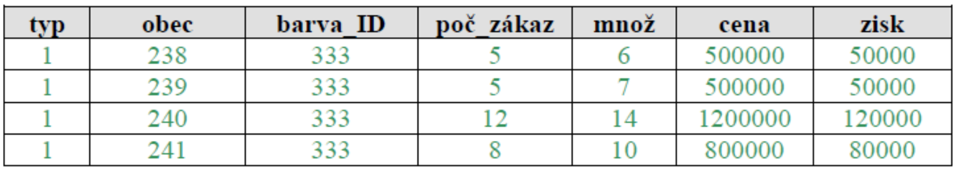
Part of the completed fact tables Sales with the division of F-tables by level.

Sales - a fact table with a generated key of the dimension Seller



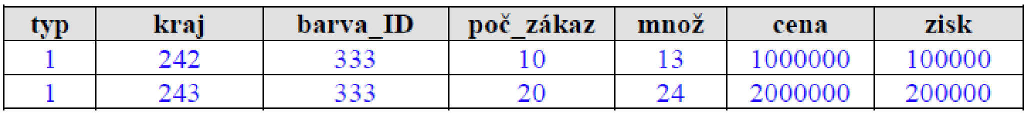
| typ | prod_ID | barva_ID | poč_zákaz | množ | cena | zisk |
|---|---|---|---|---|---|---|
| 1 | 233 | 333 | 2 | 3 | 200000 | 20000 |
| 1 | 234 | 333 | 3 | 3 | 300000 | 30000 |
| 1 | 235 | 333 | 5 | 7 | 500000 | 50000 |
| 1 | 236 | 333 | 12 | 14 | 1200000 | 120000 |
| 1 | 237 | 333 | 8 | 10 | 800000 | 80000 |

Sale_town - a fact table with a generated key dimension Seller



| typ | obec | barva_ID | poč_zákaz | množ | cena | zisk |
|---|---|---|---|---|---|---|
| 1 | 238 | 333 | 5 | 6 | 500000 | 50000 |
| 1 | 239 | 333 | 5 | 7 | 500000 | 50000 |
| 1 | 240 | 333 | 12 | 14 | 1200000 | 120000 |
| 1 | 241 | 333 | 8 | 10 | 800000 | 80000 |

Sale_region - a fact table with a generated key dimension Seller



| typ | kraj | barva_ID | poč_zákaz | množ | cena | zisk |
|---|---|---|---|---|---|---|
| 1 | 242 | 333 | 10 | 13 | 1000000 | 100000 |
| 1 | 243 | 333 | 20 | 24 | 2000000 | 200000 |

### Hierarchy dimensions using Snowflakes schema

An alternative of the previous solution is a layout of hierarchy not only at facts, but also at multiple dimensions into more tables:

- Not only F-table, but D-tables are distributed according to hierarchical levels
- Each hierarchical level of fact table has a connection to the respective hierarchical level of a dimension
- Keys of hierarchy stages show into smaller dimensional tables.

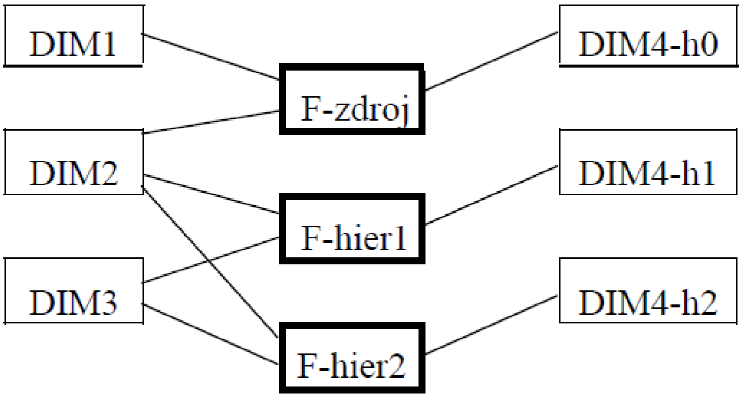


Figure 10.11 General snowflake schema hierarchy hierarchy of facts and dimensions

Example:

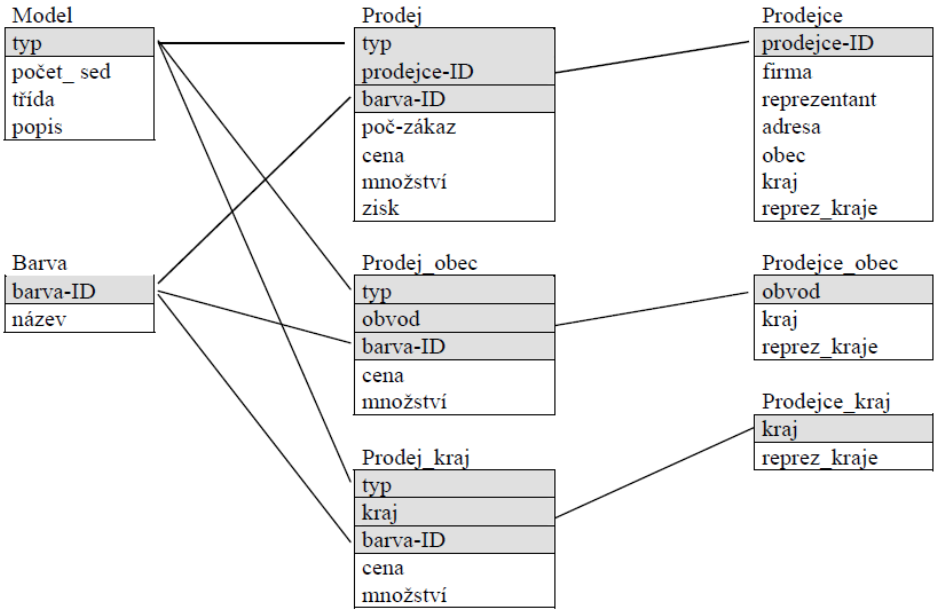


Figure 10.12 Example snowflakes along the dimension Seller

**Hierarchy dimensions – fact constellation with explicit hierarchies**

Theoretically the best resolved structure with clear and exact conceptual level is as follows:

- Dimension tables are broken down by hierarchical degrees with an explicit hierarchy definition using hierarchical links,
- For each dimension and its degree, the set of relevant facts is specified, normalization is retained (redundancy is hidden in repeating aggregated information in redundant fact sheets).
- However, the amounts of F-tables and D-tables with explicit hierarchy-dimension relationships puts great demands on designers and on the application warehouse programmer when realizing querries.

General scheme

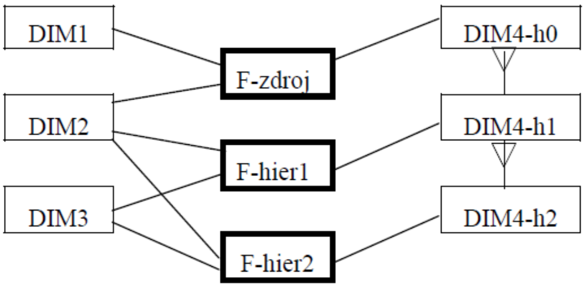


Figure 9.13 General scheme of constellation for the hierarchy of dimensions 1: M

Example:

D-table Seller is normalized (divided) into 3 tables with the connection of 1 : M, on the basic shop, town shops and shop in the region.

Similarly, D-table Type of carpet is divided into two tables with the connection of 1 : M, on the specific type of carpet, and on material of a carpet.

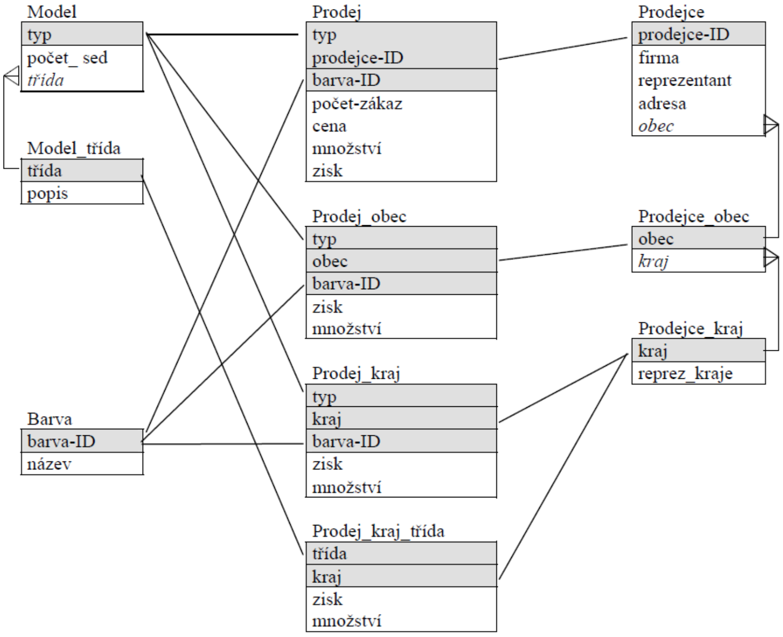


Figure 9.14 Example explicit hierarchy along dimensions Seller and Type

## 9.5 Dimensions and their hierarchies in MOLAP

Hypercube and multikostky

Multidimensional database is also used instead of the DW database model using a number of tables. It is a generalized principle of Excel tables, where it is possible to sum up row and column, or other aggregated values. The data warehouse is realized instead of a two-dimensional table by a so-called multi-dimensional (or m-dimensional "table"), which has similar properties to the above-mentioned Excel two-dimensional table. Facts on the lowest level are stored in the base fields of the dice. Since the facts in one F-table are more, then the whole n-tuples of facts are stored in these fields. Each dimension of the table corresponds to one dimension.

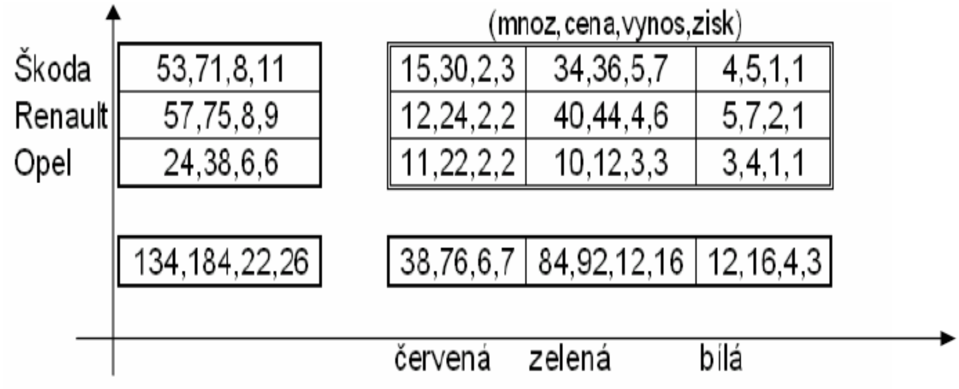
Definition:

Multidimensional database, or hypercube is m-dimensional space of dimensions over which there is n-dimensional space of facts. Above them, there may be a large-area of aggregate dimensions.

Hypercube contains aggregate (in this case summation) data, i.e., data derived. Dimension forms the axis of the cube, each described one-atribute key, n-tuples of facts are stored in individual fields of a cube. I.e., the values of n-tuples of fact depends on the respective m-tuple dimensions.

Example:

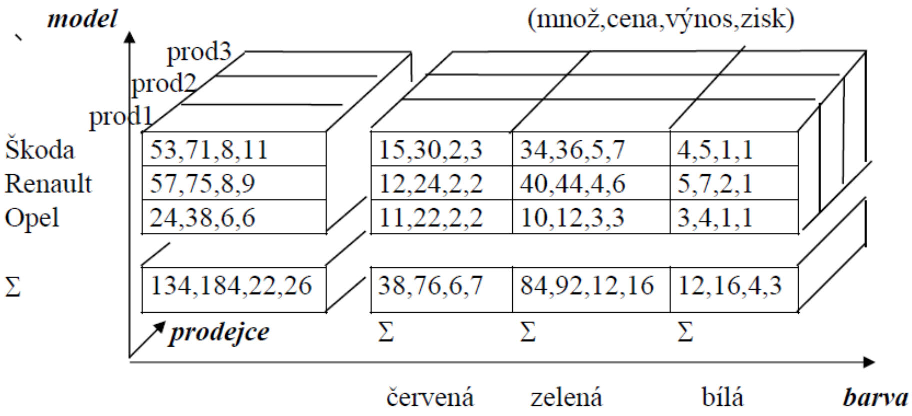
A very simple example of selling for 2 dimensions Type and Color. Hypercube is only a 2-dimensional table, a 2-dimensional space with 4-dimensional vector of facts.



| | | červená | zelená | bílá |
|---|---|---|---|---|
| Škoda | 53,71,8,11 | 15,30,2,3 | 34,36,5,7 | 4,5,1,1 |
| Renault | 57,75,8,9 | 12,24,2,2 | 40,44,4,6 | 5,7,2,1 |
| Opel | 24,38,6,6 | 11,22,2,2 | 10,12,3,3 | 3,4,1,1 |
| | 134,184,22,26 | 38,76,6,7 | 84,92,12,16 | 12,16,4,3 |

Example:

The hypercube forms 3-dimensional space for simple 3 dimensions Type, Color and Seller where values of 3 dimensions forms the 3 coordinates.



| | | červená | zelená | bílá |
|---|---|---|---|---|
| Škoda | 53,71,8,11 | 15,30,2,3 | 34,36,5,7 | 4,5,1,1 |
| Renault | 57,75,8,9 | 12,24,2,2 | 40,44,4,6 | 5,7,2,1 |
| Opel | 24,38,6,6 | 11,22,2,2 | 10,12,3,3 | 3,4,1,1 |
| Σ | 134,184,22,26 | 38,76,6,7 | 84,92,12,16 | 12,16,4,3 |
| | | Σ | Σ | Σ |

The cube model allows to record the values of the facts for all combinations of dimension values (elements of Cartesian product of corresponding domains of dimensions). Any combination of dimensions may not be present in the data and there is no n-tuple of facts, so the cube is sparsely occupied.

Determination of sparsity should be part of dimensional modeling.

In order not to waste the memory capacity, which would be very high for a larger number of domains and their large domains, this hypercube is sometimes taken only a logical model. Its realization can then be modified so that only its "occupied" parts are implemented. Let's define these parts.

**Definition:**

Multi-cube is a subspace of hypercube, in which nonexistent combinations of dimensions are not present.

Hypercube can therefore be understood as unification multi-cubes, a kind of their "packaging".

Hypercube is logically and implementary simplier, multi-cubes implementary more effective. Therefore, a combination of both models – we talk at the logical level about hypercubes, but at the physical level multi-cubes are implemented. Implementation of the data warehouse takes care of that.

Sometimes, multicubes are divided on:

- Block multi-cubes - facts and dimensions including time form the dimension of a cube,
- Serial multi-cubes - for every fact there is a separate cube with all dimensions, facts thus form dimensional ranges (e.g., time series for time dimension).

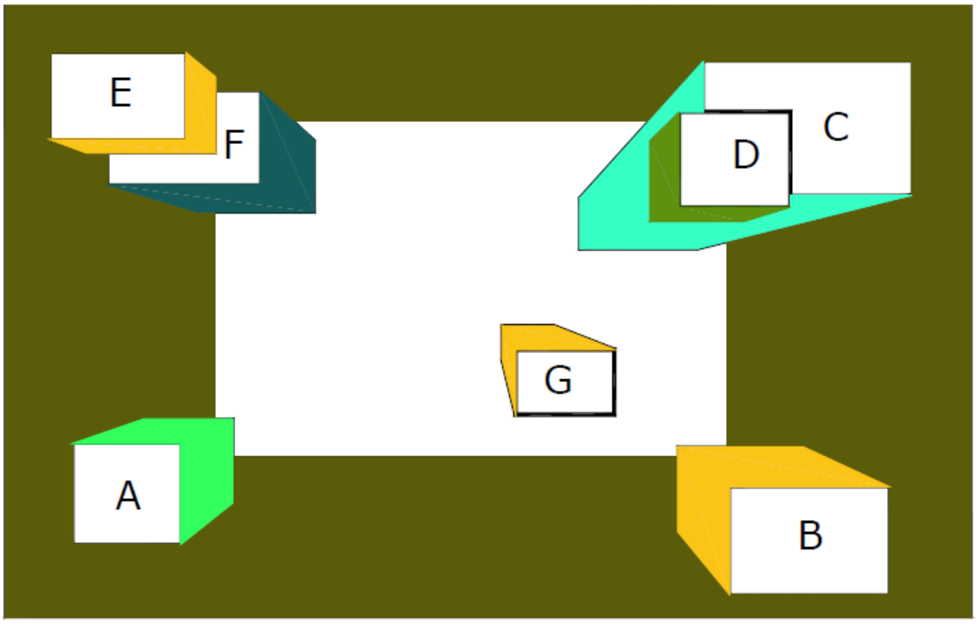


Figure 9.26 Hypercube and its multi-cubes

What variant of database model is used, it depends on the implementation of the data warehouse. Some have only one variant, others both, and user can select variant when creating the DW.

Questions:

1. Describe the course of modeling the data warehouse from the assignment after the decision on environment implementation.

2. Describe the various conceptual modeling tasks.

3. Could you describe the individual tasks multidimensional modeling.

4. Description of individual tasks of database modeling.

5. What is the difference between the three-tier architecture and database modeling data warehouse?

6. What is the role of conceptual modeling DS?

7. What is the data market and what are the options relation DS and DM?

8. What attributes are called dimensions that facts and attributes that again?

9. Which attributes aggregate and according to what rules?

10. When aggregated values are calculated?

11. How is modeled on the level of ERD data warehouse due to the division of the attributes of the dimensions and facts?

12. Use what types of DBMS and their technologies are implemented data warehouses?

13th Define and explain the concept of a star schema DS.

14. What do we call the dimension tables and fact tables what?

15. What is the cardinality of the dimension tables and fact?

16. What is called multidimensional databases?

17th Define and explain the concept constellation DS.

18. What is the additivity facts and distinguish what types of additivity?

19. What are the limitations on queries and define why?

20. What are the dimensions, hierarchies, as they know?

Why in the 21st DS store aggregated data at all levels of the hierarchy dimension?

22nd Which technologies are solved store aggregated data at multiple levels of aggregation?

23. Description storing facts hierarchy of dimensions in a single table with the generated key.

24. Describe storing facts for Dimensions hierarchy in one table samoidentifikujícím key.

25. Describe the imposition of facts on the hierarchy dimension in the hierarchy fact tables.

26. Describe the imposition of facts on the hierarchy dimension in the hierarchy fact tables using snowflakes.

27. Describe the imposition of facts on the hierarchy dimension in the hierarchy fact tables and explicitly specified

28. hypercube is and how it relates to the implementation of the data warehouse?

Why is the 29th hypercube sparsely occupied?

30. multikostky are and how they relate to hypercube?

31st Where aggregated data stored in different levels hypercube?

# 10 METADATA

For many reasons besides the data in the databases and data warehouses, data structure, content, features, and other information is needed to know. We call this "data data" metadata. It serves developers, administrators, users, and analytics.

This term, metadata, is already known from the classical information system. It is, for example, a database data dictionary that we create in a data model. Relational metadata (= table structure) include: attribute identifier, its data type, length, key, type of index, etc.

We also know that the SdBD maintains its own records of the database and its parts in the so-called system catalog. It also contains IS metadata. In addition to the above list of attributes, a list of tables and their properties, a list of indexes, defined reports and forms.

## 10.1 Data warehouse metadata

Naturally, DW will also need metadata, even more. In operational databases, metadata is basically hidden to end users. Only developers and database administrators work with them. Users work with the database through the user interface only as a black box.

However, analysts must be informed about the content of the data warehouse = its data structures in advance that they can use DW correctly and efficiently. Therefore, they must work with DW metadata, use them to search quickly required data and to interpret them correctly.

By content, we can divide the metadata for DW into several types:

**Metadata for the DW management**

Metadata for the DW management serves analysts, designers when developing the DW, and to DW administrators during the operation. They are:

- Metadata of source data for the need of analysis and design of DW:
  - deployment of databases on servers;
  - structure of the source databases;
  - structure and description of entities and their connection;
  - the definition and description attributes, their data types, domains, including units of measure, keys, indexes,
  - information on data ownership and any links between source data (one who gives
  - data).

- Data warehouse metadata - DW catalog contains:
  - a list of servers;
  - DW deployment of databases on servers;
  - definition of tables and views;
  - definition and description attributes, primary and foreign keys, indexes;
  - the distribution of attributes for dimension and hierarchy, facts and descriptive attributes, restrictions on questions;
  - definition of dimension tables and fact tables.

- Metadata for a data pump - mapping data source from operational databases to target atomic data in the primary data warehouse:
  - DW rules for each attribute: for its copying, integration functions, transformation rules, change of formats, verification, inference, limitation on questions;
  - units and conversion factors used between units, especially in the case of formula or time varying coefficients;
  - information on the timing of data transfers from DB to DW;
  - business rules and procedures, formulas for calculating the economic indicators, used formula and calculation procedures.

- Metadata for data and functions in the background DW:
  - temporary data structure for transformation, for presentation;
  - function for extraction and transformation, for quality assurance; the order of execution of these functions, program parameters,
  - the description of the strategy of DW fulfillment, the definition of temporary support tables and their functions.

- Architecture of DS – in the case of data marts:
  - data structure of DW and data marts;
  - definition of subsets of DM from DW;
  - order fulfillment DS and DM.

- Access rights and DW security:
  - Information on user roles and their rights;
  - Information about individual users and their roles.

### Metadata for end users

It serves users, whether specialized analysts or end users. Metadata for creating queries and for correct interpretation of the results belongs to this group:

- Content of data warehouse - a data structure in user-friendly form with the possibility of option; dimensions and their hierarchies, facts and their aggregated values;
- Data quality - all data must have information on their quality, which data reliable, warning if the data is incorrect or missing;
- Predefined queries and reports - used queries, catalog of output reports and graphs, the meaning of individual elements in the reports and in the results of analyzes, descriptions of methods and analyzes for users;
- Business rules and procedures - it is possible to use different formulas for calculating economic indicators, users must have information about used formulas and calculation procedures (for example for calculating of costs or profits, etc.).
- Status information - in daily operation in the case of daily replenishment of warehouse, data may be in a different stage of update, the user must be informed if DW contains only old data, or data is just in the process of actualization, or if new data is not yet available, or if DW already contains up to date data;
- Rules for data cleaning - rules when data can be removed from data warehouse or archive;
- History of data warehouse fulfillment - the history of the data warehouse fulfillment; all transfers should be recorded - transmitted volumes of data, reports on identified errors in the data, the time required for transmission and aggregation calculations; records of history to be synchronized with status information; update schedule should be available to users to be informed about when they become available new data.

### Metadata for optimization

These data can be used to optimize the design and performance of the data warehouse. These metadata include:

- Definitions of aggregations and their location - a description of the navigation between the D-tables and F-tables of a large warehouse in ROLAP to accelerate the access to required data;
- Restrictions on queries - for faster filling of DW, smaller capacity, faster acces to data;
- Statistics of data warehouse - monitoring the frequency of different types of queries over the data warehouse; this information is feedback to database administrator, who may identify which data are frequently used and which are not, and adjust accordingly the DW content.

### Metadata as the basis for automation of supporting processes

They can serve as a basis for future automation of functions previously tailored for DW. Usually, they are continuously logged:

- Metadata for the extraction and transformation – the assigning of data source to target can serve as basis for generating scripts for extraction, transformation and integration;

- Data quality - users can enter permissible values for the different attributes, it is used to detect bugs with possible subsequent automatic correction;
- Generating queries - data structure is recorded to generate custom SQL queries;
- End tools - tools for presentation of structure of tables or content of summation tables.

## 10.2 Metadata standardization

Sharing and exchanging metadata is one of the important issues in the design of data warehouse. An effort to standardize metadata exists among database vendors.

These efforts can be seen in three levels:

- Generaly used repository of metadata are created (for example Platinum Repository, Microsoft Repository, Unisys UREP, ...); they can be used for any DW and therefore access to them should be standardized from most data warehouses;

• Standards for data exchange are defined (for example of the CWMI - Common Warehouse Metadata Interchange from OMG - Object Management Group – IBM, Oracle, Unisys; MDIS - Meta Data Interchange Specification from the MDC - Meta Data Coalition - Microsoft). Although everyone has their own metadata, it is possible to create a conversion scripts in and out of standardized metadata, and in this manner, it is possible to communicate with data warehouses of other vendors;

• Open API for products - most vendors provide data warehousing open API for third-party products and applications to their metadata (for example the Hyperion Integration Server, IBM Meta Data Interchange Language ).

# 11 TECHNOLOGIES FOR THE IMPLEMENTATION DS

**Methods and tools**

High volumes of data in DW and complicated nature of their use compared to operational databases also lead to other physical models of a database and for other technologies of access to data. Technology specific for data warehouses or technology known from other data storage are used.

A key requirement is the maximum speed of access to data. This is currently achieved by several ways and their combinations:

- Incremental fulfillment of warehouse and calculating aggregations;
- Precomputation and storing of predicted aggregation;
- Dividing the data warehouse into smaller data marts;
- Using special indexing technology;
- Using parallel data access;

Precomputation and storage of aggregated data, we discussed in the previous chapter, we also discussed data marts. These problems belong to database modeling. Other methods will be discussed in the following paragraphs:

**Incremental fulfillment of data warehouse**

We reported that the data warehouse is initially filled once by data from archives and other older sources or from current operational databases. Then data warehouse is fulfilled with the help of data pump periodically with new data, arising in operational databases since the last fulfilling. The period may be of different lengths, from day cycle (for example, business data) to very long interval (e.g., with annual data intended for longer-term research in the medical sector or elsewhere).

Especially for data fulfilled per day, it is necessary to optimize the work of data pump.

The calculation of aggregate values takes the longest time. As the total amount of data is expanding, increasing volumes of data must be processed. Very much time can be saved when an appropriate incremental method of calculation of aggregate functions is used:

a new sum = sum of the previous + sum of increment

new minimum = min (previous minimum, minimum of increment)

Similarly, amount and maximum can be calculated. It is not necessary to sum all again. Only new means must be calculated. Counts, sums, extremes of increments are counted continuously during data conversion, the foregoing values are stored in the DW.

### Indexing technology

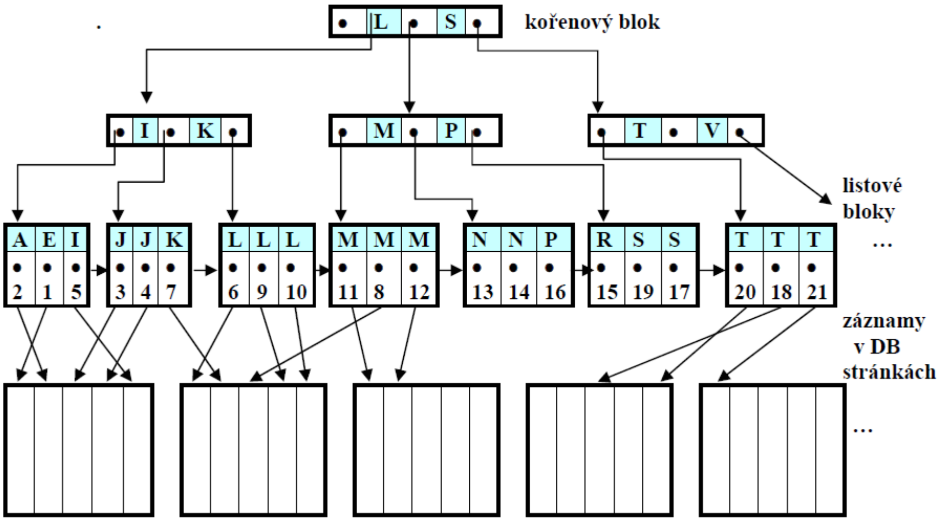
Several types of indexes are used in DW. Some of them are know from relational databases others are new.

- B + trees (the are used generally in relational DBMS);
- Binary index matrix (used in DW and OLAP);
- Join indexes (solution is a part of multidimensional analysis);
- R trees and bitmap indexes.

### B + tree (balanced tree)

One or more columns are indexing with the help of multilevel data structure comprising a root node with pointers to nodes in the next level. The lowest level contains blocks of leaves for each row of indexed table.

If leaves are linked by pointers, the sequential table traversing is enable, querying on intervals, sort by index key without the traversing through the whole B tree – then the structure is called B + trees.



B+ tree

Other possibilities are:

▪ Trimming of index values in non-leaves blocks with long chained columns;

▪ Preservation of primary key for queries involving both attributes (although the index is effective when its record is much shorter in comparison to records from data tables);

▪ Combining multiple indexes based on complex queries using selective conditions AND / OR, creating of several temporary simple indexes, logically to combine them according to conditions in a SELECT from and only to accede to data rows in conclusion;

▪ SQL optimizers process with star schema initially limiting conditions over dimension tables, and eventually join them to fact table.

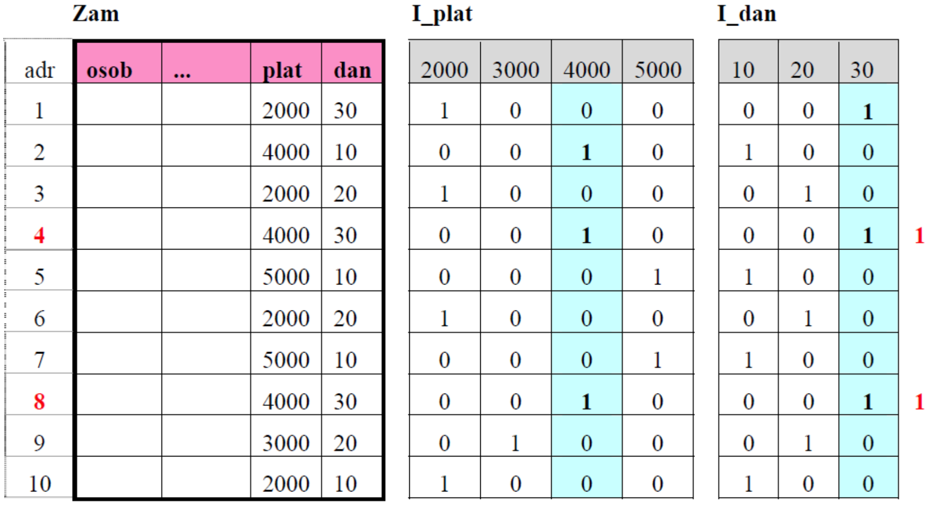
These technologies are used by current DBMS manufacturers - IBM, Oracle, Sybase, etc.

### Binary index matrix

For secondary attributes with a small domain (the small number of possible values), sometimes the indexing with the help of binary matrix is used. This method can save capacity. In the index file, there is not the only attribute indexed, but all its possible values. The value of attribute is wrote down by the position of a unity bit in a sequence that has as many bits as the number of values.

Example:

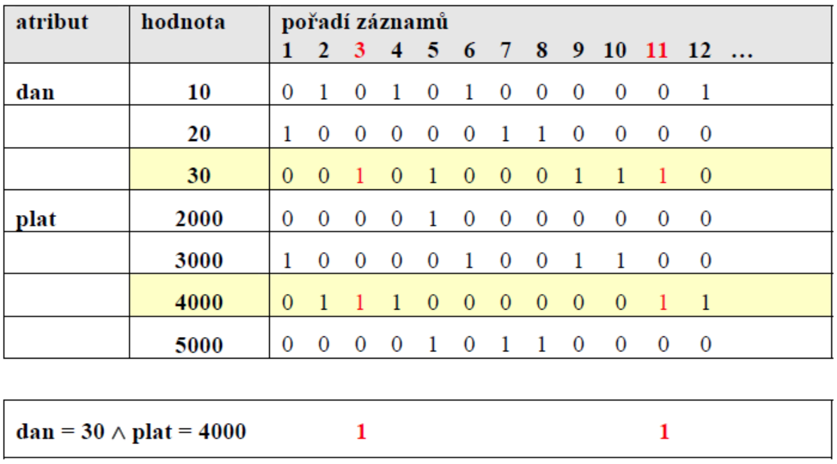
Let us have a set of employees with personal numbers, salary (4 values) and the percentage of taxes (3 values). Then the appropriate indexes are as follows:



| Zam | | | | | I_plat | | | | I_dan | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| adr | osob | ... | plat | dan | 2000 | 3000 | 4000 | 5000 | 10 | 20 | 30 | |
| 1 | | | 2000 | 30 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 2 | | | 4000 | 10 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | |
| 3 | | | 2000 | 20 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 4 | | | 4000 | 30 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 5 | | | 5000 | 10 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |
| 6 | | | 2000 | 20 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 7 | | | 5000 | 10 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |
| 8 | | | 4000 | 30 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 9 | | | 3000 | 20 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 10 | | | 2000 | 10 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | |

If we are looking for employees with a salary of 4000 and a tax of 30, the corresponding columns in the one index and the second index are searched for one. The order of the record in the data table is determined to them.

Even more advantageous is to store the index transposed - column entries and binary vectors for the same attribute value in rows. Then it works easily with whole binary vectors.



| atribut | hodnota | pořadí záznamů | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | … |
| dan | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | |
| | 20 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | |
| | 30 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | |
| plat | 2000 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 3000 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | |
| | 4000 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| | 5000 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dan = 30 ∧ plat = 4000 | | | 1 | | | | | | | | 1 | |

Binary matrices are especially useful when secondary attribute values do not change when the records are not changed, or they are only added serially to the end of the file. Another advantage of binary matrices is the easy realization of combined queries using logical operators of negation, conjunctions, and disjunctions.

Thus, the auxiliary index structure uses the bit as a flag for the attribute value of an attribute to that entity. The number of bitmap columns corresponds to the cardinality of the indexed column. Use of bitmap indexes is not beneficial for INSERT and UPGRADE operations. However, as DW reads only, its use is advantageous in this case. This technology is used, for example, by Oracle, Sybase.

### Indexes to support connections (Join Index)

For very complex queries, involving many join operations (Join) on large databases, the classical link algorithms used are not fast enough. There is a large number of connections in data warehouses - the fact table merges with a series of dimensional tables to get both the desired dimension values and user-friendly descriptions of attributes, not just their id.

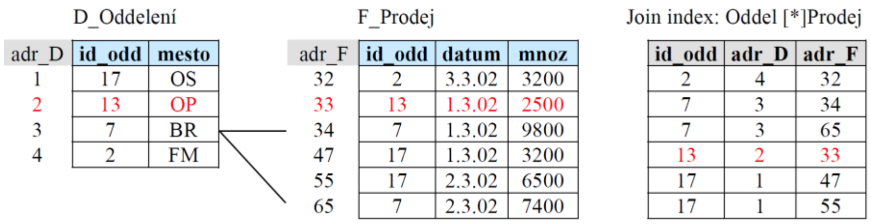
Therefore, a new data structure - an index for link support - was created specifically to support the connection operation. This is a spreadsheet that consists of two or more columns. It contains, in addition to the index, the addresses of the corresponding records in 2 or more linked tables according to the indexed link.

Take F-table F with line addresses adr_F and dimensional D-table D with adr_D rows, the connection is performed using the id_D attribute. Then the index file has the id_D, adr_F, adr_D columns, sorted for binary search by id_D. When the id_D is searched, the corresponding addresses on the row searched indicate the line numbers at the same time in both tables F and D. Additionally, by creating a B + tree above this index, we get quick access to more complex queries.

Example:

Join index for connection of dimensional tables Department and fact table Sales by the connection condition:

Oddelení.id_odd = Prodej.id_odd



D_Oddelení

| adr_D | id_odd | mesto |
|---|---|---|
| 1 | 17 | OS |
| 2 | 13 | OP |
| 3 | 7 | BR |
| 4 | 2 | FM |

F_Prodej

| adr_F | id_odd | datum | mnoz |
|---|---|---|---|
| 32 | 2 | 3.3.02 | 3200 |
| 33 | 13 | 1.3.02 | 2500 |
| 34 | 7 | 1.3.02 | 9800 |
| 47 | 17 | 1.3.02 | 3200 |
| 55 | 17 | 2.3.02 | 6500 |
| 65 | 7 | 2.3.02 | 7400 |

Join index: Oddel [*]Prodej

| id_odd | adr_D | adr_F |
|---|---|---|
| 2 | 4 | 32 |
| 7 | 3 | 34 |
| 7 | 3 | 65 |
| 13 | 2 | 33 |
| 17 | 1 | 47 |
| 17 | 1 | 55 |

To access the Odds and Sales records for the required id_odd = 13, we will obtain the value 13 in the index (red lines), where we will find the addresses of the corresponding records in both data tables. Or, if we need a sorted listing of the entire Sales table, including a city from the Department table, we sequentially browse the index file and read the addresses of the data tables directly according to the addresses.

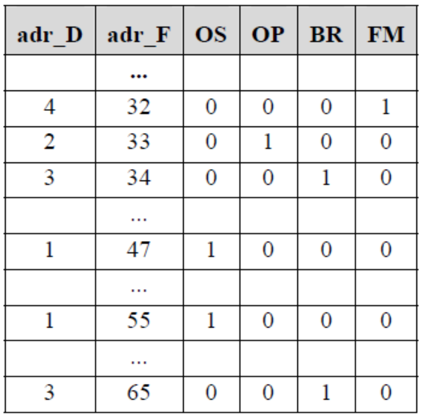
### The combined index

By combining the principles of bitmap indexes and join indexes we get a combined index.

This index is similar to previous index, but a fact table is associated with the dimensional table so that bitmap index is constructed to descriptively secondary attribute. Thus, it is possible to access the facts associated with the dimensions and limit values of attributions by using bit operations.

Example:

Combined join index for connection of dimensional tables Department and fact table Sales indexed by binary index for the atribute city.



| adr_D | adr_F | OS | OP | BR | FM |
|---|---|---|---|---|---|
| | ... | | | | |
| 4 | 32 | 0 | 0 | 0 | 1 |
| 2 | 33 | 0 | 1 | 0 | 0 |
| 3 | 34 | 0 | 0 | 1 | 0 |
| | ... | | | | |
| 1 | 47 | 1 | 0 | 0 | 0 |
| | ... | | | | |
| 1 | 55 | 1 | 0 | 0 | 0 |
| | ... | | | | |
| 3 | 65 | 0 | 0 | 1 | 0 |

### R trees

To support spatial data, special modification of B tree was designed. It ensure efficient access to the planar (2D) or space (3D) objects in a relational or object-relational database. Information on borders of the respective object are in leaves outside id_rows. In blocks of higher levels, information about borders of unified lower-level objects are kept. For 2D planar objects, it is, for example, the coordinates [x, y] for the lower-right and upper-left corner of one object or unification of multiple objects.

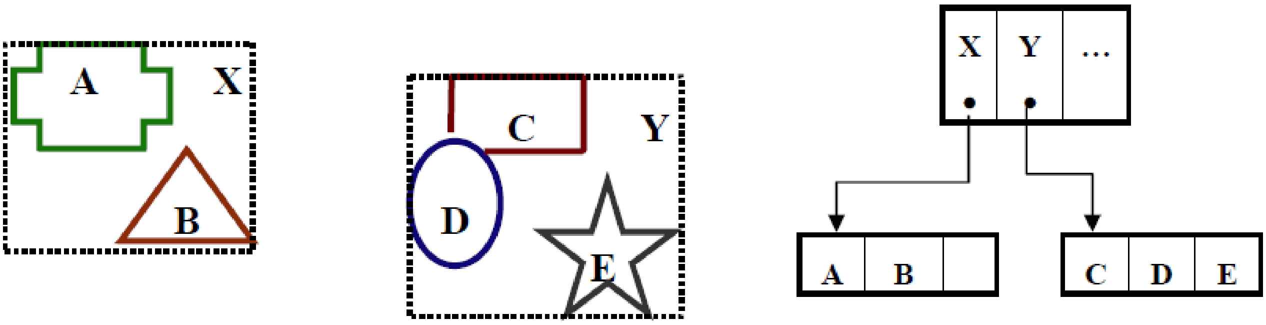


Figure 11.1 Planar objects and their R-tree

More used variant of R trees are bitmapped indexes (also grid indexes). Mapped space is split by grille, its individual blocks are numbered and therefore classifiable. Blocks of the first level are split aging by the forming a second thicker

Grid. Thus, a variant of B tree, suitable for access to the spatial data I screated.

### Parallelism

For efficient operation of the DW, parallel technologies are used. Two methods are used for parallel approach:

1. Distribution of database into parallel accessible parts (partitioning)
   - horizontal division - a disjoint set of rows is created by selecting a table; these are placed into special fragments; the main candidate for division is the chosen dimension and time; negative factors are "hot spots", fragments overused against other fragments (such as the last time period); the advantage is the possibility of distributing the respective indexes, which increases the response rate;
   - Vertical division - the fact sheet is divided into multiple fragments, which are linked by a redundant primary key; The advantage is faster access to data for users who are only interested in the allocated data (shorter records, higher number of records per page).
2. Symmetric multiprocessing / massive parallel processing (SMP / MPP) Real parallel processing of one application on

multiple processors simultaneously by splitting a job into threads; identical processors accessing shared memory are used in SMP; MPP can contain different processors with its own operating memory and are thus collaborating computers.

Two separate types of processes - data pump and OLAP queries, or separate queries of different users, make it especially important to process DW in parallel.

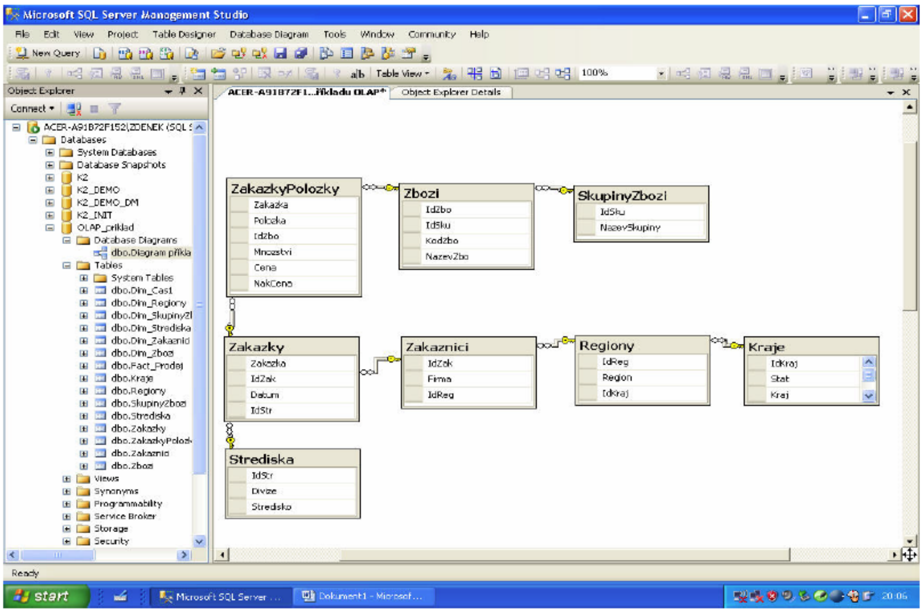
# 12 SW AND USE DATA WAREHOUSES

Data warehouse in MS SQL Server 2005

Many large DBMS supports outside of classical information systems also data warehouses and OLAP. We will show them to create and use a small example implemented in MS SQL Server 2008.

Example:

We have the source database from the following diagram:

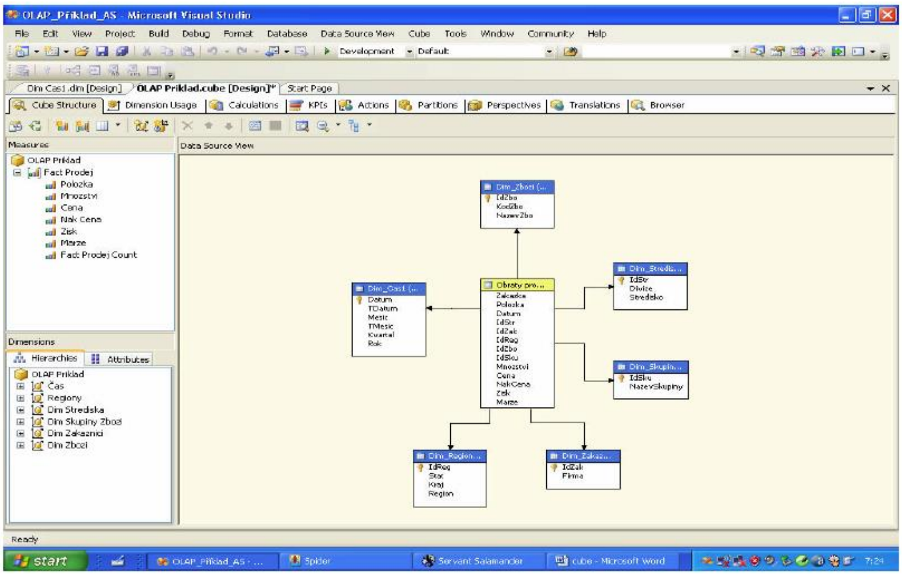


Data warehouse model derived from this database defines:

Facts: the number of items, the sum of quantity, price, cost price, profit margins,

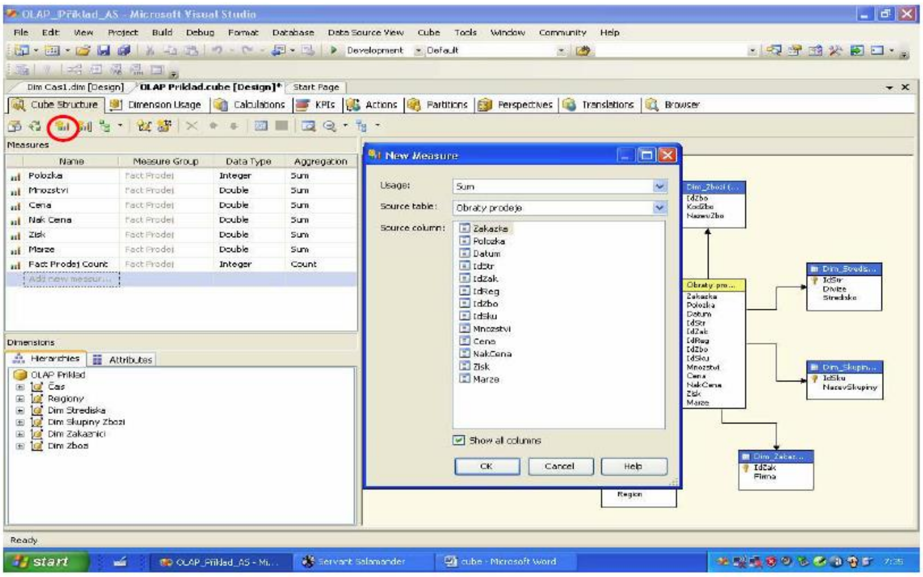
Dimensions : time, regions, products, product groups, branch office, customers.

The logical structure of the star model is as follows:



The definition of the data warehouse is carried out with the help pf classically defined tables using commands CREATE TABLE:

Definitions facts by the icon New Measures (marked red), by the selecting of aggregate functions, by the selecting of the source table and by selecting of the source attribute in the window New Measures:

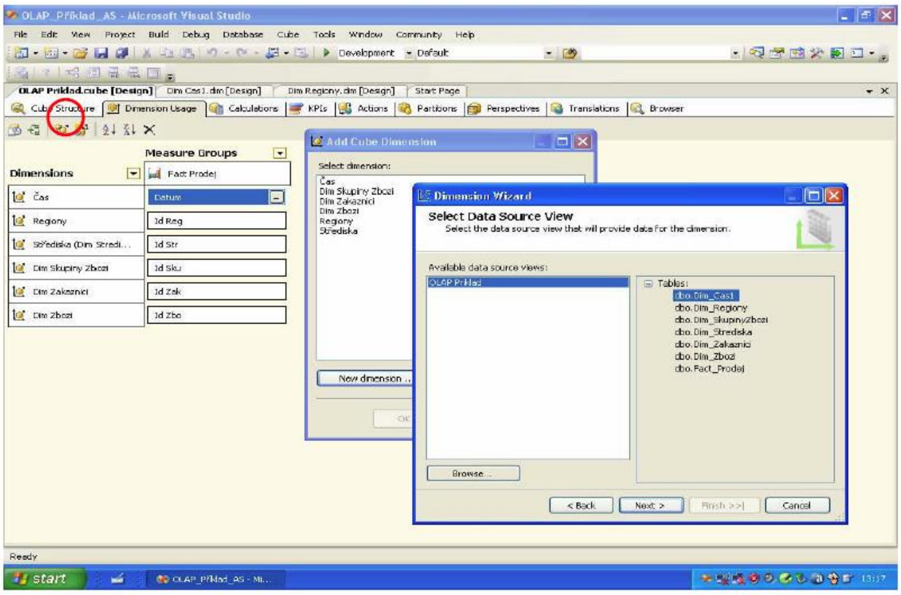


Measures

| Name | Measure Group | Data Type | Aggregation |
|---|---|---|---|
| Polozka | Fact Prodej | Integer | Sum |
| Mnozstvi | Fact Prodej | Double | Sum |
| Cena | Fact Prodej | Double | Sum |
| Nak Cena | Fact Prodej | Double | Sum |
| Zisk | Fact Prodej | Double | Sum |
| Marze | Fact Prodej | Double | Sum |
| Fact Prodej Count | Fact Prodej | Integer | Count |
| Add new measur... | | | |

Usage: Sum
Source table: Obraty prodeje
Source column: Zakazka, Polozka, Datum, IdStr, IdZak, IdReg, IdZbo, IdSku, Mnozstvi, Cena, NakCena, Zisk, Marze
☑ Show all columns

We perform the definition of dimensions by the icon Add Cube Dimension , down by selecting New Dimension and by the selecting dimensional table:



| Dimensions | Measure Groups |
|---|---|
| | Fact Prodej |
| Čas | Datum |
| Regiony | Id Reg |
| Střediska (Dim Středi... | Id Str |
| Dim Skupiny Zbozi | Id Sku |
| Dim Zakaznici | Id Zak |
| Dim Zbozi | Id Zbo |

Add Cube Dimension

Select dimension:

Čas
Dim Skupiny Zbozi
Dim Zakaznici
Dim Zbozi
Regiony
Střediska

Dimension Wizard

**Select Data Source View**

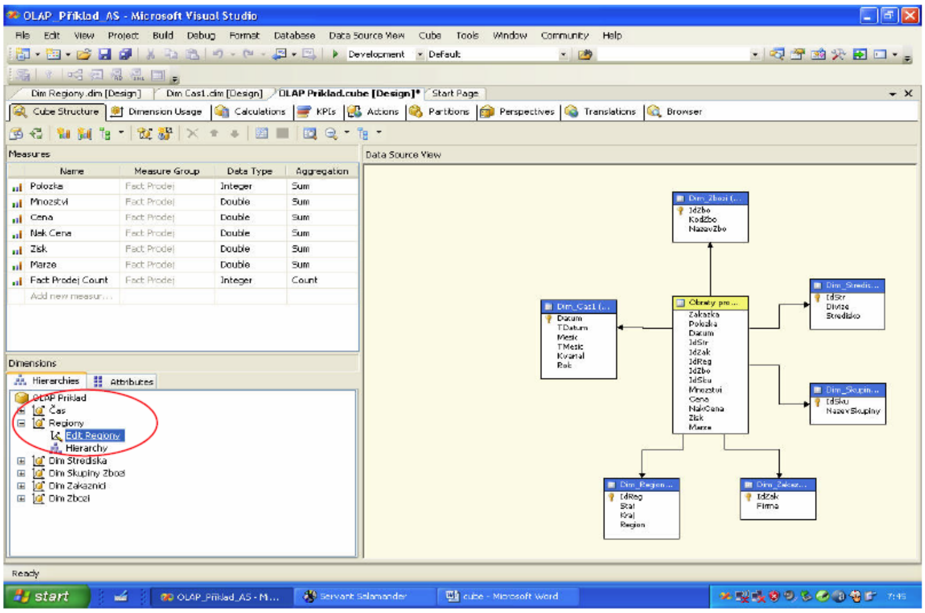
Select the data source view that will provide data for the dimension.

Available data source views:

OLAP Priklad

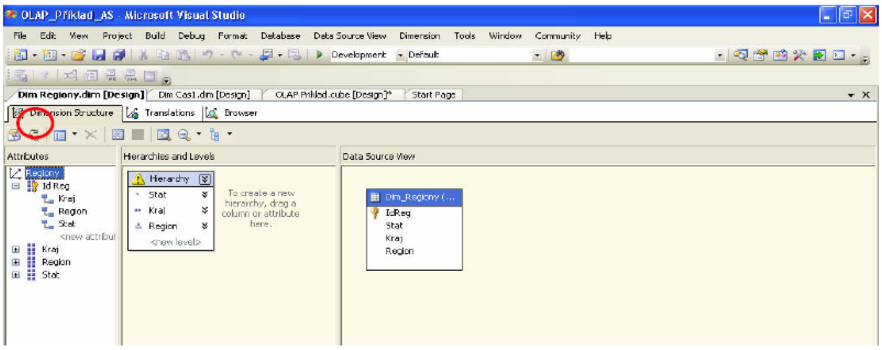
Tables:
- dbo.Dim_Cas1
- dbo.Dim_Regiony
- dbo.Dim_SkupinyZbozi
- dbo.Dim_Strediska
- dbo.Dim_Zakaznici
- dbo.Dim_Zbozi
- dbo.Fact_Prodej

Definition of a hierarchy of dimension by the editing of selected dimension



| Name | Measure Group | Data Type | Aggregation |
|---|---|---|---|
| Polozka | Fact Prodej | Integer | Sum |
| Mnozstvi | Fact Prodej | Double | Sum |
| Cena | Fact Prodej | Double | Sum |
| Nak Cena | Fact Prodej | Double | Sum |
| Zisk | Fact Prodej | Double | Sum |
| Marze | Fact Prodej | Double | Sum |
| Fact Prodej Count | Fact Prodej | Integer | Count |
| Add new measur... | | | |

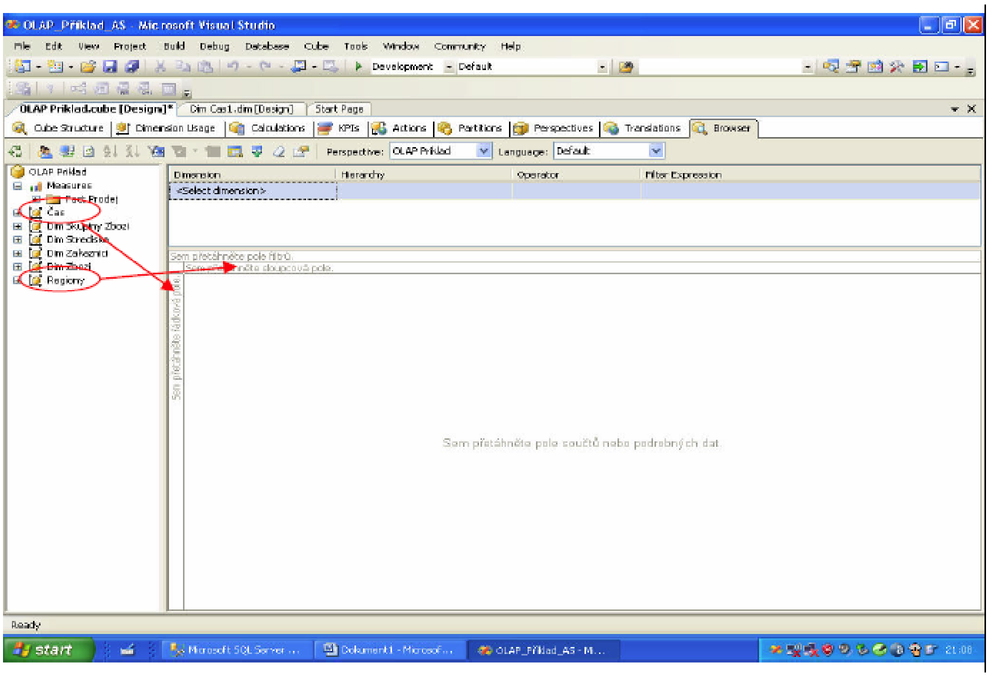
The adding of a hierarchy and respective descriptive attributes:



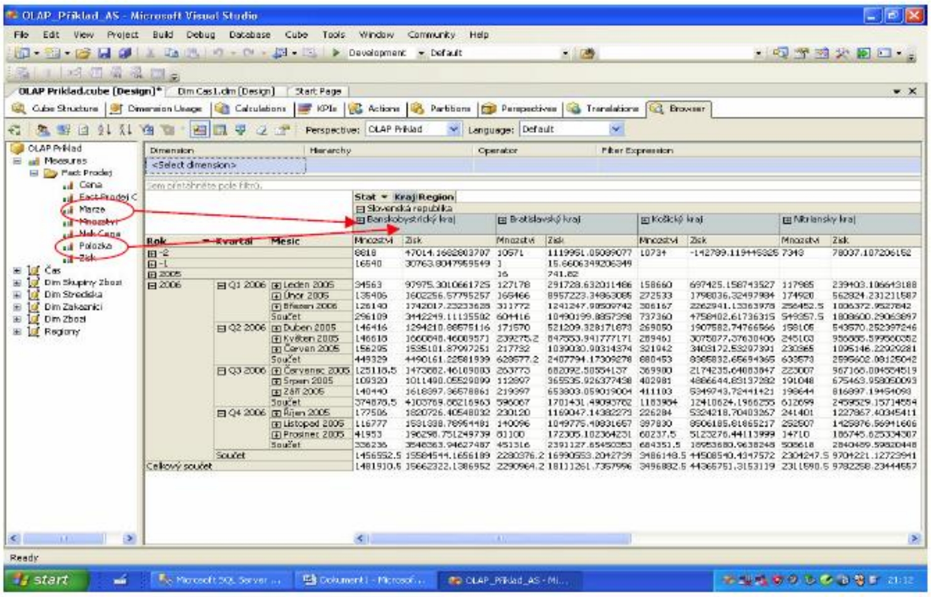
A cube is created after the defining its structure with the icon Process, the set aggregation are counted.

After assembly and filling cubes (data warehouse), it is possible to view any content defined by a user with selected set of dimensions and facts.

We select the tab Browser and get a blank area with help where we can drag from left selected dimensions – we can click their hierarchical structure:



and selected facts:


OLAP_Příklad_AS - Microsoft Visual Studio
File Edit View Project Build Debug Database Cube Tools Window Community Help
Development Default
OLAP Priklad.cube [Design]* Dim Cas1.dim [Design] Start Page
Cube Structure Dimension Usage Calculations KPIs Actions Partitions Perspectives Translations Browser
OLAP Příklad
Measures
Fact Prodej
Cena
Fact Prodej C
Marze
Mnozstvi
Nak Cena
Polozka
Zisk
Čas
Dim Skupiny Zboží
Dim Strediska
Dim Zakaznici
Dim Zbozi
Regiony
Ready
start
Microsoft SQL Server ...
Dokument1 - Microsof...
OLAP_Příklad_AS - Mi...
21:12

Perspective: OLAP Příklad  Language: Default

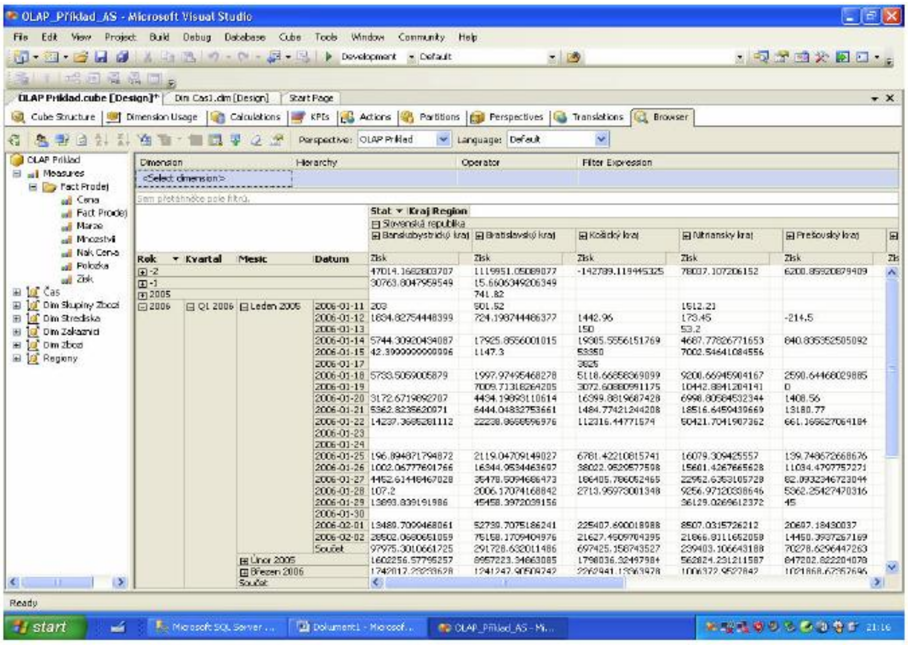
| Dimension | Hierarchy | Operator | Filter Expression |
|---|---|---|---|
| <Select dimension> | | | |

Sem přetáhněte pole filtrů.

| | | | Stat ▾ Kraj Region | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | ⊟ Slovenská republika | | | | | | | |
| | | | ⊞ Banskobystrický kraj | | ⊞ Bratislavský kraj | | ⊞ Košický kraj | | ⊞ Nitriansky kraj | |
| Rok | ▾ Kvartal | Mesic | Mnozstvi | Zisk | Mnozstvi | Zisk | Mnozstvi | Zisk | Mnozstvi | Zisk |
| ⊞ -2 | | | 8818 | 47014.1682803707 | 10571 | 1119951.05089077 | 10734 | -142789.119445325 | 7348 | 78037.107206152 |
| ⊞ -1 | | | 16540 | 30763.8047959549 | 1 | 15.6606349206349 | | | | |
| ⊞ 2005 | | | | | 16 | 741.82 | | | | |
| ⊟ 2006 | ⊟ Q1 2006 | ⊞ Leden 2005 | 34563 | 97975.3010661725 | 127178 | 291728.632011486 | 158660 | 697425.158743527 | 117985 | 239403.106643188 |
| | | ⊞ Únor 2005 | 135406 | 1602256.57795257 | 165466 | 8957223.34863085 | 272533 | 1798036.32497984 | 174920 | 562824.231211587 |
| | | ⊞ Březen 2006 | 126140 | 1742017.23233628 | 311772 | 1241247.90509742 | 306167 | 2262941.13363978 | 256452.5 | 1006372.9527842 |
| | | Součet | 296109 | 3442249.11135502 | 604416 | 10490199.8857398 | 737360 | 4758402.61736315 | 549357.5 | 1808600.29063897 |
| | ⊟ Q2 2006 | ⊞ Duben 2005 | 146416 | 1294210.88575116 | 171570 | 521209.328171873 | 269050 | 1907582.74766566 | 158106 | 543570.252397246 |
| | | ⊞ Květen 2005 | 146618 | 1660848.46089571 | 239275.2 | 847553.941777171 | 289461 | 3075877.37630406 | 245103 | 956885.599560352 |
| | | ⊞ Červen 2005 | 156295 | 1535101.87997251 | 217792 | 1039030.90314374 | 321942 | 3403172.53297391 | 230365 | 1095146.22929281 |
| | | Součet | 449329 | 4490161.22581939 | 628677.2 | 2407794.17309278 | 880453 | 8385832.65694365 | 633573 | 2595602.08125042 |
| | ⊟ Q3 2006 | ⊞ Červenec 2005 | 125118.5 | 1473882.46109083 | 263773 | 682092.50554137 | 369900 | 2174235.64083847 | 223007 | 967165.004554519 |
| | | ⊞ Srpen 2005 | 109320 | 1011490.05529099 | 112897 | 365535.926377438 | 402981 | 4886644.83137282 | 191048 | 675463.958060093 |
| | | ⊞ Září 2006 | 140440 | 1618397.36578861 | 219397 | 653803.059019004 | 411103 | 5349743.72441421 | 198644 | 816897.19454093 |
| | | Součet | 374878.5 | 4103769.88216983 | 596067 | 1701431.49093762 | 1183984 | 12410624.1966255 | 612699 | 2459529.15714554 |
| | ⊟ Q4 2006 | ⊞ Říjen 2005 | 177506 | 1820726.40548032 | 230120 | 1169047.14382273 | 226284 | 5324218.70403267 | 241401 | 1227867.40345411 |
| | | ⊞ Listopad 2005 | 116777 | 1581338.78954481 | 140096 | 1049775.40831657 | 397830 | 8506185.81865217 | 252507 | 1425876.56941606 |
| | | ⊞ Prosinec 2005 | 41953 | 196298.751249739 | 81100 | 172305.102364231 | 60237.5 | 5123276.44113999 | 14710 | 186745.625334307 |
| | | Součet | 336236 | 3548363.94627487 | 451316 | 2391127.65450353 | 684351.5 | 18953680.9638248 | 508618 | 2840489.59820448 |
| | Součet | | 1456552.5 | 15584544.1656189 | 2280376.2 | 16990553.2042739 | 3486148.5 | 44508540.4347572 | 2304247.5 | 9704221.12723941 |
| Celkový součet | | | 1481910.5 | 15662322.1386952 | 2290964.2 | 18111261.7357996 | 3496882.5 | 44365751.3153119 | 2311590.5 | 9782258.23444657 |

The result here is a two-dimensional table (time, region) with 2-dimensional facts (amount, profit).

We can get a more detailed level (January ⇒ days) by a simple click on higher level dimension:


OLAP_Příklad_AS - Microsoft Visual Studio
File Edit View Project Build Debug Database Cube Tools Window Community Help
Development Default
OLAP Priklad.cube [Design]* Dim Cas1.dim [Design] Start Page
Cube Structure Dimension Usage Calculations KPIs Actions Partitions Perspectives Translations Browser
OLAP Příklad
Measures
Fact Prodej
Cena
Fact Prodej
Marze
Mnozstvi
Nak Cena
Polozka
Zisk
Čas
Dim Skupiny Zboží
Dim Strediska
Dim Zakaznici
Dim Zbozi
Regiony
Ready
start
Microsoft SQL Server ...
Dokument1 - Microsof...
OLAP_Příklad_AS - Mi...
21:16

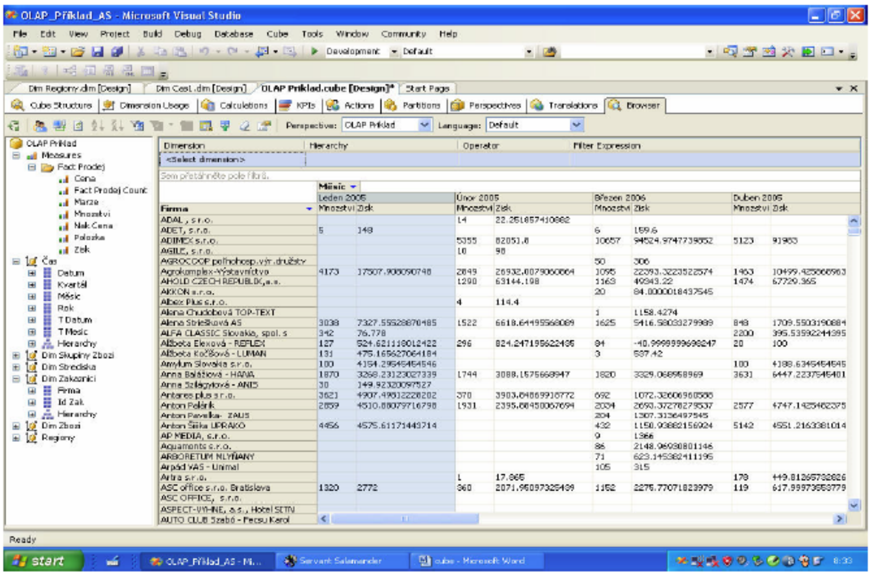
Perspective: OLAP Příklad  Language: Default

| Dimension | Hierarchy | Operator | Filter Expression |
|---|---|---|---|
| <Select dimension> | | | |

Sem přetáhněte pole filtrů.

| | | | | Stat ▾ Kraj Region | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | ⊟ Slovenská republika | | | | | |
| | | | | ⊞ Banskobystrický kraj | ⊞ Bratislavský kraj | ⊞ Košický kraj | ⊞ Nitriansky kraj | ⊞ Prešovský kraj | ⊞ |
| Rok | ▾ Kvartal | Mesic | Datum | Zisk | Zisk | Zisk | Zisk | Zisk | Zis |
| ⊞ -2 | | | | 47014.1682803707 | 1119951.05089077 | -142789.119445325 | 78037.107206152 | 6200.85920879409 | |
| ⊞ -1 | | | | 30763.8047959549 | 15.6606349206349 | | | | |
| ⊞ 2005 | | | | | 741.82 | | | | |
| ⊟ 2006 | ⊟ Q1 2006 | ⊟ Leden 2005 | 2006-01-11 | 203 | 901.52 | | 1512.21 | | |
| | | | 2006-01-12 | 1834.82754448399 | 724.198744486377 | 1442.96 | 173.45 | -214.5 | |
| | | | 2006-01-13 | | | 150 | 53.2 | | |
| | | | 2006-01-14 | 5744.30920434087 | 17925.8556001015 | 19305.5556151769 | 4687.77826771653 | 840.835352505092 | |
| | | | 2006-01-15 | 42.3999999999996 | 1147.3 | 53350 | 7002.54641084556 | | |
| | | | 2006-01-17 | | | 3825 | | | |
| | | | 2006-01-18 | 5733.5059005879 | 1997.97495468278 | 5118.66858369099 | 9200.66945904167 | 2590.64468029885 | |
| | | | 2006-01-19 | | 7009.71318264205 | 3072.60880991175 | 10442.8841204141 | 0 | |
| | | | 2006-01-20 | 3172.6719892707 | 4434.19893110614 | 16399.8819687428 | 6998.80584532344 | 1408.56 | |
| | | | 2006-01-21 | 5362.8235620871 | 6444.04832753661 | 1484.77421244208 | 18516.6459439669 | 13180.77 | |
| | | | 2006-01-22 | 14237.3685281112 | 22238.8658596976 | 112316.44771574 | 50421.7041907362 | 661.169627064184 | |
| | | | 2006-01-23 | | | | | | |
| | | | 2006-01-24 | | | | | | |
| | | | 2006-01-25 | 196.894871794872 | 2119.04709149027 | 6781.42210815741 | 16079.309425557 | 139.748672668676 | |
| | | | 2006-01-26 | 1002.06777601766 | 16344.9534463607 | 38022.9520577508 | 15601.4267665628 | 11034.4797757271 | |
| | | | 2006-01-27 | 4452.61448467028 | 35478.5094686473 | 186405.786052465 | 22952.6353105728 | 82.0932346723044 | |
| | | | 2006-01-28 | 107.2 | 2006.17074168842 | 2713.95973001348 | 9256.97120338646 | 5362.25427470316 | |
| | | | 2006-01-29 | 13893.839191986 | 45458.3972039156 | | 36129.0269612372 | 45 | |
| | | | 2006-01-30 | | | | | | |
| | | | 2006-02-01 | 13480.7099468061 | 52739.7075186241 | 225407.690018988 | 8507.0315726212 | 20687.18430037 | |
| | | | 2006-02-02 | 28502.0680651059 | 75158.1709404976 | 21627.4909704395 | 21866.8111652058 | 14450.3937267169 | |
| | | | Součet | 97975.3010661725 | 291728.632011486 | 697425.158743527 | 239403.106643188 | 70278.6296447263 | |
| | | ⊞ Únor 2005 | | 1602256.57795257 | 8957223.34863085 | 1798036.32497984 | 562824.231211587 | 847202.822204078 | |
| | | ⊞ Březen 2006 | | 1742017.23233628 | 1241247.90509742 | 2262941.13363978 | 1006372.9527842 | 1021868.67357696 | |
| | | Součet | | | | | | | |

Similarly, you can select any other combination and location of dimensions and facts:



OLAP_Příklad_AS - Microsoft Visual Studio

Perspective: OLAP Priklad
Language: Default

- OLAP Priklad
  - Measures
    - Fact Prodej
      - Cena
      - Fact Prodej Count
      - Marze
      - Mnozstvi
      - Nak Cena
      - Polozka
      - Zisk
  - Čas
    - Datum
    - Kvartál
    - Měsíc
    - Rok
    - T Datum
    - T Mesic
    - Hierarchy
  - Dim Skupiny Zbozi
  - Dim Strediska
  - Dim Zakaznici
    - Firma
    - Id Zak
    - Hierarchy
  - Dim Zbozi
  - Regiony

| Dimension | Hierarchy | Operator | Filter Expression |
|---|---|---|---|
| <Select dimension> | | | |

Sem přetáhněte pole filtrů.

| | Měsíc | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Leden 2006 | | Únor 2005 | | Březen 2006 | | Duben 2005 | |
| Firma | Mnozstvi | Zisk | Mnozstvi | Zisk | Mnozstvi | Zisk | Mnozstvi | Zisk |
| ADAL , s r.o. | | | 14 | 22.251857410882 | | | | |
| ADET, s.r.o. | 5 | 148 | | | 6 | 159.6 | | |
| ADIMEX s.r.o. | | | 5355 | 82051.8 | 10657 | 94524.9747739852 | 5123 | 91983 |
| AGILE, s.r.o. | | | 10 | 98 | | | | |
| AGROCOOP polnohosp.výr.družstv | | | | | 50 | 306 | | |
| Agrokomplex-Výstavníctvo | 4173 | 17507.908090748 | 2849 | 26932.0079060864 | 1095 | 22393.3223522574 | 1463 | 10499.425866963 |
| AHOLD CZECH REPUBLIK,a.s. | | | 1290 | 63144.198 | 1163 | 49343.22 | 1474 | 67729.365 |
| AKKON s.r.o. | | | | | 20 | 84.0000018437545 | | |
| Albex Plus s.r.o. | | | 4 | 114.4 | | | | |
| Alena Chudobová TOP-TEXT | | | | | 1 | 1158.4274 | | |
| Alena Strieškova AS | 3038 | 7327.55528870485 | 1522 | 6618.64495568089 | 1625 | 5416.58033279989 | 848 | 1709.5503190884 |
| ALFA CLASSIC Slovakia, spol. s | 342 | 76.778 | | | | | 2200 | 395.53592244395 |
| Alžbeta Elexová - REFLEX | 127 | 524.621118012422 | 296 | 824.247195622435 | 84 | -40.9999999698247 | 20 | 100 |
| Alžbeta Kočšová - LUMAN | 131 | 475.165627064184 | | | 3 | 537.42 | | |
| Amylum Slovakia s.r.o. | 100 | 4154.29545454546 | | | | | 100 | 4188.6345454545 |
| Anna Balážiová - HANA | 1870 | 3268.23123027339 | 1744 | 3088.1575668947 | 1820 | 3329.068958969 | 3631 | 6447.2237545401 |
| Anna Szilágyiová - ANIS | 30 | 149.92320097527 | | | | | | |
| Antares plus s.r.o. | 3621 | 4907.49812228202 | 370 | 3903.64869916772 | 692 | 1072.32606960586 | | |
| Anton Pelárik | 2859 | 4510.88079716798 | 1931 | 2395.88450067694 | 2034 | 2690.37278279537 | 2577 | 4747.1425462375 |
| Anton Pavelka- ZAUS | | | | | 204 | 1307.3136497545 | | |
| Anton Šiška UPRAKO | 4456 | 4575.61171443714 | | | 432 | 1150.93882156924 | 5142 | 4551.2163381014 |
| AP MEDIA, s.r.o. | | | | | 9 | 1366 | | |
| Aquamonts s.r.o. | | | | | 86 | 2148.96930801146 | | |
| ARBORETUM MLYŇANY | | | | | 71 | 623.145382411195 | | |
| Arpád VAS - Unimal | | | | | 105 | 315 | | |
| Artra s.r.o. | | | 1 | 17.865 | | | 178 | 449.81265732826 |
| ASC office s.r.o. Bratislava | 1320 | 2772 | 360 | 2071.96097329409 | 1152 | 2275.77071823979 | 119 | 617.99973653779 |
| ASC OFFICE, s.r.o. | | | | | | | | |
| ASPECT-VYHNE, a.s., Hotel SITN | | | | | | | | |
| AUTO CLUB Szabó - Pecsu Karol | | | | | | | | |

Ready

# 13 DATA MINING

Data mining, more generally predictive modeling, is a sophisticated method of data analysis. Its purpose is primarily to detect important dependencies in the data, respectively, formulations (such as client behavior) and anticipate possible changes or consequences.

There are extensive internal and external data sources available to companies: client transactions (purchasing, cash), demographics about clients, external debtors' registers, etc. Not so long ago, according to some estimates, organizations have used only about 10% they have even generated themselves over the years. And in some areas, such as banking or telecommunications, the bulk of business takes place through information systems, i.e., it generates huge amounts of data. The low data usage was largely due to our low ability to transform data into useful information, knowledge, and then use this information correctly for action.

In an environment of growing competition and globalization of markets, the use of all available information (of course, taking into account restrictions imposed by, for example, legal protection of personal data) becomes an actual but also a critical issue in many areas of business. The possibility of using it, thanks to economic reality, is a necessary condition of survival.

Data mining is a process defined as non-trivial extraction of hidden, hitherto unknown and potentially useful information (relationships, patterns) from data using automated or automated means. This method uses statistical methods and procedures in conjunction with the methods of machine learning and pattern recognition. Data mining is sometimes seen as an analytical part of knowledge discovery in databases (KDD).

How, however, do the inexhaustible stacks of available data make useful information and, consequently, the knowledge that motivates to successful action? This transformation is a general goal of business intelligence, and data mining provides appropriate methods.

The path leading to revealing the facts concealed in the data may not be at all unintelligible. Classical analysis by creating tables, graphical depictions, statistical analyzes, or advanced multidimensional (OLAP) analyzes may not reveal all the important aspects concealed in the data. Additionally, while these types of analyzes provide, in particular, an answer to the questions of "What Happened?" Or "Why It Happened?", Data Mining aims in particular to provide information on questions "What happens?" What probability? ". Data Mining thus provides predictive decision support models, most often in relation to customer relationship / product offer optimization for clients, taking into account the expected benefits and risks.

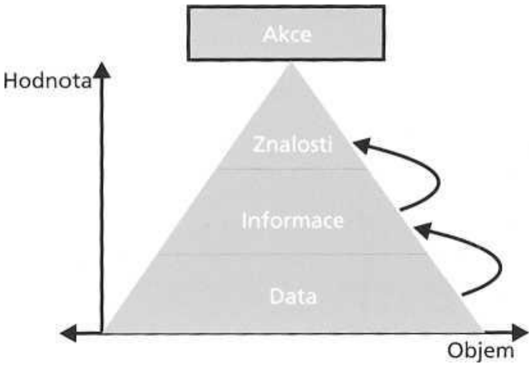


Fig. 13.1 Data Mining – Transaformation of data to knowledge and aciton

Data mining can also be understood as a means of transforming large volumes of historical data into small volumes of essential knowledge, and therefore it has a positive impact, namely saving technology assets in the company.

## 13.1 Data mining methodology

The process of development and use of data mining methods can by no means be considered as a process with a predetermined course, content and assumptions. The process takes place in iterative cycles. In the development phase, it is necessary to correct, e.g., the default set of data, or the used methods. In the production phase, it is necessary to evaluate continually the results, to verify whether they still support the set business objectives, because the reality is constantly evolving, and the well-tuned model may become obsolete over time.

Therefore, it is not possible (even for a generally valid objective) to provide a clear choice of the default set of data, a suitable model, or a specific procedure. Nevertheless, during the 1990s, there were methodologies describing the process at least in general steps, for example. CRISP-DM (Cross-industry Standard Process for Data Mining), or SEMMA methodology (Sample, Explore, Modify, Model, Assess). The common essence of these methodologies is the sequence of several steps, as shown in Figure 13.2

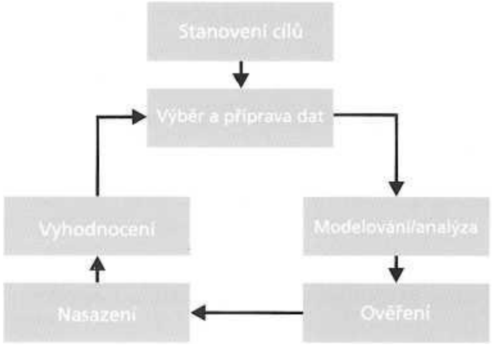


Fig 13.2 General steps of the data mining methodology

These steps are:

1. The defining objectives - formulating the task and understanding the issue from the point of view of business (business understanding).
2. The selection and preparation of data - basic data analysis, control of their quality and format, definition of derived variables.
3. The modeling / analysis - choice of methods, modeling.
4. The verification - the verifiction that the model is correct on another set of data.
5. The implementation - putting into practice using a single use of model outputs, or integrating a model into a re-run business process.
6. Th evaluation - continuous verification of validity / topicality of outputs of the model and the benefits of its use.

An inappropriate model can provide a reliable but trivial outcome (for example, the most likely a fined driver holding a B license is a person over 18 years of age). Therefore, the results obtained can not be interpreted without understanding their meaningful meaning.

There are a number of tools ("packages") on the market that support, in the interactive user mode, in particular the data preparation, modeling and analyzing, verification and evaluation phases, and some instruments are "industrialized". it is possible to use them in non-interactive mode. However, integrating the entire data mining process into regular business operations is more complex than just buying and installing tools and developing models. This activity must take into account aspects such as:

- The automatization of the collection and preparation of data for repeated data mining and ensuring its quality.
- The running of the model calculations as needed or according a schedule.
- The transfer of a model outputs to operating systems, changes in these systems associated with the use of outputs in them.
- The reorganizing internal processes leading to efficient use of model outputs.

An example of such a complex activity can be the integration of data mining methods into the process of preparation and management of marketing campaigns and subsequent management of contacts and relationships with the client.

## 13.2 Data minig methods

Data Mining is based on statistical methods and procedures in conjunction with machine learning and pattern recognition methods. When applying statistical procedures, there is no unusual violation of some weaker default settings (such as the assumption of probability distribution of data) in favor of finding the solution you want. The important is the result, not the 100% mathematical correctness of the path leading to it.

Data Mining usually works with client data. The basic types of these data are:

• Socio-demographic data - age, place of residence, education, nationality, gender, number of children, etc.

• Behavioral data - describes customer behavior and habits, such as product structure or shopping cart, quantity and volume of transactions performed, distribution of channels used, payment history and morals (repayment of credit or invoice payment), contact transactions.

Each model has different capabilities in different tasks, one model can use tens to hundreds of input variables. In addition to tasks using client data, there are a number of tasks in which technical data are processed.

Depending on how the model represents its internal content, we distinguish the following types of models:

**Analytic model** is generally represented by a set of parametric functions defined analytically, with the parameters being set in the learning process. The advantage is the easy interpretation of the individual parameters and the model, and the simplicity of the analysis (what-if analysis) and the proof of the model (eg the regression model).

**Distributed model** - parameters of a model are closely related to the model topology and may not be easily interpretable separately (especially for neural networks). The advantage is the robustness and ability to achieve good results even in the basic setting, the disadvantage is higher computational difficulty (e.g., decision trees, association rules, production rules, neural networks).

**Probability model** - uses constructions from probability theory, a priori and conditional probabilities (e.g., Bayes network).

The main categories of methods used in the development of the model are:

- Regression - linear (e.g., logistic), nonlinear, etc.
- Classification - classification trees, nearest neighbor method, neural networks, etc.
- Clustering - cluster analysis, K-means, method K averages and its modification, self-organizing maps (SOM), etc.

The use of multiple methods and models, comparing the results achieved, and choosing the approach that best fits the set goal is an appropriate way to solve a specific task.

Each method is associated with a particular model type and only supports specific types or types of learning. Learning is an iterative process that adapts the parameters (including topology) of the model. We generally recognize these types of learning:

- Supervised learning - learning uses feedback on the output of a model delivered externally by a "teacher." The appropriately initialized model based on inputs determines outputs the accuracy of which is determined by the "teacher". This type of learning requires the allocation of input data ideally to three sets: training, validation and testing. On a training set, the model adapts its parameters to learn to detect hidden relationships in the data. The validation set is used to assess whether and to what extent the model has already been taught. The learned model is then used and put into practice after processing of the test set.
- Unsupervised learning - feedback from outside is not available. Data transformation and the self-adapting model are based on input data provided. A typical example is cluster analysis.
- Reinforcement learning has no significant use in BI. The actual process of learning is done by evaluating the action and its use in processing additional data

Problem of over-learning (over-fitting):

It is a state where the model did not capture the real dependence between the data but essentially "memorized" the relationships of the training data set. The result is that the model can generalize the learned relationships to another data set.

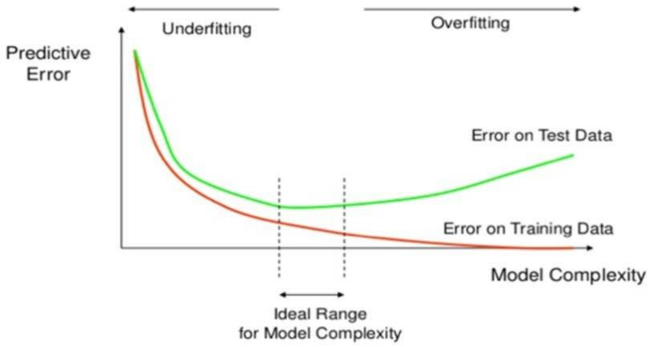


Fig. 13.3 The general relationship between the learning rate and the error rate of models on different data sets

The problem of over-fitting can be prevented by introducing a validation set of data. In the case of a small number of input data, where the training and test set is too small, there is the possibility of using cross-validation: the input data is divided into N equally numerous sets, with the training on N-1 set, testing on the remaining sets. This rotation is performed over all the combinations of the training / test set distribution, the resulting N outputs of the model are appropriately combined.

A specific mining case is so-called text mining, which allows the detection of useful knowledge concealed in text-specific or document-related data, by categorizing texts, etc. However, text mining methods require for each language the definition of relevant dictionaries and grammatical rules (e.g., declension, timing).

## 13.3 Examples of data mining practice

The general purpose data mining tasks include the areas of marketing, sales and customer relationship management (CRM):

- Churn / attrition management prediction.
- Estimation of the customer's propensity to buy.
- Market-basket analysis.
- Cross-selling, up-seiling.
- Fraud detection / prediction.

Specific areas of use are, for example;

- Anti money laundering (AML).
- Risk estimation (eg credit) associated with the client.
- Web mining (analysis of web page passages).
- Detecting terrorist activities.

Data Mining is also applicable in more "technological" areas:

- Prediction of system failures.
- Analysis of causes of inferiority.
- etc.

# 14 FURTHER BI DEVELOPMET

BI analytical tools are not the privilege of analysts and top executives at present, and are expanding among all the roles in the organization:

- Analytical tools are incorporated into business processes.
- Static assemblies no longer meet information needs, and are replaced by analytical applications.

A few years ago, management was able to know the consolidated state of the company at the end of a quarter or a month. Today is working hours.

When the first data warehouses began to appear, the available information about the state of the company was reduced to weeks or days. For a long time, a one-day delay was a magical limit that few were able to cross. This is related to the fact that most organizations work in one-day business cycles, where primary transaction systems are loaded during the day, and overnight data is processed in analytical systems. You just have to find a way to get source data more often than once a day and the imaginary barrier will be broken. Then it is only a question of the cost of repeating this cycle after hours or tens of minutes.

Of course, not every business needs such quick access to information about its activities, but more and more business entities draw their competitive advantage just from the ability to react lightly to immediate development. A set of tools to monitor and evaluate ongoing transactions is called "Business Activity Monitoring". Typical examples include tracking the cail center load, tracking the number of requests in individual steps of the loan approval process, or other business process.

The question remains about what is likely to happen in the near future. Within this framework, we have to solve several problems: Will the current state of inventory be sufficient for the next receipt of goods? Do I have enough qualified people to process the upcoming campaign? What is the likely number of responses? However, it is easy to answer everything using Data Mining tools or specialized forecasting applications.

Unlike the classic reporting tools that make historical data available in the rear view mirror, Business Activity Monitoring (BAM) and predictive technology allow you to track the current state and look to the near future. Therefore, you can see how predictive technology slowly but surely begins to deploy as part of a BI / DWH solution.

Practice shows that the key to success is not the formulation of a successful development strategy, but the ability to implement a strategy at all.

Until recently, BI tools were designed exclusively for managers, as if they only needed to make decisions based on facts. If the company wants to achieve its goals, it needs to work as one. To do this, all employees have to work with the uniform information model of the company. Such requirements have an impact on technology that, instead of handling a few analysts' queries and generating tens or hundreds of reports, must be able to serve thousands of workers.

How can analytical tools be available to the common users? One way is to apply one of the integration tools and analytical tools to business processes. Each approach has resons for and against. The ideal state is to offer so-called "Embedded Analytics" that will provide the greatest possible penetration between users and the potential of analytical tools.

# 15 LITERATURE

ECKERSON, Wayne W. Performance dashboards: measuring, monitoring, and managing your business. 2nd ed. Hoboken, N.J.: Wiley, 2011.

POUR, Jan, Miloš MARYŠKA, Iva STANOVSKÁ a Zuzana ŠEDIVÁ. Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. Praha: Grada Publishing, 2018. Management v informační společnosti. ISBN 978-80-271-0616-5.

TYRYCHTR, Jan a Alexandr VASILENKO. *Business intelligence in agribusiness: fundamental concepts and research*. Brno: Konvoj, 2015. Monografie (Konvoj). ISBN 978-80-7302-170-2

NĚMEC, Radek. *Principy projektování a implementace systémů Business Intelligence*. Ostrava: VŠB-TU Ostrava, 2014. ISBN 978-80-248-3452-8.

TYRYCHTR, Jan. *Business intelligence*. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2014. ISBN 978-80-213-2516-6.

POUR, Jan, Miloš MARYŠKA a Ota NOVOTNÝ. *Business intelligence v podnikové praxi*. Praha: Professional Publishing, 2012. ISBN 978-80-7431-065-2.

LABERGE, Robert. *Datové sklady: agilní metody a business intelligence*. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.

ŽIŽKA, Jan. *Business intelligence*. Praha: Vysoká škola ekonomie a managementu, 2011. ISBN 978-80-86730-79-0.

LACKO, Ľuboslav. *Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby*. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.

ARNOŠT, Daniel. *Business intelligence: příručka manažera*. Praha: TATE International, 2007. Příručka manažera. ISBN 978-80-86813-12-7.

NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. *Business intelligence: jak využít bohatství ve vašich datech*. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1094-3.

LIEBOWITZ, Jay, ed. *Strategic intelligence: business intelligence, competitive intelligence, and knowledge management*. Boca Raton, FL: Auerbach Publications, 2006. ISBN 0-8493-9868-1.

HOLÝ, Pavel, KUČEROVÁ, Jana, ed. *Application of data mining methods in strategic planning*. Praha: České vysoké učení technické v Praze, c2012. ISBN 978-80-01-04953-2.

PETR, Pavel. *Data Mining*. Vyd. 3. Pardubice: Univerzita Pardubice, 2010-. ISBN 978-80-7395-325-6.

DREIBELBIS, Allen. *Enterprise master data management: an SOA approach to managing core information*. Upper Saddle River, NJ: IBM Press/Pearson, c2008. ISBN 0-13-236625-8.

ARMSTRONG, Michael. *Armstrong's handbook of performance management: an evidence-based guide to delivering high performance*. Sixth edition. London: KoganPage, 2018. ISBN 978-0-7494-8120-9.

BLAHOVÁ, Michaela. *Strategic framework and model for managing business performance: utilisation of synergies of selected management systems in the global environment*. Praha: Wolters Kluwer, 2017. ISBN 978-80-7552-922-0.

VILLIERS, Charlotte. *Corporate reporting and company law*. New York: Cambridge University Press, 2006. Cambridge studies in corporate law.

CHLAPEK, Dušan, Václav ŘEPA a Iva STANOVSKÁ. *Analýza a návrh informačních systémů*. Praha: Oeconomica, 2011. ISBN 978-80-245-1782-7.

HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK. *Big Data a NoSQL databáze*. Praha: Grada, 2015. Profesionál. ISBN 978-80-247-5466-6.

KROENKE, David a David J AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

POKORNÝ, Jaroslav a Michal VALENTA. *Databázové systémy*. Praha: České vysoké učení technické v Praze, 2013. ISBN 978-80-01-05212-9

BÍLA, Jiří. *Informační technologie: databázové a znalostní systémy*. Vyd. 3., přeprac. V Praze: České vysoké učení technické, 2009. ISBN 978-80-01-04409-4.

ŠARMANOVÁ, Jana. *Informační systémy a datové sklady*. Ostrava: Vysoká škola báňská - Technická univerzita, [2008]. ISBN 978-80-248-1500-8.

LACKO, Ľuboslav. *Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle*. Brno: Computer Press, 2003. ISBN 80-7226-969-0.

LABERGE, Robert. *Datové sklady: agilní metody a business intelligence*. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.

INMON, William H. *Building the data warehouse*. 4th ed. Indianapolis: Wiley Publishing, c2005. ISBN 0-7645-9944-5.