



Lecture – DWH & DM



Lecture

Data Warehouse & Data Mining (DWH & DM)

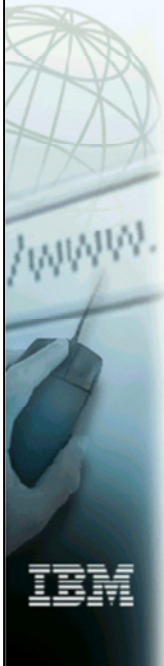
Stuttgart, October 2015

Dr. Hermann Völlinger

Executive IT Architect

IBM Software Group (SWG) Services

Data Warehousing (DWH) & Business Intelligence (BI)



Folie: 1

Dr. H. Völlinger, IBM

General Remarks to Lecture DWH & DM

- Time: each Friday, 14:00 – 16:45 o'clock
- (i.e.: 120 minutes lecture, 30 minutes exercises, 15 minutes break)
- In exercises everyone should present at least one time his exercise solution
- Starting: Fri., 02.10.2015, ending: Fri., 18.12.2015
- Collection of solutions (“Musterlösungen”) of exercises exist in the Homepage.
- 10 Lessons + 1 Examination (in week 14.12.- 18.12.2015)
- BA Lecture Homepage: <http://www.ba-stuttgart.de/~hvoellin/>
 - Solutions to Exercises
 - Sample data for exercises
 - Other information
- Evaluation of examination will be seen on BA Diploma Certificate



Content: Data Warehouse & Data Mining

Goal: Introduction, Architecture and Basic Concepts

1. Introduction to Data Warehousing (DWH) & Business Intelligence (BI) (Friday, 02.10.15)
2. DWH Architecture (Virtual, 1-Tier, 2-Tier), Advantages & Disadvantages (Fri., 09.10.15)
3. Overview about DBMS (i.e. Relational Databases) (Fri., 16.10.15)
4. Introduction to Basics of SQL & Examples (Fri., 23.10.15)
5. Multi-dimensional Data Modeling (MDDM), OLAP examples in DWH (Fri. 6.11.15)
6. ETL – Reference Architecture (Introduction) (Fri., (13.11.15)
7. ETL – Data Population Techniques, Tool Examples (Fri., 20.11.15)
8. Analysis (OLAP) & Reporting Concepts, Examples (Fri., 27.11.15)
9. Data Mining Business concepts, Examples (Fri., 04.12.15)
10. Data Mining Techniques, Tool example -IBM Intelligent Miner (Fri., 11.04.15)
11. Examination (14.12.-18.12.2015)

Literature List – Part 1

1. General DWH – Barry Devlin 'Data Warehouse....', Addison-Wesley, ISBN: 0-201-96425-2
2. General DWH – R. Kimball 'The Data Warehouse Toolkit.', John Wiley & Sons, NY 1996, ISBN: 0-471-15337-0
3. General DWH – Andreas Bauer, Holger Günzel (Hrsg.): 'Data Warehouse Systeme - Architektur, Entwicklung, Anwendung' DPunkt Verlag Heidelberg 2004, 3. Auflage, ISBN: 978-3-89864-540-9
4. General DWH – R. Kimball,..... 'The Data Warehouse Lifecycle Toolkit.', John Wiley & Sons, NY 1998, ISBN: 0-471-25547-5
5. DWH & Business Intelligence (BI) – Stefan Eckrich, 'From Multiplatform Operational Data to Data Warehousing and Business Intelligence', IBM Redbook, SG24-5174-00, ISBN: 0-7384-0032-7
6. Business Intelligence (BI) on S/390 – V. Anavi-Chaput, 'Business Intelligence Architecture on S/390 –Presentation Guide', IBM Redbook, SG24-5641-00, ISBN: 0-7384-1752-1
7. Meta Data – David Marco 'Building & Managing the Meta Data Repository', , John Wiley & Sons 2000, ISBN: 0-471-35523-2
8. DB2 OLAP Server – Corinne Baragoin , 'DB2 OLAP Server Theory and Practices', IBM Redbook, SG624-6138-00, ISBN: 0-7384-1968-0

IBM Redbooks can be found in the Internet under : <http://www.redbooks.ibm.com>



e-business



Lecture – DWH & DM



Literature List – Part 2

9. Databases (i.e. IBM DB2 UDB) – *Don Chamberlin 'A Complete Guide to DB2 Universal Database'*, Morgan Kaufmann Publ. Inc., ISBN: 1-55860-482-0
10. VLDB – *J. Cook, 'Managing VLDB Using DB2 UDB EEE'*, IBM Redbook, SG24-5105-00
9. **Data Modeling (Historical Models)** – *C. Ballard, D. Herreman, 'Data Modeling Techniques for Data Warehousing'*, IBM Redbook, SG24-2238-00
10. ETL – *Thomas Groh, ... 'BI Services -Technology Enablement Data Warehouse -Perform Guide'* IBM Redbook, ZZ91-0487-00
11. **ETL & OLAP** - *Thomas Groh, ... 'Managing Multidimensional Data Marts with Visual Warehouse and DB2 OLAP Server'* IBM Redbook, SG24-5270-00, ISBN: 0-7384-1241-4
12. **Data Mining** – *Peter Cabena 'Discovering Data Mining,'*, Prentice Hall PTR, ISBN: 0-13-743980-6
13. Data Mining – *P. Cabena 'Intelligent Miner for Data – Applications Guide'*, IBM Redbook, SG24-5252-00, ISBN: 0-7384-1276-7
14. Data Mining – *C. Baragoin 'Mining your Own Business in Telecoms'*, IBM Redbook, SG24-6273-00, ISBN: 0-7384-2296-7

Folie: 5

Dr. H. Völlinger, IBM

List of Topics for BA Semester Work (5 th Semester) or BA Diploma Thesis (6 th Semester)

See current version on Homepage: <http://www.ba-stuttgart.de/~hvoellin/>

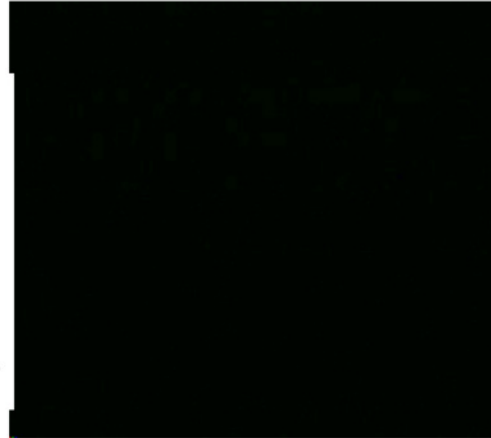
No	Topic	Remarks
1	New Requirements for Data Population and Data Quality for a Data Warehouse by the new Solvency II law (Insurance industry)	Compare existing documentation about Solvency II in the internet and also the official description published by the Bank of International Settlements. Generic data models and papers about these new requirements exist. Hint: http://www.solvency-ii-requirements.de
2	New Requirements for Modeling (Business Model and Logical Data Model) by the new Solvency II law (Insurance industry)	Compare existing documentation about Solvency II in the internet and also the official description published by the Bank of International Settlements. Generic data models and papers about these models also exist. Compare for example what predefined solution-assets exist in the industry (i.e. from Microsoft or from IBM with DW-Insurance Information Warehouse. Hint: http://www.microsoft.com/downloads/details.aspx?familyid=14877a3-1111-1019-b3&docid=5a1c8a6c-4e7d-406b-9000-000000000000
3	Examination of Agile Project Development Method: for building a Data Warehouse (Pro's and Con's)	Data warehouses take too long to build; that, once built, they often no longer match their user's needs; that, once in production, they are too hard to change. Agile BI methods build for data warehousing integrated development environment (IDE), transform traditional approaches to building, deploying and managing data warehouses, applying agile methodology and integrated data flows to streamline data warehouse development, deployment, and reversion. Such an approach supports the entire data warehouse management life cycle, integrating source system exploration, schema design, meta data management, warehouse scheduling and enhancement into a single, simple integrated design. Hint: http://www.schibstedigital.com/Whitepapers/Document14877a3-1111-1019-b3&docid=5a1c8a6c-4e7d-406b-9000-000000000000
4	Examine the Influence of Data Quality (DQ) in the Development and Operation of a Data Warehouse (DWH) and how can DQ Tools help	Examine the Influence of Data Quality (DQ) in the Development and Operation of a Data Warehouse (DWH) and how can DQ Tools help
5	The role of metadata in the Development and Operation of a Data Warehouse (DWH)	see the book of David Marco (literature list)
6	Technical Evaluation of ETL Tools and their usage in seven typical DWH BI project/customer scenarios	Create and use a criteria catalogue for the evaluation. Use the information from the internet about the tools
7	Technical Evaluation of OLAP & Reporting Tools and their usage in seven typical BI/CRM project/customer scenarios	Create and use a criteria catalogue for the evaluation. Use the information from the internet about the tools
8	Analysis of the basic concepts of an OLAP Tool (i.e. IBM OLAP Server) versus a Data Mining Tool (i.e. IBM Intelligent Miner). Show weaknesses and strengths of each concept	Use the information from the lesson DWH-DM and information from the standard books about data mining and OLAP or information from the internet
9	Historization concepts in a DWH Comparison and Description of these Concepts	Describe the most popular concepts and show their usage in a data warehouse environment. Show advantages and disadvantages of each of these concepts. Develop use cases for concrete implementation scenarios
10	Data Load in a DWH – Techniques and their Usage	Describe the different techniques of data detection and data load in the data population process in a DWH environment. Show advantages and disadvantages of each of these concepts. Develop use cases for concrete implementation scenarios

Start Time: Winter Semester for Semester Work & Summer Semester for Diploma
Maximum Number per Year: 2 for Semester Works, 1 for Diploma Thesis
Evaluation: Dr. Hermann Völlinger



Lesson 1

Introduction to DWH & BI



Motivation - What is Business Intelligence (BI) – the Problem

Water, Water Everywhere ...

...But not a drop to drink



Data, Data Everywhere...

...But none to help me think

Competition



The foil shows the main problem in today's business:

We have a big mass of data existing in transaction system, but we can not use it, since the data is not available or accessible.

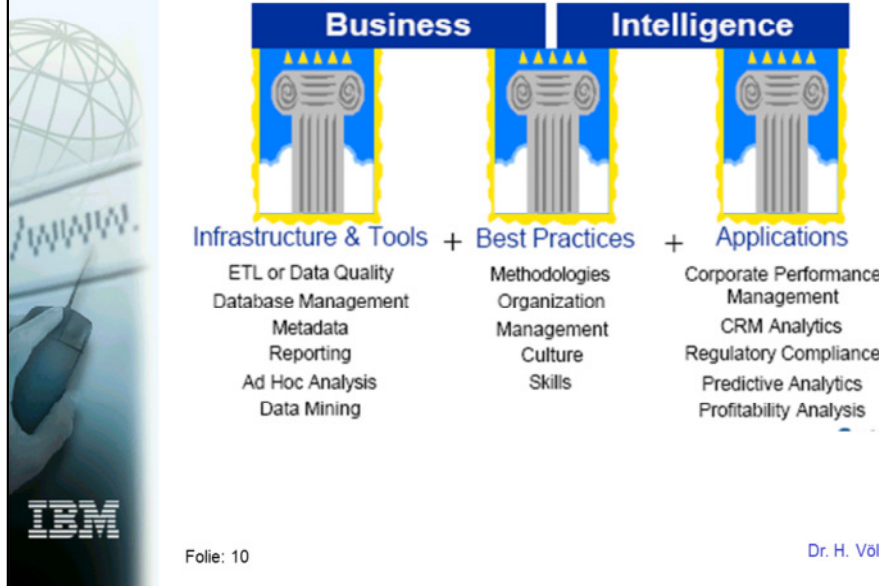
First Definition: What is BI & BI Mission



BI has to mission to get more knowledge about my business.

But before I can get to this knowledge I have to prepare the infrastructure for analysis and BI , i.e. I have to build a DWH (this is the rough idea of Data Warehousing).

What is BI – the three Pillars of BI



Folie: 10

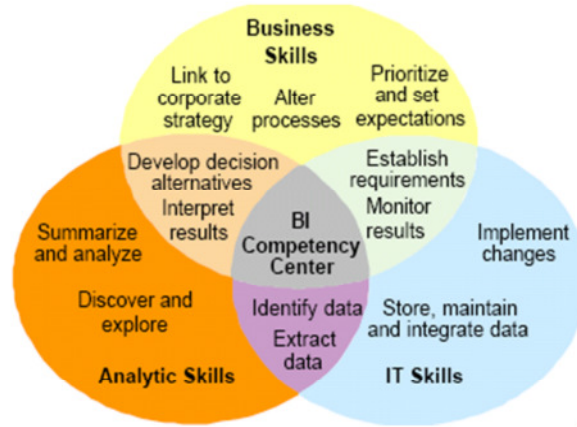
Dr. H. Völlinger, IBM



BI is based on these three pillars.

Pillar one is necessary for pillar three.


Pillar two is the method to build a successful DWH/BI solution.

What is BI – the BI Competency Center



 **Lecture – DWH & DM** 

BI - Getting the Answers you Need




What products sold well this month in Germany?

What types of marketing campaigns should we deploy?

This history of sales in USA by city, by product shows we still have tremendous potential

Look at these sales numbers compared to the last 3 years

Let's have a campaign on these 3 products where profit is over 15%

 Folie: 12 Dr. H. Völlinger, IBM

The foil shows the questions and goals of a Business Analysis in a company. The answer to these questions is called 'Analytical CRM'.

We will learn more about this in the following chapters

- This foil shows people asking the kind of questions that good data analysis and query tools do an excellent job of answering. Questions such as "Which products are selling best?", and "What campaigns should we execute?" can provide real business advantages for companies if they are answered correctly, efficiently and promptly. The problem is to provide the answers needed for decision making requires information. To get information, you must start with data -- data which is available from many sources internal to the organization and from external sources.
- Organizations today do not suffer from a lack of data, but from an **abundance of redundant and inconsistent** data that is difficult to manage effectively, is increasingly difficult to access and difficult to use for decision making purposes. No matter how good the tool is, if the database being accessed does not have the right data in the right form, the answers will be unsatisfactory.
- What is required is an architected solution that makes the best use of all the available data and transforms it into a more appropriate format for decision making. What is required is Data Warehousing.
- Data Warehousing makes the best use of all available data and turns it into data structures, which could be accessed and analyzed by analytical processes and tools like OLAP and Data Mining. So we get out of data the valuable information.

Lecture – DWH & DM

Different Data for Different Users

Operational Systems

- Order Entry
- Payroll
- Accounts Receivable
- Personnel

Informational Systems

- Product Sales Analysis
- Trend Analysis
- Ad-Hoc Queries
- Data Mining

Folie: 13 Dr. H. Völlinger, IBM

Operational and Informational Applications


Much of the data inside organizations is captured and used by the day-to-day **operational applications** that help run the business, e.g. order entry, payroll, etc. These applications are optimized for operational purposes and the data typically is real-time and constantly changing. It is typically not in a format or structure that is easily understood or easily accessed by end users for informational purposes. These systems are also often called **Legacy Systems**.

Informational applications require a more stable source of data that is in a format more suitable for analysis, e.g. query and reporting, trend and point-in-time analysis, data mining. The data is usually "transformed", to remove data of use only in the operational environment, change the data format, and eliminate anomalies to improve data quality. Transformation also makes the data more easy to understand and use by business users. The data is stable, consistent as of some point-in-time and will not be changed until the end user desires. It may reflect a history of values, i.e. changing values over time.


Different Data for Different Uses

Most organizations need two data environments, one for operational and one for informational applications. The original data for both types of applications may be the same but the data used by these two applications is fundamentally different. Using the same data environment to support both these applications can compromise the performance, capability, and benefit they provide.

A data warehouse provides a separate physical data store to better support informational applications ('**dispositive**' data). The data to support these informational applications, however, does need to be organized in specific ways.



Lecture – DWH & DM




Turning Data to Information

- **The need for a warehouse model**

To identify the data sources available & to define target informational data
- **The need to transform the data**

To identify the transformations required to build the data structure and data granularity
- **The need for an information catalogue**

Capturing the metadata - which helps you to understand the structure and the meaning of the data



Folie: 14 Dr. H. Völlinger, IBM

Need for a data warehouse model

➤ To enable the movement of data into the informational databases (data warehouses), the "master" data sources must be identified. This sometimes requires an "inventory" of the current operational data to make sure the appropriate data elements are accounted for. This information is used to create the definitions for the target informational databases. The formal term for this activity is "creating a data warehouse data model". This model will define the target informational data. We will see these in ore detail in a following chapter of the lecture.

Need to transform the data

➤ Next, the transformations and enhancements that are needed must be defined. For example, rather than sending all the detailed data only summary information will be sent; or only a subset of the detailed data sent based on specific criteria. Data could be enhanced by such actions as adding a time value. Based on these transformation requirements, the data in the warehouse could consist of **reconciled, derived, and/or changed** data.

Need for an information catalog

➤ Another part of the process involves capturing the metadata that defines the informational data in the data warehouse. This metadata is used to populate an information catalog that enables users to understand what data is in the data warehouse and exactly what it means. Metadata is key in data warehousing as it provides users a "window" to the world of data in the data warehouse.

Before looking more closely at the data-to-information process, lets first look at the data types involved.

Structuring the Data – Five Data Types

1. **Real-Time Data** - mainly used by operational systems
2. **Reconciled Data** - cleaned, adjusted or enhanced
3. **Derived Data** - summarized, averaged or aggregated
4. **Changed Data** - data history, build time stamps
5. **Metadata** - data about data, descriptive information about the data (structure and meaning)

There are 5 main data types involved in data warehousing:

Real-Time Data: Typically used only by operational applications. It contains all the individual, detailed data values, each update overlaying the previous value so there exists no history of changes. It may exist redundantly in multiple locations, which may or may not be synchronized together, e.g. data from several bank branches. It may also be inconsistent in representation, meaning or both. Real-time data normally requires some transformation before being used by informational applications.

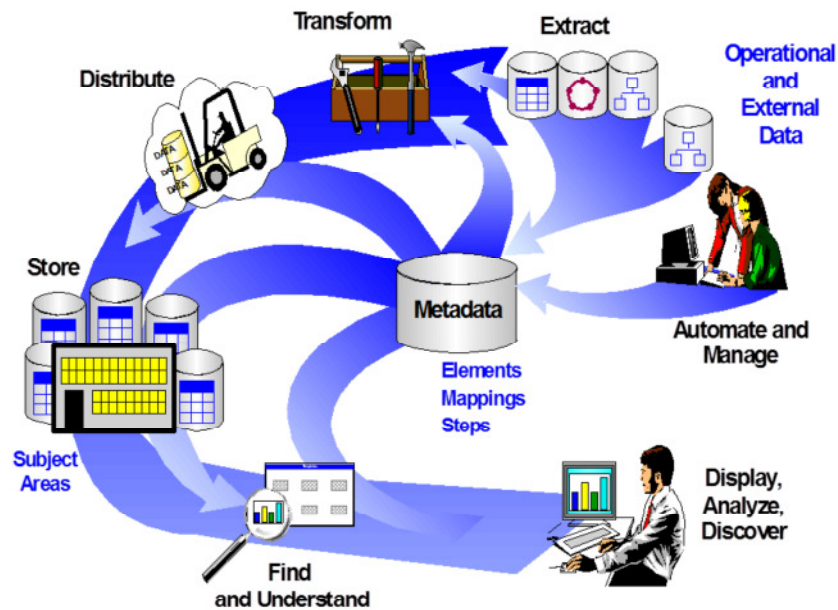
Reconciled Data: Real-time data that has been cleaned, adjusted, or enhanced to provide a source of quality data for use by informational applications.

Derived Data: Summarized, averaged or aggregated data from multiple sources of real-time or reconciled data, providing detailed data in a suitable format for decision making, reducing resource requirements and increasing application response time.

Changed Data: Data that can kept as a continuum and reflects the data history. It is a record of all the changes to selected real-time data, time-stamped to document the level of currency. Since all changes are included, trend or point-in-time analysis can be achieved. This does present issues such as how to archive the data to make retrieval easy when desired, and could mean storage of a large volume of data.

Metadata: Provides descriptive information on what data is available for accessing, exactly what it represents, how current it is, etc. for the development of informational applications. Building a data warehouse requires capturing both data and metadata. Typically metadata is used for database design and application development, but with data warehousing, metadata definitions are also transformed into business terms for end users and an information catalog is provided to make it easy for end users to search for and use the metadata.

Turning Data into Information (Part1)



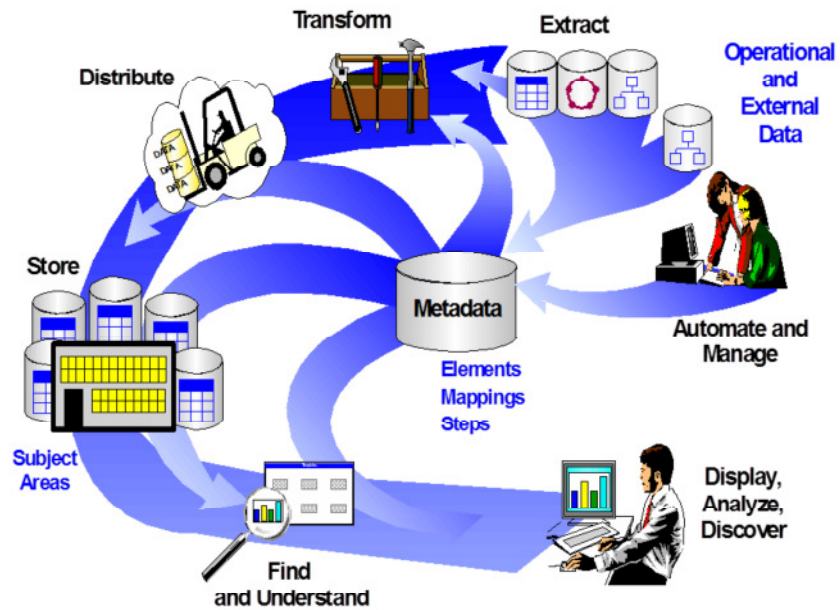
Folie: 16

Dr. H. Völlinger, IBM

➤ **In general, the data-to-information process takes the following steps:**

- Data is **extracted** from internal operational data and from data available in sources external to the enterprise. Administrative support is required to define the input data, transformation requirements, method of storing the data in the warehouse, and its presentation to the end user. Since this is an ongoing process, management and automation of the process tasks is important to minimize the burden of operations and have tools that make administration easy.
- Based on the types of information applications required, decisions will be made on how the data needs to be **transformed**, e.g. what operational data is not required and can be deleted, how data should be changed to make it understandable by end users. Some data may need to be enhanced (summaries rather than all the detailed data), and some might be aggregated from multiple different data sources into a single data element. Once transformed, data is placed in the warehouse data stores based on a selected business subject area structure.
- Data in a data warehouse is typically organized by "**subject areas**". Examples are "customer", "product", "employee", or functional business areas (Finance). With the data organized in the data warehouse in this way, it can be more consistent, at known levels of currency, minimally redundant, of better quality, and more easily available to end users for access and analysis.
- **Business views** of the data are developed as part of the structure and made available to users for easy access to the data for their business purposes. By mapping the data to a Business View and presenting it to the end user in the context of a chart, graph, or report, the data becomes information - information that can be used to make better informed business decisions.

Turning Data into Information (Part2)



Folie: 17

Dr. H. Völlinger, IBM

Definitions of the data elements, transformation processes, and business views are stored as **metadata**:

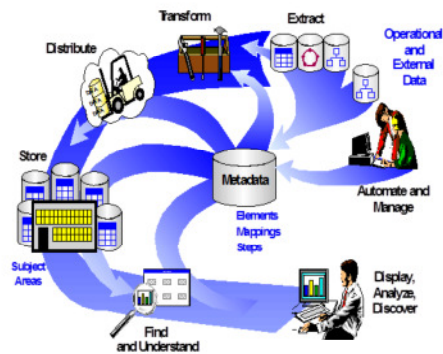
- **Technical Metadata** (data elements, transformation mappings) is used to build and maintain the data warehouse processes.
- **Business Metadata** is used to help end users understand what data is in the warehouse, in business terms they can understand.
- A **metadata store**, or information catalog, is built and maintained by the data warehouse administrators. Metadata management is key both for the processes involved in data warehousing and for enabling end users' access to the data.

What is a Data Warehouse ?

‘A subject-oriented,
integrated, time-variant,
non-volatile collection
of data in support of
management decisions

... ‘ **W. H. Inmon**

**Goal: Turning Data into
Information !**



The foils shows also the Data Warehouse definition of W. H. Inmon, who is one of the most famous architects of Data Warehouse community.

Data in a data warehouse are build, created and transformed out of operational data such that they full-fill the following four attributes:

1. Subject-oriented
2. Integrated
3. Time-variant
4. Non-volatile

Data Marts or Data Warehouses

•Which Is Right For You?

•Identify business problems that the data mart or data warehouse will address

•Scope of data mart or data warehouse

- Size
- Budget
- Timescale
- Resource

•Type of users that data mart or data warehouse will serve

•Amount of growth of data mart or data warehouse over time



Folie: 19

Dr. H. Völlinger, IBM

Providing a choice of implementation leaves some companies wondering whether it would be best to start with a data mart and build up to a data warehouse, or go the opposite route and build a data warehouse that can be broken down into data marts.

➤ **Data marts work like data warehouses** - moving data from source to target databases. But data marts are different than data warehouses, because they serve specific business purposes/solve particular business problems. Although they still have to collect, transform and clean data, there's less of it, so therefore you'll spend less money, less time, and fewer resources to put data marts in place and maintain them.

➤ The first step is to **identify the business problems** that need solving by the data warehouse, and find the best solution to these business needs. The other main considerations are those shown in this foil. Of course, if your decision has been to start with a data mart, you can still evolve to a data warehouse. Note, however, that building a warehouse from several data mart requires strategic planning. One way may be to start with a small number of data marts, and spend up-front time building the architecture for the data warehouse as you build the data marts.

➤ You can get up and running quickly now with data marts, and can evolve over time to include the tools you need to grow your data mart to a full-scale warehousing solution for your enterprise, as long as you make the right planning decisions at the start.

e-business

Lecture – DWH & DM

DHBW
Duale Hochschule
Baden-Württemberg
Stuttgart

The need for an Information Catalog

- Finding & Understanding the Data

You will learn more about this later

IBM

Folie: 20



Dr. H. Völlinger, IBM

Having set up the data warehouse and provided a data store that is optimized for informational applications, it is important to provide end users with a **way to find and understand the data** in the data warehouse. What is required is an Information Catalog for users, with data descriptions in business terms they can understand.

➤ To access the data, users must first know **what is available** in the data warehouse. This leads to the requirements for an information catalog. To achieve this, the metadata from the source databases is captured and used in the definition of the data warehouse target tables. That source metadata, and metadata that describes any newly defined data elements, provide the base for the data warehouse model. It will have two uses: "technical" metadata used by the administrator for representing the data model, and as input to a "business-oriented" metadata catalog.

➤ **Catalog Information for the Administrators and End Users:** For the administrator, metadata contains the description of the data environment and is the basis for the data warehouse model. It will provide the basis for a "technical" metadata catalog that will support the associations that enable actual access to the data. This technical catalog is used to enable the creation of the business-oriented information catalog.


➤ **The Information Catalog:** For the end user, the information catalog is the key that opens up a world of information. It describes what data exists, where it is located, its currency and so on. It may also be able to catalog and describe information objects, such as queries, charts and reports. All of these descriptions are in business terms that end users can understand. This helps to assure they access and use the correct data and information objects. The data warehouse entry point for the end user is an information catalog: it tells the end user what exists, what it means, and how to use it, all expressed in business terms.

Lecture – DWH & DM

Seven Benefits of Data Warehousing

1. **Data Warehousing Solves Business Problems**
2. **Provides an Integrated Source of High Quality Data for Analysis and Decision Making**
3. **Provides a Consistent View of Data to All Users**
4. **Satisfies the Data Needs of a Business in a Cost Effective Manner**
5. **Minimises Operations Impact**
6. **Data that is Easy to Find, Understand, and Use**
7. **Business Bottom Line**
 - Reduces Costs
 - Increases Profit
 - Increases Competitive Advantage

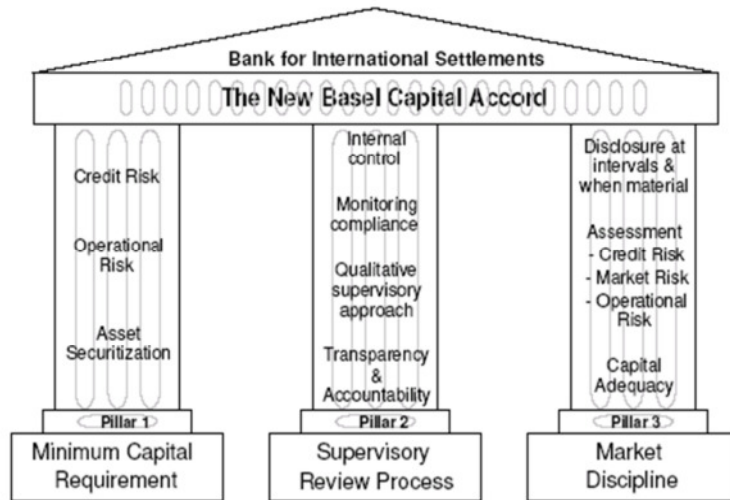


Folie: 21 Dr. H. Völlinger, IBM

Data warehousing provides an excellent approach for transforming data into useful and reliable information to support the analytical and decision making techniques that are so important in today's competitive environment, such as data mining, multidimensional data analysis, query and reporting. Making use of these techniques along with data warehousing can result in more informed decision making capability and lead to significant business advantage.

- Data Warehousing helps **solve business problems** by catering for the needs of business users
- Provides **an integrated source** of high quality data for analysis and decision making - population of the warehouse from a broad range of data sources (both operational and external) provides a single data store, with easy access to source, summarized and historical data, for more informed decision making.
- Provides a **consistent view** of data to all users improving effectiveness
- **Satisfies the business data needs** in a cost effective manner by reducing I/S workload, automating data flow and process tasks to relieve I/S from many of the manual administrative and operations tasks, and significantly reducing the resources required to operate and maintain the data warehouse.
- **Minimizes operations impact** by providing a separate physical data store optimized for informational applications.
- **Data that is easy to find, understand, and use** - enables users to directly access the data to perform their own data analysis and create their own queries and reports by providing an information catalog which makes users aware of what data is in the warehouse, what it means in business terms, and how to use it.

Use Case I – Basel II (Definition)



Folie: 22

Dr. H. Völlinger, IBM

Context

Introduces key features of Basel II, the Pillar Concept and proposed timeline.

Key Points

- Basel II covers credit, market and operational risk
- There are 3 pillars – P1, Minimum capital charge, P2, Supervisory review and P3, Market Discipline
- IBM focus is credit and operational risk as majority of banks will have some solution for their market risk issues (this area is also well covered by risk systems vendors)
- Live by beginning of 2007 but there is a significant amount of preparatory work to be completed well before this date (important to act)
- Option to choose strategy – from basic to advanced
- Basic less advantageous as it requires significant levels of regulator input
- Advanced can deliver significantly higher benefits

Lecture – DWH & DM

Basel II - key challenges – Systems & Data Management

Data Management is the key challenge in meeting Basel II

Challenge	Percentage
Data mgt.	~95%
Project mgt.	~35%
Impl. of sup. Review proces	~25%
Adherence to disclosures	~15%
Sufficient resources	~10%
Mgt buy-in	~5%
Other	~15%

Source: IBM Institute for Business Value analysis, *Banks and Basel II: How Prepared Are They?*, October 2002 interviews with 32 Financial institutions worldwide

10 Common signs of unstable data foundation

1. There's no single enterprise view of data
2. Inability to gather data for as yet unspecified reporting requirements.
3. Senior Management requests for information require intensive manual effort to respond, and far longer than desired.
4. Multiple databases or spreadsheets storing similar data; no common data "dictionary" across the enterprise
5. No ownership of data
6. Difficulty complying with regulatory requirements like Basel II Accord
7. Senior management questions quality, timeliness, reliability of information used to make multi-million dollar decisions
8. Difficulty answering questions about the origins and business processes performed against data
9. Inability to consolidate data from multiple diverse sources
10. Difficulty in building a single architecture to address both data consolidation and data aggregation requirements.

Folie: 23

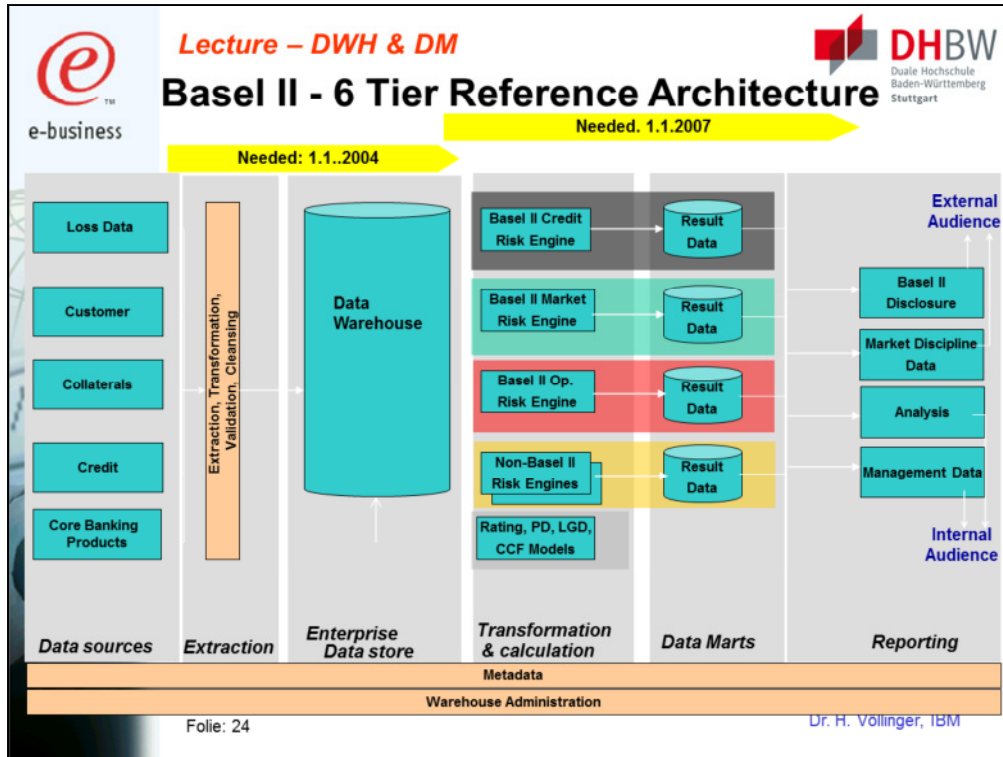
Dr. H. Völlinger, IBM

Data Management is the key challenge - despite these data problems we see today in the companies the need for Basel II to fulfil many **IT Requirements**. These are essential for the success of Basel II:

1. **Standardized data practices**, across all divisions and sources
2. **Extensive time series** data for credit risk drivers
3. **Capacity for massive data volumes**
4. **Sophisticated methodologies** to model the influence of credit risk drivers.
5. **New data modelling processes** (credit rating, LGD, PD, EAD)
6. **Open, flexible architecture** to integrate all kinds of banking models
7. **Internal and external reporting**
8. **Certain analytics**, e.g. calculation of migration matrixes
9. **Integration** with calculation engines

What are Banks doing about it:

- Most Banks have decided on their strategies during 2003
- Banks are now doing data gap analysis based on CP3 to know what data to collect and ensure it is available from source systems
- RFIs and RFPs are being issued for Basel II data storage solutions (Credit Risk Database or Data Warehouse)
- During 2004 most Banks will implementing their Basel II data collection strategies and will start collecting data for Basel 2 compliance. **3 years data needed by 2007**
- 2004 was the Data Management Infrastructure Year



The picture shows the IBM proposal for IT Infrastructure.

It shows a 6-Tier model.

We have a Central DWH. We have Marts as a second, specialised and enriched data store. The reporting layers gives output to internal and also external audience.

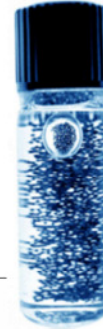
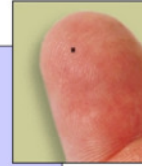
Metadata & Administration layers should work over all tiers.

Build up of CDW should start with 2004, since you need three years of data history. The calculation and the reporting facilities should work by end of 2006, beginning with 2007.

Use Case II – RFID Problem

Tags

Active	Includes a power source to help transmit a signal
Passive	No power to transmit signal; relies on readers
Frequency	Radio wave frequency at which signals are transmitted (Telephone example: 900 Mhz, 2.4 Ghz, 5.8 Ghz)
Data Capacity	Many options, will depend on application
Antenna	Device attached to tag to help capture signals from readers

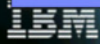
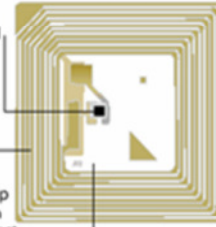


Readers

Reader	Interrogators that typically emit a radio signal via an antenna and collect information that is captured from "scans" using some form of "controller software"
Antenna	Device attached to a reader which helps transmit radio signals and captures "scan" readings

RFID tags are made up of three parts:*

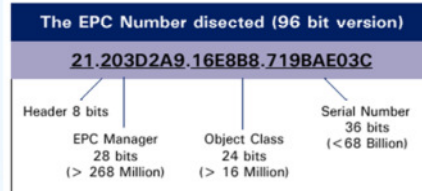
- 1) **Chip:** holds information about the physical object to which the tag is attached.
- 2) **Antenna:** transmits information to a reader (e.g., handheld, warehouse portal, store shelf) using radio waves
- 3) **Packaging:** encases the chip and antenna so that tag can be attached to physical object



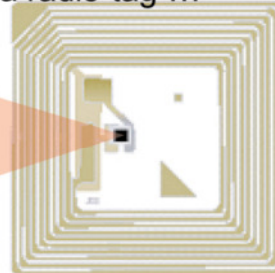
Use Case II - The RFID Numbers

The base of the vision is the Electronic Product Code (EPC) – a robust labeling convention that is embedded into each RFID tag

A number in a radio tag ...







Source: Auto-ID Center



...which together, uniquely identifies an object

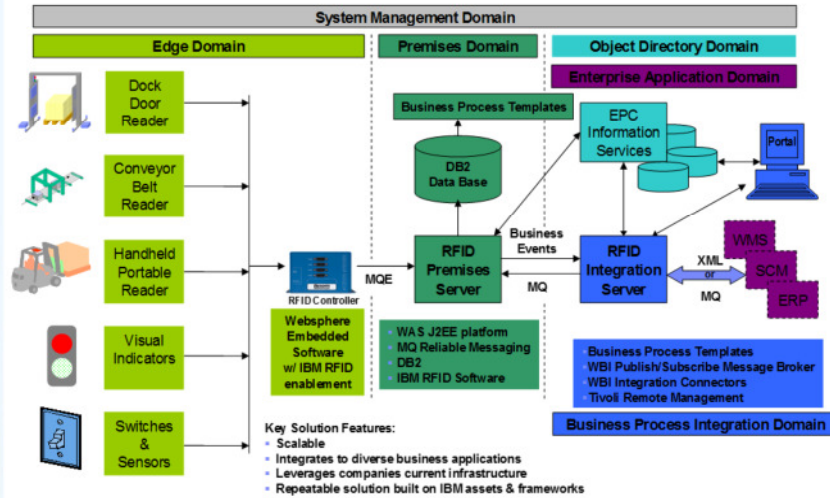
The EPC can catalog over 1.3×10^{16} discrete items annually (about the number of grains of rice consumed globally each year). In contrast, the 12 digit UPC barcode can only identify 100,000 products per manufacturer.

Use Case II – The RFID Infrastructure


			
RFID Self-Checkout	Distribution Center Palette Control (DC Exit)	RFID/AutoID Warehouse	EPC RFID Demo
A supermarket scenario similar to the IBM Commercial „Supermarket“	Verify palette packaging before leaving the distribution center	An order pickup scenario	Represent 3 different points in the supply chain via portals (retail store, retail DC, supplier)



Use Case II – RFID Solution with DWH



Exercise to Lesson 1



Task: Prepare a report and present it at the next exercise session (next week, duration = 10 minutes). Information sources are newspaper or magazine articles or books (see literature list).

Theme: Trends or new development in the following areas (project reports are also possible):

1. Data Warehousing (DWH)
2. Business Intelligence (BI)
3. Customer Relationship Management (CRM)

For Explanation of these Catchword see the next foils

Exercise 1: Market Today – ‘Catchwords’

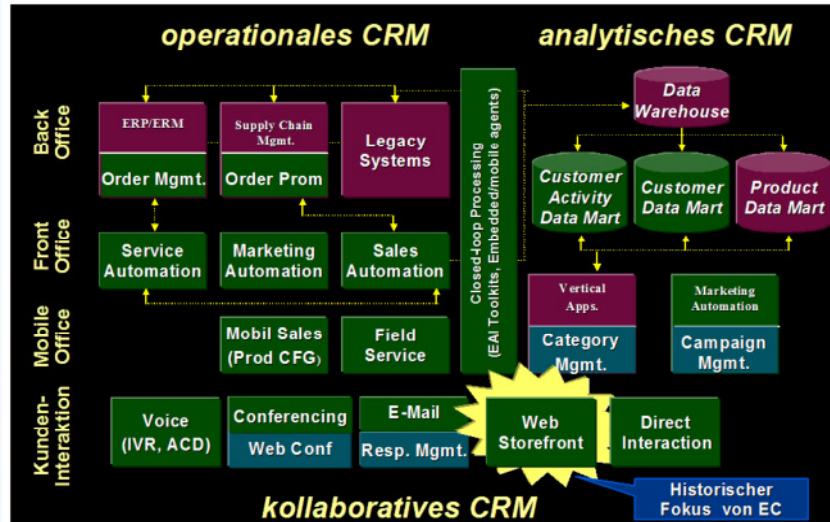
- **DWH** – Data Warehouse
- **BI** – Business Intelligence
- **CRM** – Customer Relationship Management
 - operational, analytical, collaborative
- **All the Others**: OLAP, OLTP, ETL, ERP, EAI

One Goal of the Lecture:

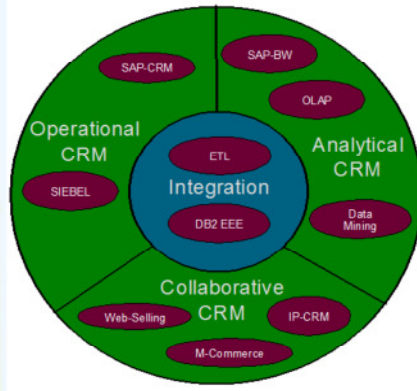
To learn the meaning of these ‘Catchwords’




Exercise1: The BI / CRM Topology



Exercise1: Practices of the BI/CRM Topology



Exercise 2: Basel II & RFID





Task: Prepare a report and present it at the next exercise session (next week, duration = 15 minutes). Information sources are newspaper or magazine articles or internet

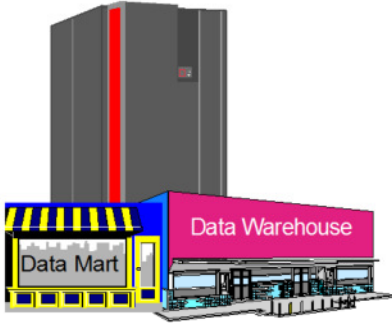
Theme: Give a definition (5 Minutes) and impact of these new trends on Data Warehousing (10 Minutes)


1. Basel II
2. RFID

Look also for examples of current projects in Germany

 **Lecture – DWH & DM** 

Lesson 2
Introduction to DWH Architecture





 Folie: 34 Dr. H. Völlinger, IBM

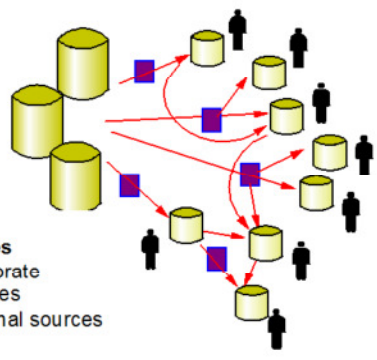
The following chapter gives an introduction into the architecture of a data warehouse. It explains the architectural ideas behind a Data Warehouse solution. It shows aspects of DWH architecture. It explains the different architectural concepts. What can go wrong, etc.....

In especially the following topics are covered:

- “Setting the Scene” with several steps on the way to a Central Data Warehouse (CDW)
- The possible architectural approaches from 0-Tier DWH to a 2-Tier DWH
- The “Big Picture” of a DWH architecture
- Not using architectural concepts: “What can Go Wrong?”
- The Data Warehouse Data Layers
- The IBM DWH Reference Architecture
- Example of a DWH Architecture for a Financial Market customer

 **Lecture – DWH & DM** 

Ad-Hoc Evolving DW Environments



Data Sources

- ✗ Corporate sources
- ✗ External sources

Problems

- Lack of credibility of the data
- Inconsistent information derivation
- Low productivity/High costs
- Complexity

Dr. H. Völlinger, IBM

Folie: 35

The foil shows the typical situation, which you will see in typical customer environments. This current scenario is build without control over several years in the enterprise..

Special solutions, special data population processes are developed for independent data marts over the years.

You can also see on the foil, that the ETL processes even get data from other data marts - --> complex process management.

Summary: The 5 main problems are:

1. Lack of credibility in the data
2. Inconsistent data derivation
3. Complexity of processes
4. Low productivity
5. High costs

Lecture – DWH & DM

Setting the Scene

Data Sources
 x Corporate sources
 x External sources

Data Warehouse Environment

Business Intelligence
 Decision support applications
 Information Analysis applications
 OLAP
 Knowledge Discovery applications
 Data Mining
 Statistical Data Analysis

Integrated collection of data
 "Corporate memory"
 Non-volatile data
 Business Data

Transient data → Business Information

e-business

IBM

Folie: 36

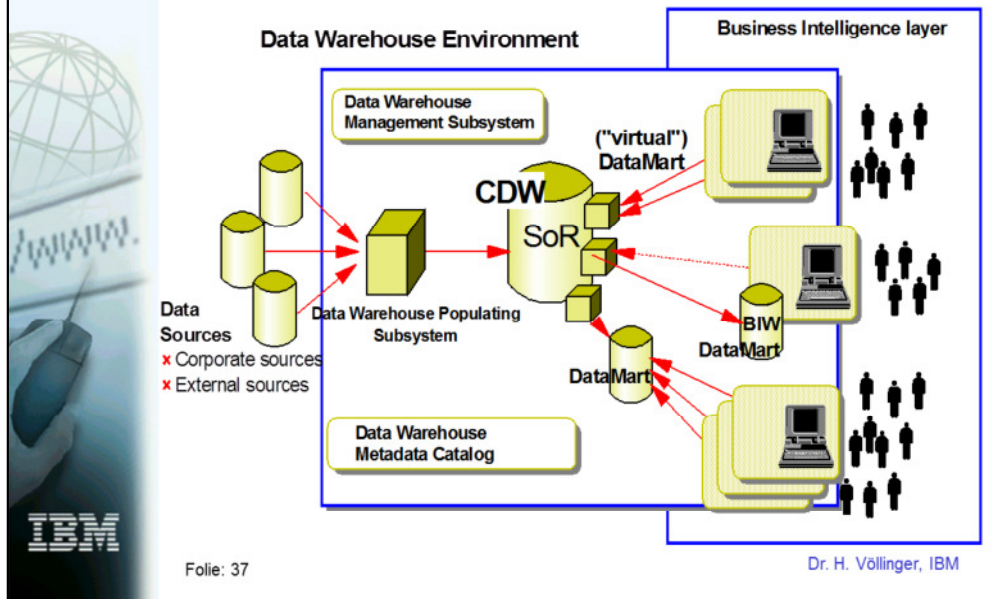
Dr. H. Völlinger, IBM

The picture show the first step – building a integrated enterprise wide data DWH. This results in a data pool with ‘good‘ data..

Main problems and goals which have to be solved for this step are:

- Storage of Data (non-volatile)
- Structure the data (data model)
- Integrate data from different sources
- Build historical data (historical data model)
- Consistent data (run check and plausibility processes)
- Access to the data (create an user- and security-concept)
- Build a metadata repository with technical and business metadata

Setting the Scene (Cont)



Folie: 37

Dr. H. Völlinger, IBM

See the next step in this foil:

After we have build the Central Data Warehouse (CDW), which has all detail data in it (therefore also called System of Record –SoR). Data Marts are build for special applications and user groups.

Lecture – DWH & DM

DW - Possible Approaches

The diagram shows three architectural approaches for data warehousing:

- Two-tier data warehouse:**
 - "Corporate data warehousing"
 - DataMarts with a "broad scope of interest"
- One-tier data warehouse:**
 - DataMarts and simple departmental solutions
- "Virtual" data warehouse:**
 - DataMarts and simple departmental solutions

Labels in the diagram include: Data Sources, Data Warehouse Environment, Data Warehouse, Management Information System, and IBM.

Folie: 38 Dr. H. Völlinger, IBM

What are the possible architectural approaches:

At the planning stage in a data warehouse project, it is important to decide what the scope of the warehouse is going to be. Historically data warehouse implementations have been global or enterprise-wide in scope, but this does not necessarily have to be the case. There are 3 main implementation alternatives for data warehousing:

- **Global Warehouse (2-tier architecture)** - here the primary responsibility for the overall architecture and implementation belongs to the IT department. Requirements and implementation priority would be based on the needs of the enterprise as a whole. This global warehouse could be physically centralized, decentralized, or logically centralized and physically distributed over multiple platforms and locations. The design could include support for any number of data marts, but these would not be the same as the stand-alone and dependent data marts already mentioned. The data marts are designed specifically to be part of the global data warehouse and are populated from it.
- **Stand-alone Data Mart (1-tier architecture)** - enables a department/workgroup to implement a data mart covering one particular business subject area, e.g. sales and marketing, or finance, with minimal, or no, impact on the IT department. It may require some technical skills, but these could be managed by the workgroup. This approach could also apply to a smaller organization that might not have the support of the IT department.
- **Virtual Data Warehouse (0-tier architecture)** - similar to the stand-alone data mart, except that data is not stored in a own database. The data from the source system are visualized/accessed by the applications. The data connectivity to the data sources managed by the IT department is required. These connectivity impacts the operational system. But implementation should still have minimal impact on the IT department. The department decides what data is accessed, the frequency of access, and may even provide tools and skills necessary to extract the data.

What can go Wrong?

1. **Data Outhouse** - Built too fast; full of dirty, incomplete, out-of-date data; no-one will use it.
2. **Data Basement** - A DW with poor access and/or performance. Not used much.
3. **Data Mausoleum** - Like the basement, but built with the finest hardware/software.
4. **Data Shack** - Will soon collapse due to insufficient funding and management commitment.
5. **Data Cottage** - Individual department's own personal DW's. (Outside the company's full DW architecture, hence not a Data Mart). Allowed to carry on, you end up with a cute data village.
6. **Data Jailhouse** - Built to such a high spec, with such tight controls, that no-one can get access to the data, even though IT will swear it's there.
7. **Data Tenement** - The result of a chaos- or ostrich-based implementation strategy, where some outsider is trusted to build the DW for you. It ends up satisfying no particular business requirements but you do get to say you have one.

See foil itself for a description:

The German equivalence words are:

- Data Outhouse – ‘Schellschuss‘
- Data Basement - ‘Grab‘
- Data Mausoleum - ‘Mausoleum‘
- Data Shack - ‘Schatten‘
- Data Cottage - ‘Hütte‘
- Data Jailhouse - ‘Gefängnis‘
- Data Tenement - ‘Mietwohnung‘

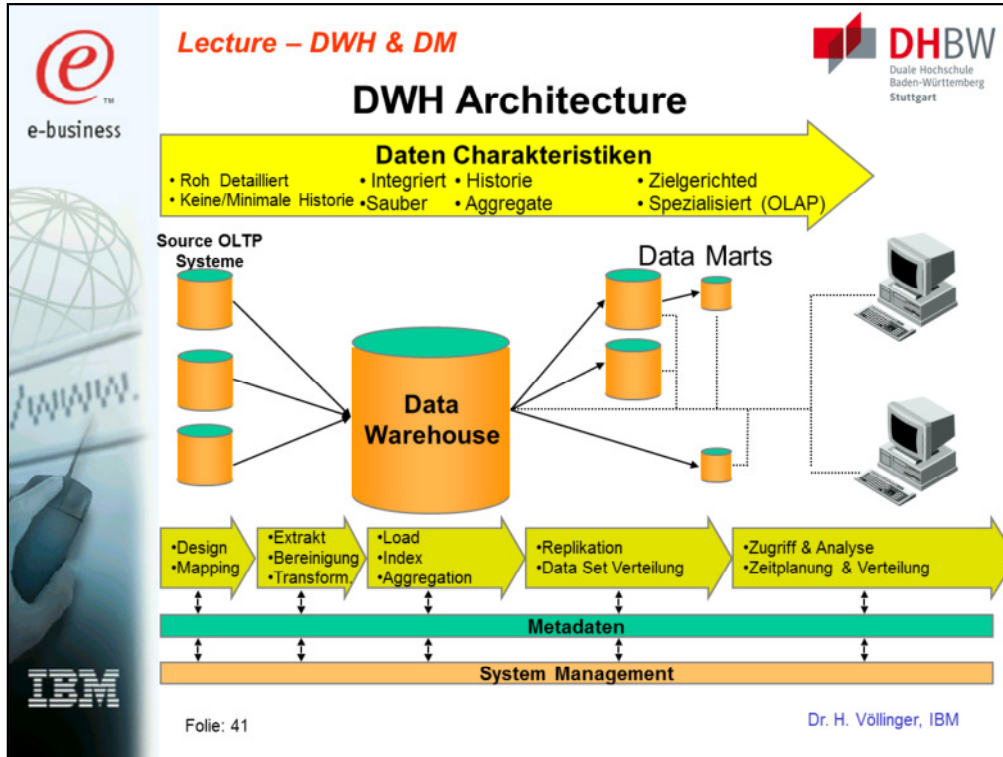
Experience shows That ...

80% of the cost of building and maintaining a Data Warehouse Environment usually relates to the Populating Subsystem ...



This is a well-known fact coming from the practice. The reason lies in the high efforts for data cleansing and data enrichment (i.e. data quality tasks).

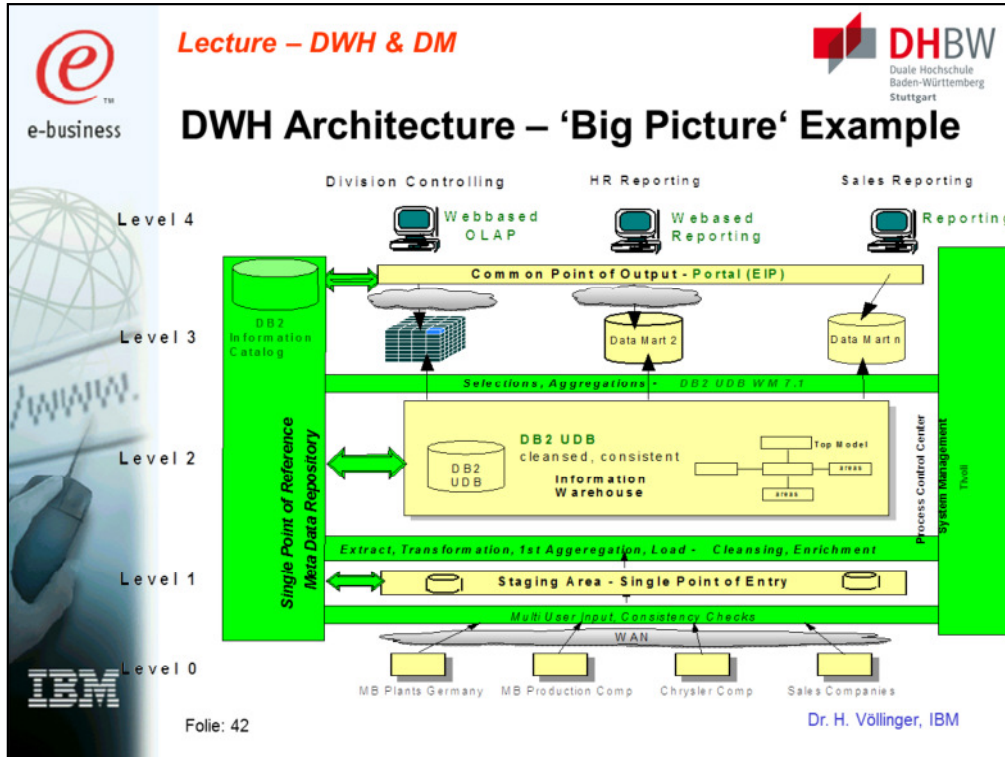
The data quality processes are very complex and therefore we need much effort to finish this work.



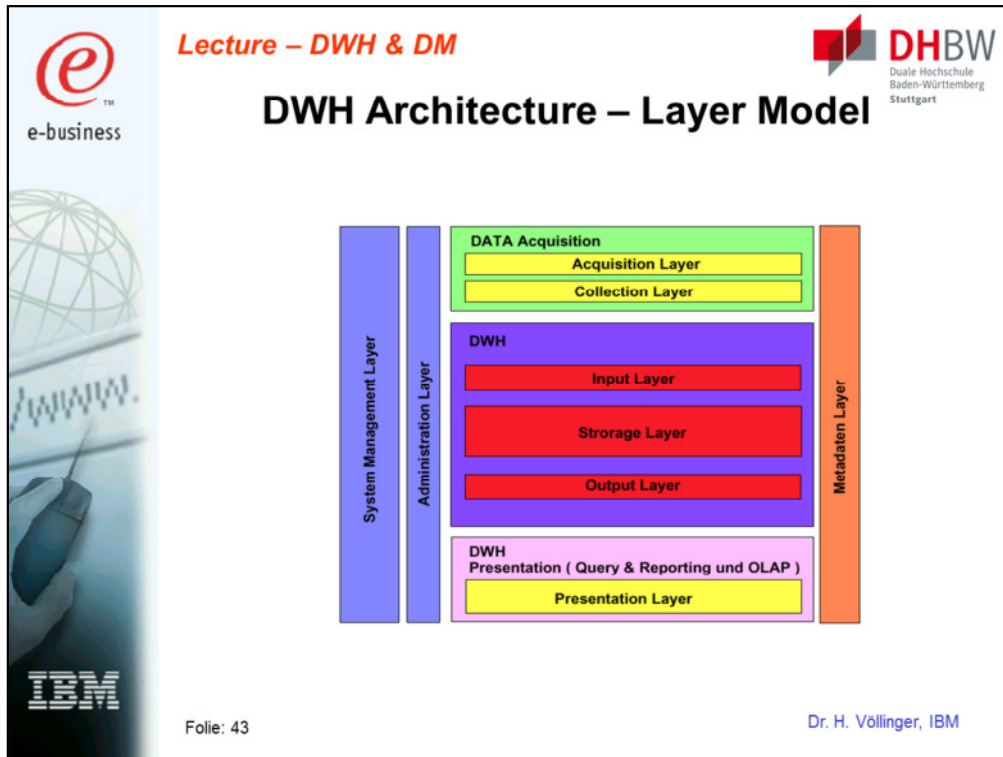
The foils show now the full scope architecture from both sides:

- Processes workflow (see bottom line)
- Data characteristics (see top line)

As general layers over the DWH, we see the Meta Data Layer and the System Management Layer.



Shows the same as before, but with ‘bottom-top’ approach of the layers.



The picture shows the data layers, which build up the DWH:

• **Data Acquisition:**

- collection all input data,
- checking data for its structure
- checking data for completeness

• **DWH Layer:**

• **Input Layer:** Sometimes also called Staging Area

- optional: loading in staging tables
- delta discovery, preparation for delta load
- reformat data

• **Storage Layer:** all database tables, Historization, ER Model

• **Output Layer:** Sub-area, optimized and aggregated data, Star- or Snowflake-Schema

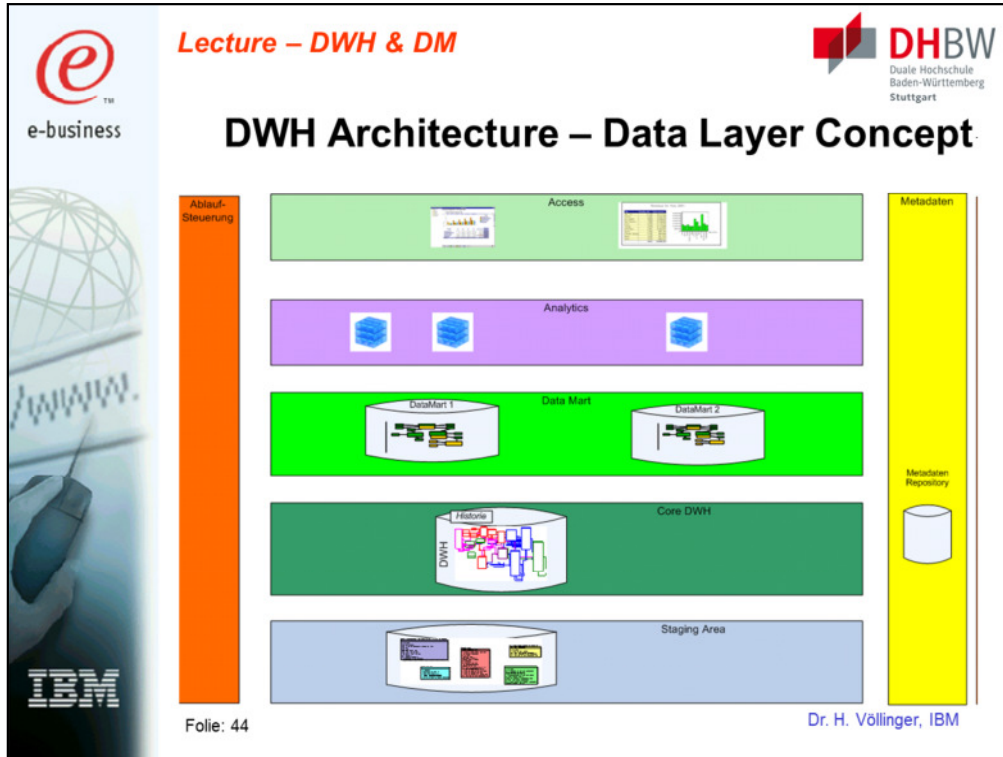
• **Presentation Layer:** all Front-end tools and applications, for example like:

- OLAP Tools, i.e. ROLAP, MOLAP, HOLAP ...
- Data Mining
- Query- and Reporting- Tools
- CRM Tools

• **Metadata Layer :** stores the metadata for all layers

• **System Management Layer:** Control of whole system for example. with TIVOLI

• **Administration Layer:** Scheduling and Control of all ETL and DWH processes



The tasks of the data layers and the properties of data in the layer is described:

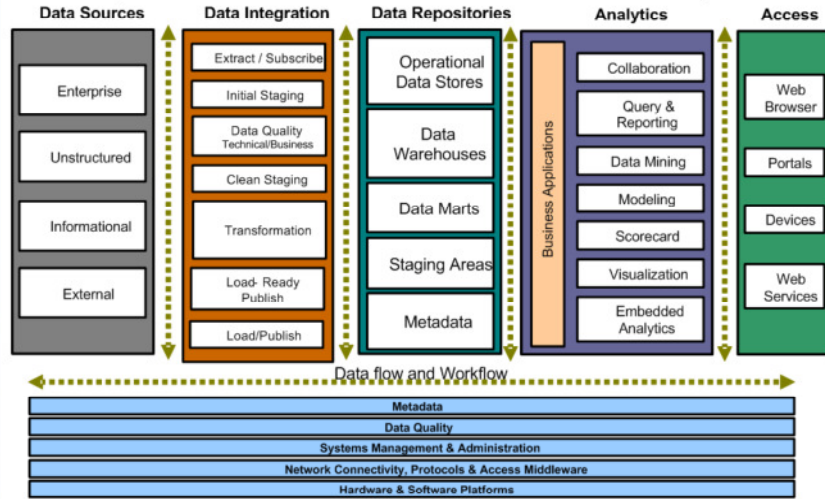
The Five Data Layers:

1. **„Staging Area (SA)“** - Collects the data from the source systems. The ETL process accomplishes some technical „Plausi“ rules during the load. Data are here put down as detail data in the data base. As soon as the completeness of the data is guaranteed, the data are loaded into the next data layer. Data are stored for technical reasons some days, afterwards the data are deleted.
2. **„Core DWH (CW)“** – At the next processing step in the layer „Core DWH “technical computations (transformations of data) take place. The most important technical transformation are accomplished here. The data are also filtered while the loading. Only the fields defined in the data model (see also chapters for data modeling) are to be filled. The data in the core DWH are historized and archived. The data in the DWH are not queried. Evaluations „do not reach through“ up to the Core DWH Layer.
3. **„Data Mart (DM)“** – Here the data are in a multi-dimensional data model. Measures and dimensions are defined here. This data model is optimized for the respective type of analysis. Evaluations access these data. Exemplarily for instance a OLAP cube (multi-dimensional data base) from these data can be built with Cognos. This data structure is then optimal for the demanded technical evaluation for an appropriate type of analysis.. „. The meta data Layer contains the Repository of meta data. This Repository can access and administer for instance over IBM meta data Workbench. These tools processes essentially technical meta data. Technical meta data can be stored also here. For this a further tool would be as for instance „IBM Business Glossary “necessary. Which meta data are exact that, in the concept to the meta data one defines.
4. **„Analytics (AN)“** – this layer contains all data for analytic evaluations. Exemplarily the OLAP can be cubes for a certain evaluation. The historization and archiving of these data are specified in the concept for the historization and archiving.
5. **„Access (AC)“** - this Layer contains reports in the form of Excel, HTML or pdf files. These files can be accessed by portal. The historization and archiving of these data are specified in the concept for the historization and archiving.

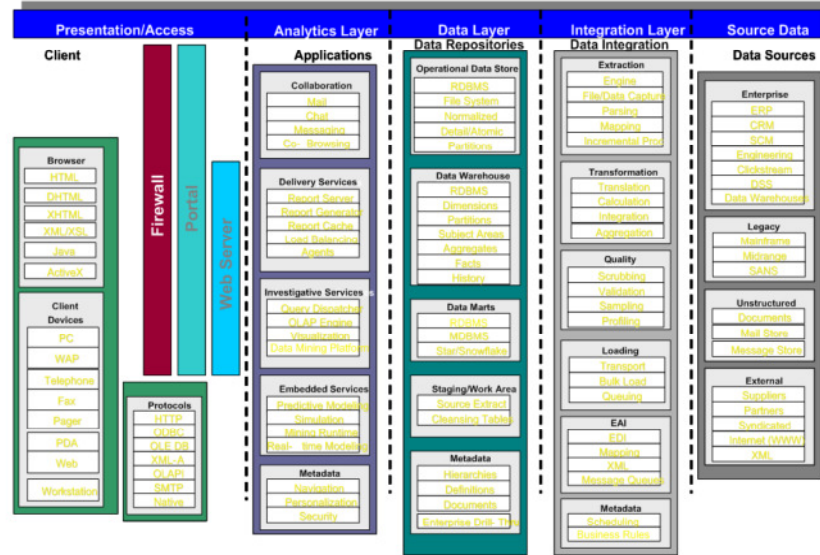
The **Metadata Layer** contains the repository of metadata. This repository can be accessed and administered for instance by IBM Metadata Workbench. This tool processes essentially technical metadata. Business Metadata can be stored also here. For this a further tool would be as for instance „IBM Business Glossary“. Which Metadata is stored here is defined in the concept of the Metadata Management.

The next layer is the **Layer for Flow Control („Ablaufsteuerung“)** - over this layer all processes of the DWH are started, steered and supervised. For this the necessary Metadata are stored in the Metadata repository in the Metadata layer.

IBM DWH Reference Architecture (outcome of IBM Unified Method Framework)



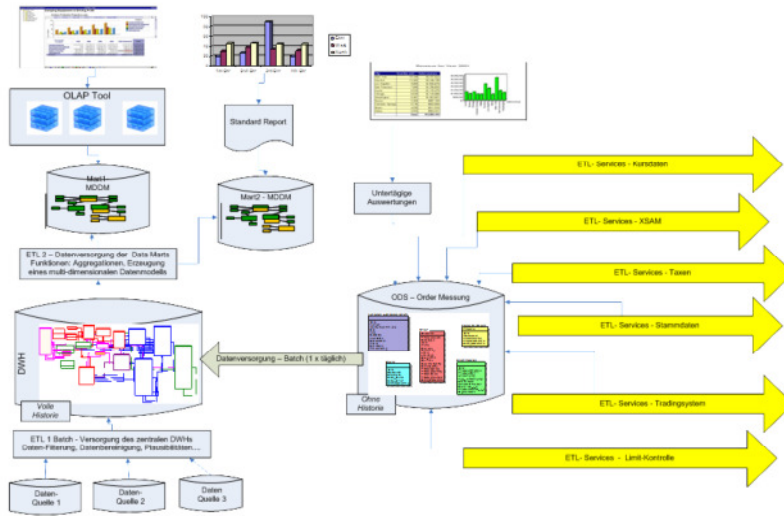
IBM DWH Reference Architecture – Details



Folie: 46

Dr. H. Völlinger, IBM

Example of a Financial Market DWH



Exercise to Lesson 2

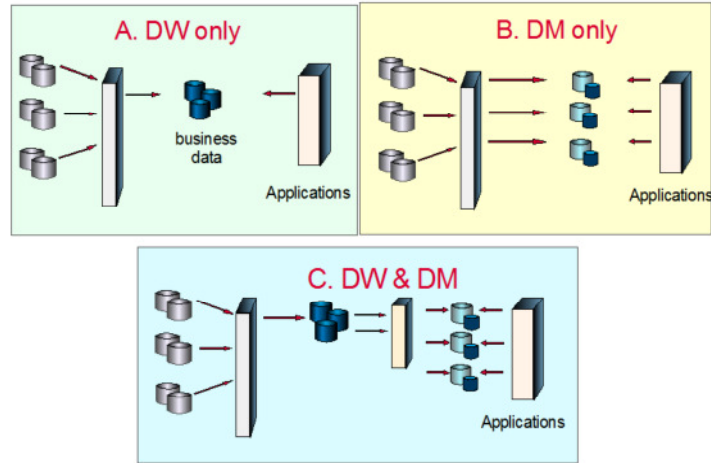
Task: Compare the three DWH architectures (DW only, DM only and DW & DM) in the next slide. List the advantages and disadvantages and give a detailed explanation for it. Find also a fourth possible architecture (hint: 'virtual' DWH)

Solution: Use a table of the following form:

	DW Only	DM Only	DW & DM	????	Explanation
Criteria 1	++	+	0	0	Text1
Criteria 2	--	-	+	-	Text2
Criteria 3					
....					



Exercise to Lesson 2 (cont)



Extra - Exercise to Lesson 2

Task: Build 6 groups (5 persons each). Prepare a small report about the following database themes. Concentrate only on basics. The presentation should just give an overview about the theme.

1. Non-relational databases (IMS, VSAM ...) (group1 & 4)
2. Relational DBMS (Group2 & 5)
3. SQL Basics (Group3 & 6)

For this you can use the material you learned in the former BA database lesson or use standard literature sources.

Goal: Present your report in the next exercise session (10 minutes duration). Send your solution to vgr@de.ibm.com

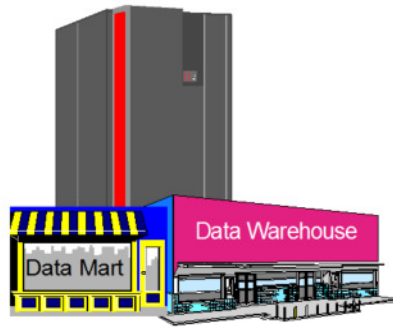




Solution of Exercise to Lesson 1 3 Students Presentations (10 minutes each)

Lesson 3

Overview about DBMS (i.e Relational Databases)



The four Goals of a DBMS

DBMS (Database Management Systems) are designed to achieve the following four main goals:

1. Increase Data Independence
 - Data & programs are independent
 - Change in data did not affect user programs
2. Reduce Data Redundancy
 - Data is only stored once
 - Different applications share the same centralized data
3. Increase Data Security
 - Authorize the access to the database
 - Place restrictions on operations that may be performed on data
4. Maintain Data Integrity
 - Same data is used by many users

In traditional systems data and programs are dependant. This means if the data is changed the programs have also to be changed.-→ in database systems the processing of data and programs are **independent**.

In a convenient file systems data (for example a customer address) is stored in many files (for example: customer record, purchase order, accounts receivable) -→ in a database the data is stored only once (**no data redundancy**)

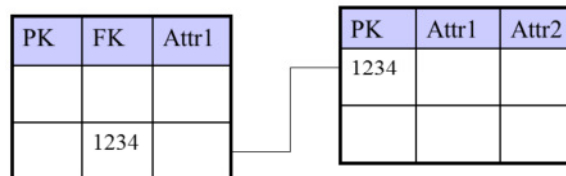
Databases use password protection to get access to DB, also further restrictions how to uses data (i.e. read or write protection)-→ **security**

Data Integrity is important, since the data are shared by many users

The Three Database Structures

Lets look on the three most popular structures of databases:

1. Hierarchical
 - Organized in the shape of a inverted tree
2. Network
 - Branches out from one or more roots in two or more directions
3. Relational
 - For example two dimensional tables that form relationships with each other



Folie: 54

Dr. H. Völlinger, IBM

These are the most popular DBMS. Beside there exist also some other DBs like:

VSAM - Virtual Storage Access Method, i.e. a key-sequenced data sets or files with an index containing extensive data set and volume information.

VSAM Overview: In the early 1970s, IBM introduced a collection of three data set organizations - sequential, indexed, and direct-access, together with the access methods and utilities to be used on the mainframe operating systems. This collection of data set organizations is called the Virtual Storage Access Method (VSAM).

The word *virtual* relates to the fact that VSAM was introduced at approximately the same time as the initial IBM virtual storage operating systems OS/VS1 and OS/VS2.

VSAM was developed to replace the Indexed Sequential Access Method (ISAM), which is a much older technology. ISAM has major processing overheads which IBM wanted to improve

What is VSAM?

VSAM is one of several access methods that defines the technique by which data is stored and retrieved. A GET/PUT interface is used to transfer data from a direct access storage device (DASD) to an application program. VSAM does not support data stored on tape. VSAM is used to organize and access data, and maintain information about this data which is stored or referenced in a catalogue.

VSAM data sets must be catalogued in an integrated catalogue facility (ICF) structure. Records are arranged by an index key or by relative byte addressing. VSAM uses direct or sequential processing of fixed and variable length records on DASD.

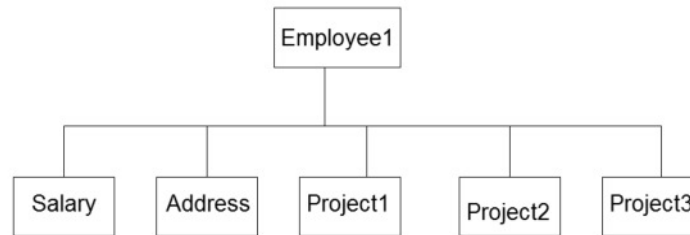
There are two major parts to VSAM: catalogue management and record management.

Hierarchical Database Structures

Organized in the shape of a inverted tree, see sample:

Each record may contain several information parts, for example:

- Employee : First Name, Last Name, Employee-Nr, ...
- Salary: Cross Pay , Income Tax,
- Address: Street, Town, Zip Code, ...
- Projectx: Start Date, Project Manager, Hours worked,



See example on the foil: An employee can work on more than one project only. There exist one parent data set and several children data sets.

The technological reason for this structure are the access possibilities of magnet-tapes.



Like the network model the **hierarchical model** stores its data in a series of **records**, which have a set of field values attached to it. It collects all the instances of a specific record together as a **record type**. These record types are the equivalent of tables in the relational model, and with the individual records being the equivalent of rows.

To create links between these record types, the hierarchical model uses **Parent Child Relationships**. These are a 1:N mapping between record types. This is done by using trees, like set theory used in the relational model, "borrowed" from mathematics.

Unlike the network model, the hierarchical model is only able to cope with a single tree, and is not able to cope with linking between branches or over multiple layers. The hierarchical model is more structured than the network model, since it only allows a single tree, with no links between layers in different branches of the tree. This means that it has a much more structured form than the relational model, this was done to improve throughput for transactions (adding, updating and deleting records) and to increase the simplicity of the interface for users.

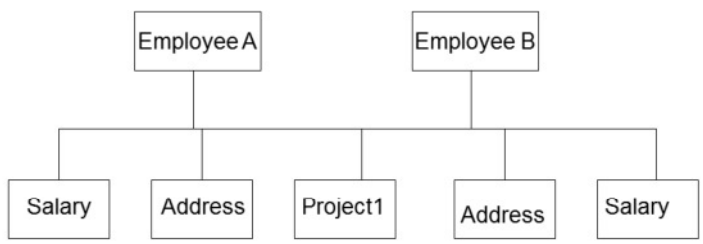
An example for such a database is IMS: **IMS- Information Management System**

More details: http://de.wikipedia.org/wiki/Information_Management_System

 **Lecture – DWH & DM** 

Network Database Structures

- More flexible
- Reduce Redundancy



Folie: 56 Dr. H. Völlinger, IBM

One can build for example out of two hierarchical employee databases (see before) one database, if the project is the same. Project1 occurs once, but both employees are connected to it---→ more flexible and reduce redundancy.

More flexible because each database has it's own set of rule which define the relationships between the records. A network database is similar to a hierarchical database, except the rules are not so strict

KEY CONCEPTS: The Network database model was first introduced in 1971 by CODASYL Data Base Task Group and because of this is sometimes called the DBTG Model. It is therefore a contemporary of the Relational Model, both in terms of its age and its basis in research done in the 1960's.

It is called the Network Model because it represents the data it contains in the form of a network of records and sets which are related to each other, forming a network of links.

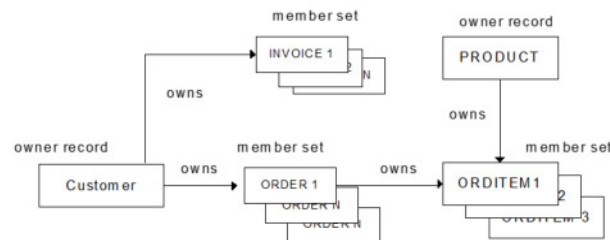
To do this it uses records, record types and set types which we shall discuss later. Because this model is only used in legacy systems, and is being phased out over time, we shall only cover the basic information about it in this section.

- **Records**, are sets of related data values. These are the equivalent of rows in the relational model. They store the name of the record type, the attributes associated with it and the format for these attributes
- **Record Types**, are set of records of same type. These are the equivalent of tables in the relational model.
- **Set Types**, are named, 1:N relationships between 2 record types. These do not have a direct parallel in the relational model, the closest comparison is to a query statement which has joined two tables together. This makes the network model faster to run certain queries, but does not allow the flexibility of the relational model.

More details: <http://en.wikipedia.org/wiki/IDMS>

Example – IDMS Database

- 'Network' Database
- Datasets are organized in 'sets'
- There are 'owner' and 'member'



The network model is not commonly used today to design database systems, however, there are a few instances of it being used by companies as part of a legacy system. Because it represents data in a format closer to the way in which it stores it than the other models, it can be faster than relational systems.

Unfortunately, this increase in speed is at the cost of its adaptability.

Migration from IDMS to DB2

Literature: IBM Redbook: 'DBMS CONVERSION GUIDE – IDMS TO DB2', GH20-7562-0

1. Normalization of the IDMS Datasets (DS) to 3. Normal Form
2. Creation of a DB2 table for the resulting IDMS DS
3. 'Translation' of an IDMS data-element into a DB2 column
4. Identification of a Primary Key for each table (IDMS owner DS)
5. Definition of a Foreign Key for each table, which belongs to IDMS 'member sets'
6. Treat special cases (support by expert skills)
7. Do much testing & validation



What is a Relational Database

- A relational database is a database that is perceived by the user as a collection of tables
- This user view is independent of the actual way the data is stored
- Tables are sets of data made up from rows and columns

Hydrogen	H	1	1.0079
Helium	He	2	4.0026
Lithium	Li	3	6.941
Berylium	Be	4	9.01218
Boron	B	5	10.81
Carbon	C	6	12.011
Nitrogen	N	7	14.0067
Oxygen	O	8	15.9994

You are probably familiar with the term relational database.

In a relational database the user views everything as a set of tables.

These tables are made up of rows and columns. They might be very large.

Relational Database Structures

- Very flexible --→ create views
- Keep the data secure (use views)
- Relation between tables
- Primary & Foreign Keys
- 'Normalization'

Employee Table

EmpNo	Workdep	Empname	Position
321-412	100	Jones	Programmer
456-673	100	Simpson	Analyst

Project Table

Project	Projlead	ProjName
100-04	321-412	Maintenance
200-15	456-673	Personnel

These are the two tables which represent employee and project information, salary and address information could be stored in other tables.

You can create a relation between the two tables by the EmpNo and the ProjLead attribute.

Views and Joins

Tables can be related to each other by the data they hold (called joins)

NAME	DEPT CODE	SEX	EXTN
Fred	10	M	4429
Mary	15	F	4642
George	15	M	4242
Susan	10	F	4559
Betty	12	F	4114

DEPT CODE	MANAGER	DEPT NAME
10	Mrs Smith	Accounts
12	Mr Black	Sales
15	Miss White	Purchasing

NAME	EXTN	MANAGER
Mary	4642	Miss White
George	4242	Miss White

Views are ways of looking at data from one or more tables

The information you are looking for is hold in different tables.

To get the relevant Information you are looking for, one has to **join** over different tables using a common key, here: DEPT CODE

These joins can be **performance bottlenecks** for large tables.

The process of combining data from two or more tables is called **joining tables**. The database manager forms all combinations of rows from the specified tables. For each combination, it tests the join condition.

A join condition is a search condition, with some restrictions. For a list of restrictions refer to the SQL Reference.

Note that the data types of the columns involved in the join condition do not have to be identical; however, they must be compatible. The join condition is evaluated the same way as any other search condition, and the same rules for comparisons apply.

If you do not specify a join condition, all combinations of rows from tables listed in the FROM clause are returned, even though the rows may be completely unrelated. The result is referred to as the **cross product** of the two tables.

The two main types of joins are **inner joins** and **outer joins**. So far, in all of our examples we have used the inner join.

Lecture – DWH & DM

The Database Join Strategies

- **Cross Product**
- **Inner Join**
- **Outer Join**
 - Left outer Join
 - Right outer Join
 - Full Outer Join

Folie: 62

Dr. H. Völlinger, IBM

Inner joins keep only the rows from the cross product that meet the join condition.

If a row exists in one table, but not the other, the information is not included in the result table.

Outer joins are a concatenation of the inner join and rows from the left table, right table, or both tables that are missing from the inner join.

When you perform an outer join on two tables, you arbitrarily assign one table as the left table and the other one as the right table.

There are three types of outer joins:

1. **left outer join** includes the inner join and the rows from the left table that are not included in the inner join.

2. **right outer join** includes the inner join and the rows from the right table that are not included in the inner join.

3. **full outer join** includes the inner join and the rows from both the left and right tables that are not included in the inner join.

Use the SELECT statement to specify the columns to be displayed.

In the FROM clause, list the name of the first table followed by the keywords LEFT OUTER JOIN, RIGHT OUTER JOIN or FULL OUTER JOIN.

Next you need to specify the second table followed by the ON keyword. Following the ON keyword, specify the join condition to express a relationship between the tables to be joined.

Summary: Relational Database - Features

1. Simplicity

- All data values are in tables
- All operations result also in tables

2. Automatic Navigation

- No need to know the 'path' to find the data
- Need only to know column and table name

3. Security / Integrity

- Access rules stated how you can perform data
- Referential Integrity – Transactions get always same results
- Recovery of lost and damaged data

4. Dynamic Definition

- No system take-down for adding new data or indexes
- Access to DB, even when Unloading or Reloading is done



Motivation & Introduction to Normal Forms

As Normalization of a relational database schema we understand the splitting of a relation (i.e. a table) via normalization algorithms in more new relations in respect of its functional dependencies.

The relation (i.e. table) will then go to first (1NF), second (2NF) or third (3NF)... Normal Form.

We will learn about the meaning of 1NF, 2NF and 3NF in the following foils.

Normal Forms are important, to:

- Reduce Redundancy
- Support Maintenance
- Reduce Inconsistency
-

of the data.

The mostly used Normal Forms in Data Warehousing are:

- 1. Normalform (1NF)
- 2. Normalform (2NF)
- 3. Normalform (3NF)
- Boyce-Codd (BCNF)
- 4. Normalform (4NF)
- 5. Normalform (5NF)

During the Normalization we split the columns of tables inside a database in new columns, for example we split addresses in post-code, town, street and house number, or we relate these columns with other tables for example a customer is related with a customer table via a concrete key.

The goal of the normalization is to reduce Redundancy (same or double information), to omit Anomalies (i.e. two data content which could not be true for both). Normalization will reduce the effort for maintenance of a database and will also guarantee the consistence of the data..

For example when we change the address of a customer in a database we have to look for all address information of the customer in a database and have to change them all. In a normalized database we have only to change one dataset, because the customer is stored only once in the database.

.....

In some cases (which we will see later) it makes sense not to normalize the database. Because of :

- Better query performance
- More simpler queries
- or some more reasons

The First Normal Form (1NF)

Rule:

A relation is in First Normal Form (1NF), when each attribute of the relation is 'atomic' and the relation is free of 'repeating groups'.

- 'Atomic' – the value of an attribute can no be split in more meaningful values. For example 'Adresse' is not an atomic attribute, because it could be split in 'PLZ', 'Ort', 'Straße' and 'Hausnummer'
- 'Repeating Groups' means that attributes which holds the same or similar information should be stored in another relation. For example { ..., Telefon1, Telefon2, Telefon3,... }. In this case is the repeating group three attributes, which hold all the same information and are dependant on each other.

Original Rule (from Codd):

All columns in a relation are only dependant from the key.

Action:

Eliminate repeating values in one atom and repeating groups.

For further definition of normal forms we need some formal notations:

Functional dependency: Attribute B has a functional dependency on attribute A if, for each value of attribute A, there is exactly one value of attribute B. For example, Employee Address has a functional dependency on Employee ID, because a particular Employee Address value corresponds to every Employee ID value. An attribute may be functionally dependent either on a single attribute or on a combination of attributes.

It is not possible to determine the extent to which a design is normalized without understanding what functional dependencies apply to the attributes within its tables; understanding this, in turn, requires knowledge of the problem domain.

Trivial functional dependency: A trivial functional dependency is a functional dependency of an attribute on a superset of itself.

{Employee ID, Employee Address} \rightarrow {Employee Address} is trivial, as is {Employee Address} \rightarrow {Employee Address}.

Full functional dependency: An attribute is fully functionally dependent on a set of attributes X if it is 1. functionally dependent on X, and 2. not functionally dependent on any proper subset of X.

{Employee Address} has a functional dependency on {Employee ID, Skill}, but not a full functional dependency, for it is also dependent on {Employee ID}.

Transitive dependency: A transitive dependency is an indirect functional dependency, one in which $X \rightarrow Z$ only by virtue of $X \rightarrow Y$ and $Y \rightarrow Z$.

Example for First Normal Form ('Atomic')

The following table is not in First Normal Form (*examples are from WIKIPEDIA).
The attribute 'Album' has information about *Interpret* and *CD Title*

CD_Lieder		
CD_ID	Album	Titelliste
4711	Anastacia - Not That Kind	{1. Not That Kind, 2. I'm Outta Love, 3. Cowboys & Kisses}
4712	Pink Floyd - Wish You Were Here	{1. Shine On You Crazy Diamond}

The attributes 'Album' and 'Titelliste' are split in atomic attributes. 'Titelliste' is split in 'Track' and 'Titel'.

CD_Lieder				
CD_ID	Albumtitel	Interpret	Track	Titel
4711	Not That Kind	Anastacia	1	Not That Kind
4711	Not That Kind	Anastacia	2	I'm Outta Love
4711	Not That Kind	Anastacia	3	Cowboys & Kisses
4712	Wish You Were Here	Pink Floyd	1	Shine On You Crazy Diamond



Example for First Normal Form ('Repeating Groups')

The following table is not in First Normal Form (1NF) – there are "Repeating Row Groups":

PO#	SUP#	SupName	Item#	ItemDescription	\$/Unit	Quant
12345	023	Acme Toys	XT108	Buttons	2.50	100
			XT111	Buttons	1.97	250
			BW322	Wheels	6.20	50
12346	094	Mitchells	BW641	Chassis	19.20	100
			BW832	Axles	3.40	220

By adding the duplicate information in the first three row to the empty row cells, we get five complete rows in this table, which have only atomic values. So we have First Normal Form. (1NF).

PO#	SUP#	SupName	Item#	ItemDescription	\$/Unit	Quant
12345	023	Acme Toys	XT108	Buttons	2.50	100
12345	023	Acme Toys	XT111	Buttons	1.97	250
12345	023	Acme Toys	BW322	Wheels	6.20	50
12346	094	Mitchells	BW641	Chassis	19.20	100
12346	094	Mitchells	BW832	Axles	3.40	220





e-business

Lecture – DWH & DM



Example - First Normal Form ('Anomalies')

Requirement: One „Prüfer“ always has only one „Fach“

PNR	Fach	Prüfer	Student MATNR	Name	Geb	Adr	Fachbereich	Dekan	Note
3	Elektronik	Richter	123456	Meier	010203	Weg 1	Informatik	Wutz	1
			124538	Schulz	050678	Str 1	Informatik	Wutz	2
4	Informatik	Schwinn	245633	Ich	021279	Gas. 2	Informatik	Wutz	1
			246354	Schulz	050678	Str 1	Informatik	Wutz	1
5	TMS	Müller	856214	Schmidt	120178	Str 2	Informatik	Wutz	3
			369852	Pitt	140677	Gas. 1	BWL	Butz	1

INPUT 'Anomalien'

How to insert a student , who never have done an examination?

DELETE 'Anomalien'

When you delete the student Pitt, you loose the information about 'Dekan BWL'

CHANGE 'Anomalien'

When a student changes his address, you have to change the street in several places.

Remark: There is another hidden problem in the data of this table? Any idea?

Folie: 68

Dr. H. Völlinger, IBM



Second Normal Form (2NF)

Rule:

- The table must be in 1NF.
- None of the non-prime attributes of the table are functionally dependent on a part (proper subset) of a candidate key; in other words, all functional dependencies of non-prime attributes on candidate keys are full functional dependencies.
- For example, in an "Employees' Skills" table whose attributes are Employee ID, Employee Address, and Skill, the combination of Employee ID and Skill uniquely identifies records within the table.
- Given that Employee Address depends on only one of those attributes – namely, Employee ID – the table is not in 2NF.
- Note that if none of a 1NF table's candidate keys are composite – i.e. every candidate key consists of just **one** attribute – then we can say immediately that the table is in 2NF.

Action:

Regroup columns dependent on only one part of the composite key



Example for Second Normal Form

The following table is not in second Normal Form (*examples are from WIKIPEDIA):
The primary key of the relation exists of the fields *CD_ID* and *Track*. The fields *Albumtitel* and *Interpret* are dependant from the field *CD_ID* but not from the field *Track*.

CD_Lieder				
<i>CD_ID</i>	Albumtitel	Interpret	<i>Track</i>	Titel
4811	Not That Kind	Anastacia	1	Not That Kind
4811	Not That Kind	Anastacia	2	I'm Outta Love
4811	Not That Kind	Anastacia	3	Cowboys & Kisses
4712	Wish You Were Here	Pink Floyd	1	Shine On You Crazy Diamond

We split the data in the table in two tables: *CD* und *Lieder*. The table *CD* consists only of fields which are full functional dependant from *CD_ID*

CD		
<i>CD_ID</i>	Albumtitel	Interpret
4811	Not That Kind	Anastacia
4712	Wish You Were Here	Pink Floyd

Lieder		
<i>CD_ID</i>	<i>Track</i>	Titel
4811	1	Not That Kind
4811	2	I'm Outta Love
4811	3	Cowboys & Kisses
4712	1	Shine On You Crazy Diamond



Third Normal Form (3NF)

Rule:

- The table must be in 2NF.
- Every non-prime attribute of the table must be non-transitively dependent on every candidate key.
- A violation of 3NF would mean that at least one non-prime attribute is only *indirectly* dependent (transitively dependent) on a candidate key.
- For example, consider a "Departments" table whose attributes are Department ID, Department Name, Manager ID, and Manager Hire Date; and suppose that each manager can manage one or more departments. {Department ID} is a candidate key. Although Manager Hire Date is functionally dependent on the candidate key {Department ID}, this is only because Manager Hire Date depends on Manager ID, which in turn depends on Department ID. This transitive dependency means the table is not in 3NF.

Action:

Regroup non-key columns representing a fact about another non-key column



Example for Third Normal Form

The following table is not in third normal form (*examples are from WIKIPEDIA):
 The field *Interpret* of the table CD is dependant from *CD_ID*, but *Gründungsjahr* is also dependant from *Interpret* and therefore transitive dependant from *CD_ID*.

CD			
<i>CD_ID</i>	Albumtitel	Interpret	Gründungsjahr
4811	Not That Kind	Anastacia	1999
4713	Bad	Michael Jackson	1971
4712	Wish You Were Here	Pink Floyd	1965

We split the relation, such that the dependant data are in its own tables. The key of the new table is a foreign key in the old table.

CD			Künstler	
<i>CD_ID</i>	Albumtitel	Interpret	<i>Interpret</i>	Gründungsjahr
4811	Not That Kind	Anastacia	Anastacia	1999
4713	Bad	Michael Jackson	Michael Jackson	1971
4712	Wish You Were Here	Pink Floyd	Pink Floyd	1965

Summary – Normal Forms 1NF-3NF

Normalization is the process of streamlining your tables and their relationships (compare also the examples in the lesson and the exercises)

1. Normal Form (1NF)

- **Action:** Eliminate repeating values in one atom and repeating groups
- **Rule:** Each column must be a fact about the key

2. Normal Form (2NF)

- **Action:** Regroup columns dependent on only one part of the composite key
- **Rule:** Each column must be a fact about the whole key

3. Normal Form (3NF)

- **Action:** Regroup non-key columns representing a fact about another non-key column
- **Rule:** Each column must be a fact about nothing but the key

“the key, the whole key, and nothing but the key - so help me Codd“

The database community has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through five (fifth normal form or 5NF). In practical applications, you'll often see 1NF, 2NF, and 3NF along with the occasional 4NF. Fifth normal form is very rarely seen.

Normalization Benefits

- **Excellent logical design methodology**
- **Translation from logical to physical design**
- **Reduced data redundancy**
- **Protection against update & delete problems**
- **Ability to add/delete tables/columns and rows without major changes**
- **Smaller tables which provide more physical room for data**



Exercise to Lesson 3 (First Part)

1. **Question:** From what you have seen for network DB, choose two statements:

1. Structure is like an inverted tree
2. Structure may have two or more roots
3. Record only have one parent record
4. Deletion rules vary depending on the system

2. **Question:** Choose two statements for Relational Database

1. The data is structured like an inverted tree
2. The data is structured in two dimensional tables
3. Its structure is the most flexible of the three
4. Each database have a unique set of deletion rules



Exercise to Lesson 3 (Second part)

Build all Join Strategies with the following tables:

- Cross Product
- Inner Join
- Outer Join
 - Left Outer Join
 - Right Outer Join
 - Full Outer Join

SAMP_PROJECT

Name	Proj
Haas	AD3100
Thompson	PL2100
Walker	MA2112
Lutz	MA2111

SAMP_STAFF

Name	Job
Haas	PRES
Thompson	MANAGER
Lucchessi	SALESREP
Nicholls	ANALYST

Folie: 76

Dr. H. Völlinger, IBM

The following example produces the **cross product** of the two tables. A join condition is not specified, so every combination of rows is present:

```
SELECT  
SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,  
SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM SAMP_PROJECT, SAMP_STAFF
```

The following example produces the **inner join** of the two tables. The inner join lists the full-time employees who are assigned to a project:

```
SELECT  
SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,  
SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM SAMP_PROJECT, SAMP_STAFF  
WHERE SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

Alternately, you can specify the inner join as follows:

```
SELECT  
SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,  
SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM  
SAMP_PROJECT INNER JOIN SAMP_STAFF ON  
SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

Exercise to Lesson 3 (Third part)

Do the normalization steps 1NF, 2NF and 3NF to the following unnormalized table (show also the immediate results):

PNR	Fach	Prüfer	Student MATNR	Name	Geb	Adr	Fachbereich	Dekan	Note
3	Elektronik	Richter	123456	Meier	010203	Weg 1	Informatik	Wutz	1
			124538	Schulz	050678	Str 1	Informatik	Wutz	2
4	Informatik	Schwinn	245633	Ich	021279	Gas. 2	Informatik	Wutz	1
			246354	Schulz	050678	Str 1	Informatik	Wutz	1
5	TMS	Müller	856214	Schmidt	120178	Str 2	Informatik	Wutz	3
			369852	Pitt	140677	Gas. 1	BWL	Butz	1

Exercise to Lesson 3 (Fourth part)

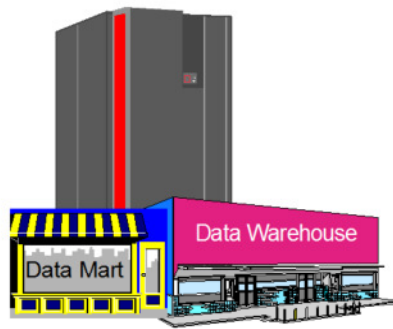
Do the normalization steps 1NF, 2NF and 3NF to the following un-normalized table (show also the immediate results):

Prerequisites: Keys are PO# and Item#, SupName = Funct (Sup#) , Quant = Funct (Item#,PO#) and S/Unit=Funct (Item#)

<u>PO#</u>	<u>SUP#</u>	<u>SupName</u>	<u>Item#</u>	ItemDescription	S/Unit	Quant
12345	023	Acme Toys	XT108	Buttons	2.50	100
			XT111	Buttons	1.97	250
			BW322	Wheels	6.20	50
12346	094	Mitchells	BW641	Chassis	19.20	100
			BW832	Axles	3.40	220

Lesson 4

Introduction to Basics of SQL



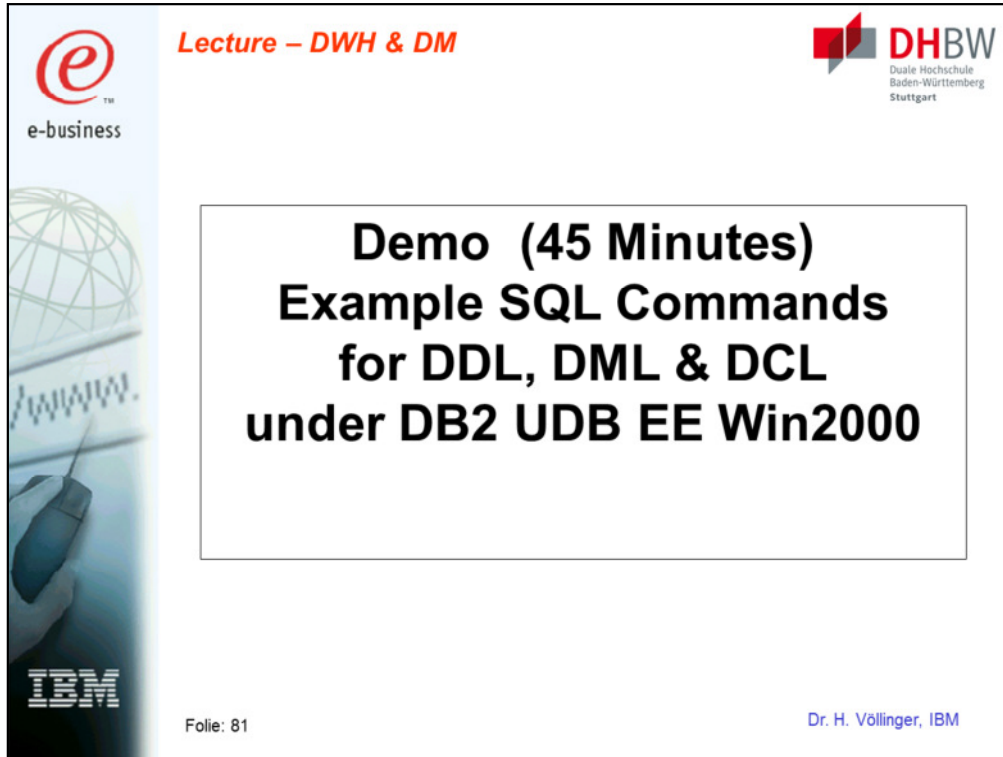
Introduction to SQL

SQL is divided into three major categories:

1. **DDL** – Data Definition Language
 - Used to **create**, **modify** or **drop** database objects
2. **DML** – Data Manipulation Language
 - Used to **select**, **insert** , **update** or **delete** database data (records)
3. **DCL** – Data Control Language
 - Used to provide data object access control



The standard language of relational database access is Structured Query Language (SQL). SQL is not a programming language. It was designed for the single purpose of accessing structured data.



The slide header features several logos and text elements. On the left, there is an 'e-business' logo with a stylized '@' symbol and a vertical image of a globe and a mouse. In the top right, the 'DHBW' logo is displayed, representing the Dual University of Baden-Württemberg, Stuttgart. The title 'Demo (45 Minutes) Example SQL Commands for DDL, DML & DCL under DB2 UDB EE Win2000' is centered in a white box. At the bottom left, the IBM logo is visible, and at the bottom right, the name 'Dr. H. Völlinger, IBM' is listed. The slide number 'Folie: 81' is located at the bottom center.

Lecture – DWH & DM

**Demo (45 Minutes)
Example SQL Commands
for DDL, DML & DCL
under DB2 UDB EE Win2000**

IBM

Folie: 81

Dr. H. Völlinger, IBM

In this demo we will see how you can create and manipulate tables and views in DB2 Universal Database. The relationship of tables and views is explored through diagrams and examples. This demo covers:

- Creating Tables and Creating Views
- Inserting Data
- Changing Data
- Deleting Data
- Selecting Data (also with using Column functions)
- Joining tables
- Connecting to database
- Define and change authority

Examples of DDL commands

Show a few examples with DB2 UDB EE for Win2000 of DDL commands, i.e.

- **create table**
- **alter table**
- **drop table**
-



Creating Tables

Create your own tables using the CREATE TABLE statement, specifying the column names and types, as well as *constraints*. Constraints are not discussed in this course. The following statement creates a table named PERS:


CREATE TABLE PERS

```
(ID SMALLINT NOT NULL, NAME VARCHAR(9),  
DEPT SMALLINT WITH DEFAULT 10, JOB CHAR(5),  
YEARS SMALLINT, SALARY DECIMAL(7,2),  
COMM DECIMAL(7,2), BIRTH_DATE DATE)
```


This statement creates a table with no data in it. The next section describes how to insert data into a new table. As shown in the example, you specify both a name and a data type for each column.

NOT NULL is optional and may be specified to indicate that null values are not allowed in a column. Default values are also optional.

There are many other options you can specify in a CREATE TABLE statement, such as unique constraints or referential constraints. For more information about all of the options, see the CREATE TABLE statement in the SQL Reference. © Copyright IBM Corp. 1993, 2000




Lecture – DWH & DM



Examples of DML commands

Show a few simple examples with DB2 UDB EE for Win2000 of DML commands, i.e.

- **select**
- **insert** (also from other tables)
- **update**
- **delete**
-



Folie: 83

Dr. H. Völlinger, IBM

Inserting Data: When you create a new table, it does not contain any data. To enter new rows into a table, you use the INSERT statement. This statement has two general forms:

- With one form, you use a VALUES clause to specify values for the columns of one or more rows. The next three examples insert data into tables using this general form.
- With the other form, rather than specifying VALUES, you specify a fullselect to identify columns from rows contained in other tables and/or views.

Fullselect is a select statement used in INSERT or CREATE VIEW statements, or following a predicate. A fullselect that is enclosed in parentheses is commonly referred to as a subquery.

Depending on the default options that you have chosen when creating your table, for every row you insert, you either supply a value for each column or accept a default value. The default values for the various data types are discussed in the SQL Reference.

The following statement uses a VALUES clause to insert one row of data into the PERS table:

```
INSERT INTO PERS VALUES (12,'Harris',20,'Sales',5,18000,1000,'1950-1-1')
```

The following statement uses the VALUES clause to insert three rows into the PERS table where only the IDs, the names, and the jobs are known. If a column is defined as NOT NULL and it does not have a default value, you must specify a value for it.

The NOT NULL clause on a column definition in a CREATE TABLE statement can be extended with the words WITH DEFAULT. If a column is defined as NOT NULL WITH DEFAULT or a constant default such as WITH DEFAULT10, and you do not specify the column in the column list, the default value is inserted into that column in the inserted row. For example, in the CREATE TABLE statement, a default value was only specified for DEPT column and it was defined to be 10. Hence, the department number (DEPT) is set to 10 and any other column that is not explicitly given a value is set to NULL.

```
INSERT INTO PERS (NAME,JOB,ID) VALUES ('Swagerman','Prgmr',500), ('Limoges','Prgmr',510), ('Li','Prgmr',520)
```

For inserting data from other tables, just execute the following command:

```
INSERT INTO PERS (ID,NAME,DEPT,JOB, YEARS,SALARY) SELECT ID,NAME,DEPT,JOB, YEARS,SALARY FROM STAFF WHERE DEPT =38
```

Examples of DML commands (Part 2)

Show now a more 'complex' example, like joining the information about several tables, i.e.

- **select** ... (from several tables)

Create views -> provide the information as a fix table to a clearly defined user group

- **create view**...

Using functions like **MAX** and **MIN** to create a more complex query:

- **select** Col1, **MAX**(Col2) **AS** Maximum,...

- To do a **join about two tables**, for example the table **org** and **staff** in the DB2 **SAMPLE** database. You have to take care that the where condition makes sense:

First example:

```
select deptname,name from org, staff  
where manager=id
```

Second example:



```
select e.empno,e. firstnme,e.lastname,d.deptno,d.deptname  
from employee e, department d  
where d.deptno=e.workdept order by e.empno
```

- To **create a view**, compare the following example:

```
create view staff_only  
as select id, name, dept, job,years  
from staff where job<>'Mgr' and dept=20
```

- To usage of **column functions** like **MAX**, **MIN** and **COUNT** can be seen in the following example:

```
select workdept, MAX(salary) as maximum, MIN(salary) as minimum  
from employee  
group by workdept  
having COUNT(*)>4
```





Lecture – DWH & DM

Examples for DCL commands

Show a few examples with DB2 UDB EE for Win2000 of DCL commands, i.e.

- **connect** to database
- **grant**
- **revoke**
- **db2audit**
-



Folie: 85

Dr. H. Völlinger, IBM

Connecting to a Database:

You need to connect to a database before you can use SQL statements to query or manipulate it. The CONNECT statement associates a database connection with a user name.

For example, to connect to the SAMPLE database, type the following command in the DB2 command line processor :

CONNECT TO SAMPLE USER USERID USING PASSWORD

(Be sure to choose a user ID and password that are valid on the server system.)

In this example, USER is USERID and USING is PASSWORD. The following message tells you that you have made a successful connection:

Database Connection Information

Database product =DB2/NT 7.1.0

SQL authorization ID =USERID

Local database alias =SAMPLE

Once you are connected, you can start manipulating the database. For further details on connections, refer to the CONNECT statement in the *SQL Reference*.

Investigating Errors:

Whenever you make a mistake typing in any of the examples, or if an error occurs during execution of an SQL statement, the database manager returns an error message. The error message consists of a message identifier, a brief explanation, and an SQLSTATE. SQLSTATE errors are error codes common to the DB2 family of products. SQLSTATE errors conform to the ISO/ANSI SQL92 standard. For example, if the user ID or password had been incorrect in the CONNECT statement, the database manager would have returned a message identifier of SQL1403N and an SQLSTATE of 08004. The message is as follows:SQL1403N The username and/or password supplied is incorrect.

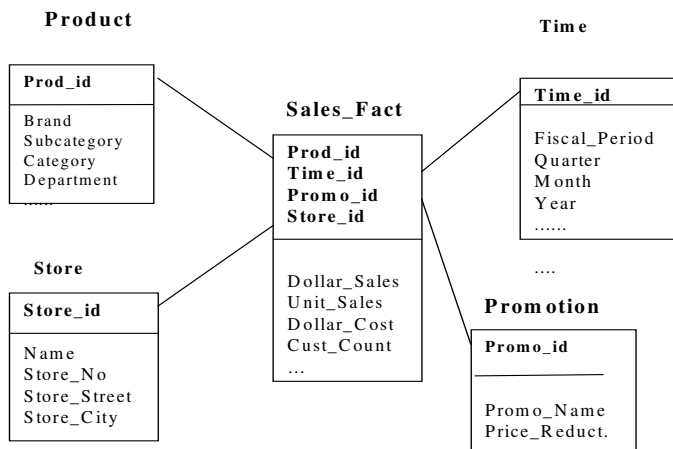
Exercise to Lesson 4 (First part)

Compare the situation from R. Kimball's Grocery example:

Folie: 86

Dr. H. Völlinger, IBM

Consider the following Star Schema:



Exercise to Lesson 4 (Part 1)

Build the SQL, such that the result is the following report, where time condition is the Fiscal_Period = '4Q95':

Brand	Dollar Sales	Unit Sales
Axon	780	263
Framis	1044	509
Widget	213	444
Zapper	95	39

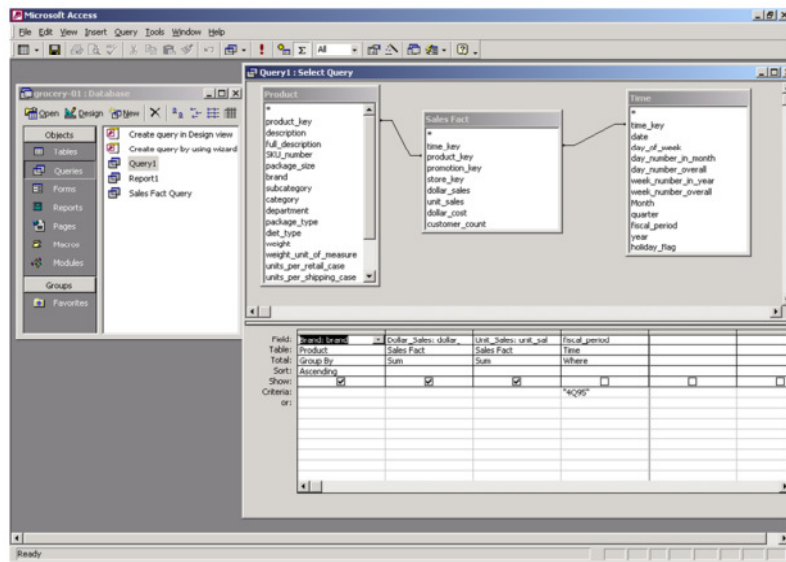
Solution with Standard SQL(for example with DB2):

```
SELECT p.brand AS Brand, Sum(s.dollar_sales) AS Dollar_Sales, Sum(s.unit_sales) AS
Unit_Sales
FROM sales_fact s, product p, time t
WHERE p.product_key = s.product_key
      AND s.time_key = t.time_key
      AND t.fiscal_period="4Q95"
GROUP BY p.brand
ORDER BY p.brand
```

By using the **SQL Wizard** (Design View) in the database **Microsoft Access**, we see the following 'Access SQL':

```
SELECT Product.brand AS Brand, Sum([Sales Fact].dollar_sales) AS Dollar_Sales,
Sum([Sales Fact].unit_sales) AS Unit_Sales
FROM ([Sales Fact] INNER JOIN [Time] ON [Sales Fact].time_key = Time.time_key) INNER
JOIN Product ON [Sales Fact].product_key = Product.product_key
WHERE (((Time.fiscal_period)="4Q95"))
GROUP BY Product.brand
ORDER BY Product.brand;
```

Solution with MS Access SQL Wizard



Folie: 88

Dr. H. Völlinger, IBM

Running the above SQL results in the following table:

Brand	Dollar_Sales	Unit_Sales
American Corn	39872,23	41544
Big Can	36375,16	39643
Chewy Industries	33765,57	43612
Cold Gourmet	64938,83	26145
Frozen Bird	70598,67	28611
National Bottle	23791	26099
Squeezable Inc	65020,68	41949
Western Vegetable	50685,69	27998

Exercise to Lesson 4 (Part 2)

Advanced Study about concepts in DWH:

Explain: What is “Referential Integrity” (RI) in a Database ?



Sub-Questions:

1. What means RI in a Data Warehouse?
2. Should one have RI in a DWH or not? (collect pro and cons)

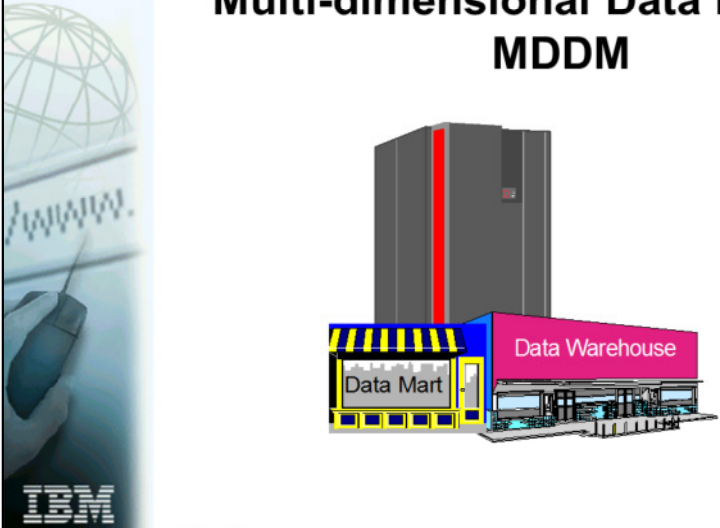
Find explanations and arguments in DWH forums or articles about this theme in the internet or in the literature.



<http://stackoverflow.com/questions/2188352/when-is-referential-integrity-not-appropriate>

 **Lecture – DWH & DM** 

Lesson 5
Multi-dimensional Data Modeling
MDDM



IBM

Folie: 90 Dr. H. Völlinger, IBM

The following chapter gives an introduction into the methods and concepts of Multi Dimensional Data Modeling (MDDM) in the process of building a data warehouse. It explains the architectural ideas behind a Multi Dimensional Model.

It is build to support business needs in arranging data in such a structure that business users can easy ask questions about their business success. This success is measured in “measures and dimensions”. These concepts together with 4 other basic concepts are shown and in the “Six Basic Concepts” of MDDM.

So to build an initial multi-dimensional data model, the following six base elements have to be identified:

1. **Measures**
2. **Dimensions**
3. **Grains of dimensions and granularities of measures and facts**
4. **Facts**
5. **Dimension hierarchies**
6. **Aggregation levels**

Lecture – DWH & DM

Requirements Analysis- Context

Requirements Analysis

- Source Data Models
- Existing DW Data Models
- Template Models
- Existing Data models of Data Marts

Initial Dimensional Models (Analysis Models)

Business Directory (Metadata)

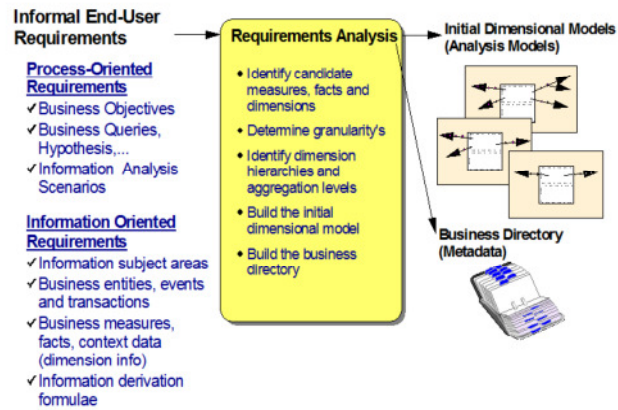
Folie: 91

Dr. H. Völlinger, IBM

Deliverables ('the output of the process') consist of a combination of:

- So called **initial dimensional data models**, symbolically represented by the box-and-arrow diagrams on the foil. We will show later how to best construct such initial dimensional models
- The **business directory or metadata definitions** of all element of the MDDM

Requirements Analysis - Activities



Folie: 92

Dr. H. Völlinger, IBM

End user requirements suitable for a data warehouse modeling project can be classified in two major categories:

1. **Process oriented requirements**
2. **Information oriented requirements**

Sample Query

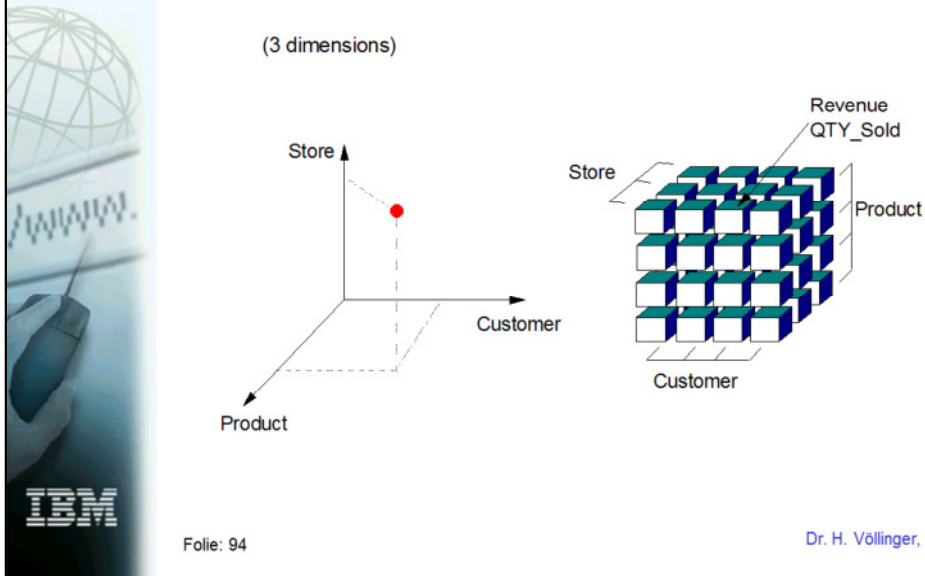
- Query:
"What are the net sales, in terms of revenue (dollars) and quantities of items sold,
Per product,
Per store and sales region,
Per customer and customer sales area,
Per day as well as aggregated over time,
Over the last two weeks?"
- Evaluation entails viewing historical sales figures from multiple perspectives such as:
 - Sales (overall)
 - Sales per product
 - Sales per store and per sales region
 - Sales per customer and customer sales area
 - Sales per day and aggregated over time
 - Sales and aggregated sales over given time periods



For developing the base concepts of multi-dimensional data modeling, we will use the sample query presented here.

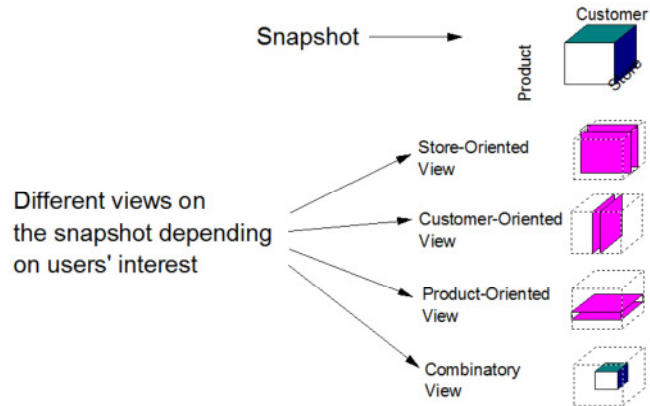
The query is a rather straightforward example of a sales analysis query.

Representation of the Query as a Cube



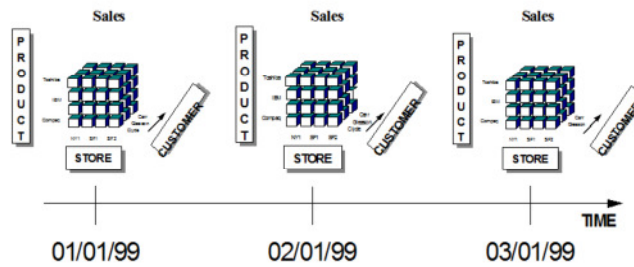
Multi-dimensional data models could be presented using cubes or using a mathematical notation technique representing points in a multi-dimensional space, for example: $QTY_SOLD = F(S,P,C,t)$

Presentation of the Query as a Cube : Usage



Hypercube Representation

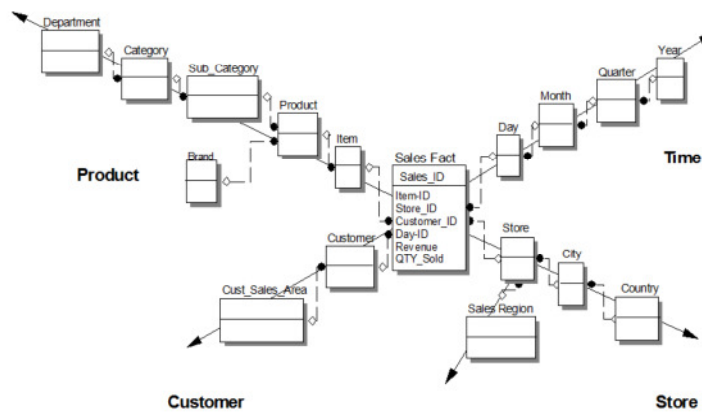
(4th dimension)



Hypercube:
Good visual representation for three dimensions
Difficult to use, when more than four dimensions

If more than three dimensions are present in the solution, the “cube“ or three-dimensional space representation is no longer usable.
The principle of the cube can be extended to “hyper-cube“

Sample Multidimensional Representation Usable for Any Number of Dimensions

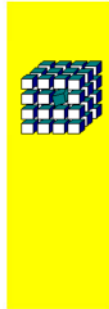


Folie: 97

Dr. H. Völlinger, IBM

The schema presented here show how we will produce the initial dimensional data models we have mentioned before.

The Six Base Concepts of MDDM



- Measures
- Dimensions
- Granularity
- Facts
- Dimension Hierarchies
- Aggregation Levels



To build an initial multi-dimensional data model, the following six base elements have to be identified:

1. Measures
2. Dimensions
3. Grains of dimensions and granularities of measures and facts
4. Facts
5. Dimension hierarchies
6. Aggregation levels

We will produce our initial multi-dimensional data models base on an analysis of given queries → **query oriented approach**

Also other approaches are possible.

Multidimensional Modeling - Base Concepts (1 of 6)

- Measure

- A measure is a data item which information analysts use in their queries to measure the performance or behavior of a business process or a business object

- Sample types of measures

- Quantities
- Sizes
- Amounts
- Durations, delay
- And so forth

Measures

Sales
Sales_ID
Item_ID
Store_ID
Customer_ID
Day_ID
Revenue
QTY_Sold

Using a query oriented approach, candidates can be identified by analyzing the given end-user query.

The sample query presented before basically include two such measures:

1. quantity of items sold
2. sales revenue

Remark about KPI: KPI – Key Performance Indicator is a common known synonym for the most important measures of a business.



e-business



Identify Candidate Measures

- Query-Oriented Approach
 - Perform a smart, not a mechanical analysis of the available queries
- Candidate Measures are
 - **Numeric, "Continuously" Valued**
 - But not every numeric attribute is a candidate measure
 - Distinguish measures from discrete valued numeric attributes which are part of dimensions
 - **Involved in Aggregation Calculations**
- Examples
 - Revenue (sales query)
 - Quantity sold (sales query)

Measures

Multidimensional Modeling - Base Concepts (2 of 6)

- Dimension
 - A dimension is an entity or a collection of related entities, used by information analysts to identify the context of the measures they work with
 - Examples: Product, Customer, Store, Time
- Dimensions are referred to through so-called Dimension keys
- Dimensions contain
 - Dimension entities
 - Dimension attributes
 - Dimension hierarchies
 - Consisting of one or more aggregation levels

Sales
Sales_ID
Item_ID
Store_ID
Customer_ID
Day_ID
Revenue
QTY_Sold

Dimensions

Measures require dimensions for their interpretation. As an example, the measure ,sales revenue‘ only make sense if we know this value for special item, special customer, at a day and in a certain store -> we got the four dimensions: **item, store, customer and time.**

Identify Candidate Dimensions

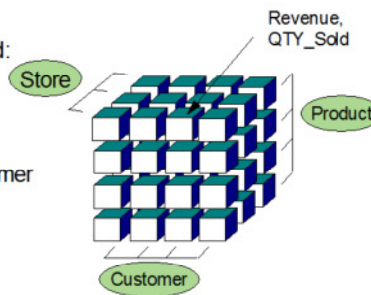
- Query-Oriented Approach

- A new dimension shows up each time a query indicates that a measure is aggregated in some way
- Who, what, where, when, how, ... questions

- Examples

- Revenue and Quantity sold:

- Who > Customer
- What > Product
- Where > Store
- When > Time
- How > Product by Customer



Dimensions



Modeling - Base Concepts (3 of 6)

- The **grain** of a dimension is the lowest level of detail available within that dimension
 - Product grain: Item
 - Customer grain: Customer
 - Store grain: Store
 - Time grain: Day
- The **granularity** of a measure is determined by the combination of the grains of all its dimensions

Granularity



For example the granularity of the measure QTY_SOLD is: **(item, customer, store, day)**.

Fine granularity enables fine analysis possibilities, but on the other side it has a big impact on the size of the Data Warehouse.

About Granularity - Example

Low Granularity Hides Information

Revenue	1/1	2/1	3/1	4/1
Sales Region 1	65	55	75	50
Sales Region 2	88	42	40	40
Sales Region 3	25	60	39	99

Revenue	1/1	2/1	3/1	4/1
Store 1	20	15	35	35
Store 2	18	13	5	5
Store 3	12	17	14	5
Store 4	15	10	21	5

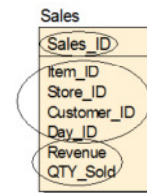
Sales Region 1

Granularity

Here we see an example how fine granularity can show 'hidden' information, like that stores in a region are better performers than other.

Multidimensional Modeling - Base Concepts (4 of 6)

- Fact
 - A fact is a collection of related measures and their associated dimensions, represented by the dimension keys
 - Example: Sales
 - A fact can represent a business object, a business transaction or an event which is used by the information analyst
- Facts contain
 - A Fact Identifier
 - Linking them with the dimensions
 - Dimension Keys
 - Measures
 - Supportive Attributes



Facts



Identify Candidate Facts

- Query-Oriented Approach:
 - Consolidating Measures into Candidate Facts
 - Candidate measures can be consolidated in facts when they have identical dimensions and granularities

	Dimension 1	Dimension 2	Product	Customer	Store	Time	(...)
Measure 1							
Measure 2							
Revenue			Item	Customer	Store	Day	
Quantity Sold			Item	Customer	Store	Day	
Measure 3							
(...)							

FACT

Facts

Folie:
106

Ur. H. Vollinger, IBM

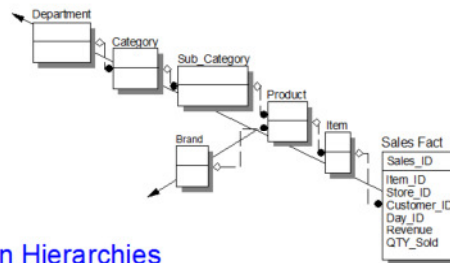
Using a query-oriented approach for finding initial modeling artifacts, facts may be identified through consolidating measures which have similar dimensions and identical granularities.

Measures which have same dimensions and same granularity are candidates to be moved in the same fact table.

There exists also more complex methods to find facts.

Multidimensional Data Modeling - Base Concepts (5 of 6)

- Dimensions consist of one or more **dimension hierarchies**
- Examples: Hierarchies in the Product Dimension
 - Product Classification Hierarchy ("Merchandising Hierarchy")
 - Branding Hierarchy
 - ...



Dimension Hierarchies

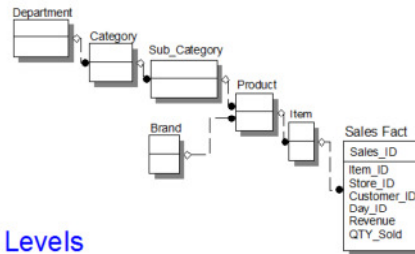
Dimensions consist on one or more dimension hierarchies.

For example the product dimension in our example has two hierarchies:

1. 'Merchandising' hierarchy
2. 'Product Brand' hierarchy

Multidimensional Data Modeling - Base Concepts (6 of 6)

- Each dimension hierarchy can include several aggregation levels
- Examples: Aggregation Levels in the Product Classification Hierarchy
 - Items -> Product -> Sub-Category -> Category -> Department



Aggregation Levels

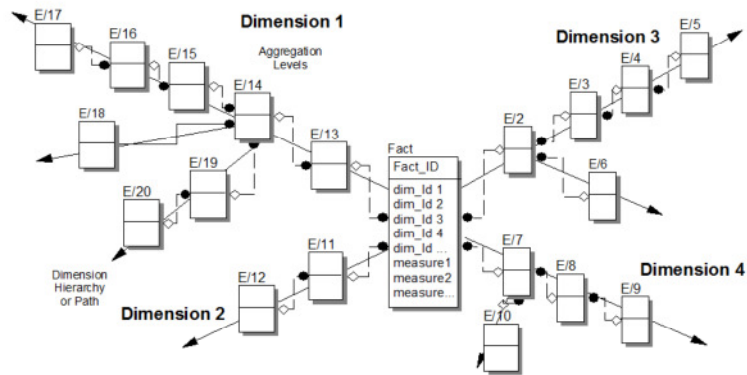
Finally, each dimension hierarchy usually include several aggregation levels.

For example:

1. **Item**: 4-pack Duracell AA Alkaline Batteries.
2. **Product**: Duracell AA Alkaline Batteries
3. **Sub-category**: AA Alkaline Batteries
4. **Category**: Batteries
5. **Department**: Supplies

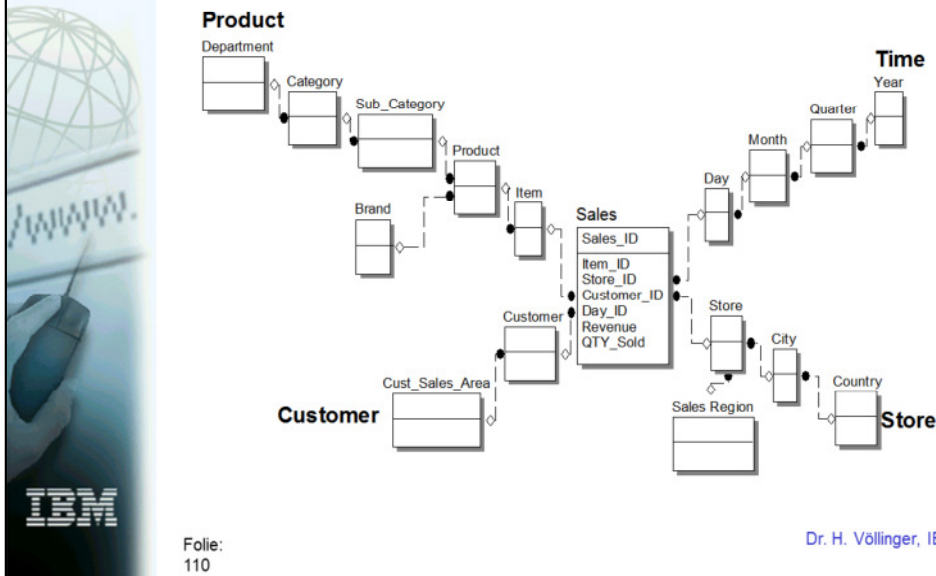
Dimension hierarchies and aggregation levels are used by users when drilling up or down.

Initial Multidimensional Model - Summary



With these candidate modeling artifacts, initial dimensional models can be built, as illustrated above.

Initial Multidimensional Model - Example



This foil shows **the six base concepts** as they apply to our Sales Query and the initial model that corresponds with that query.

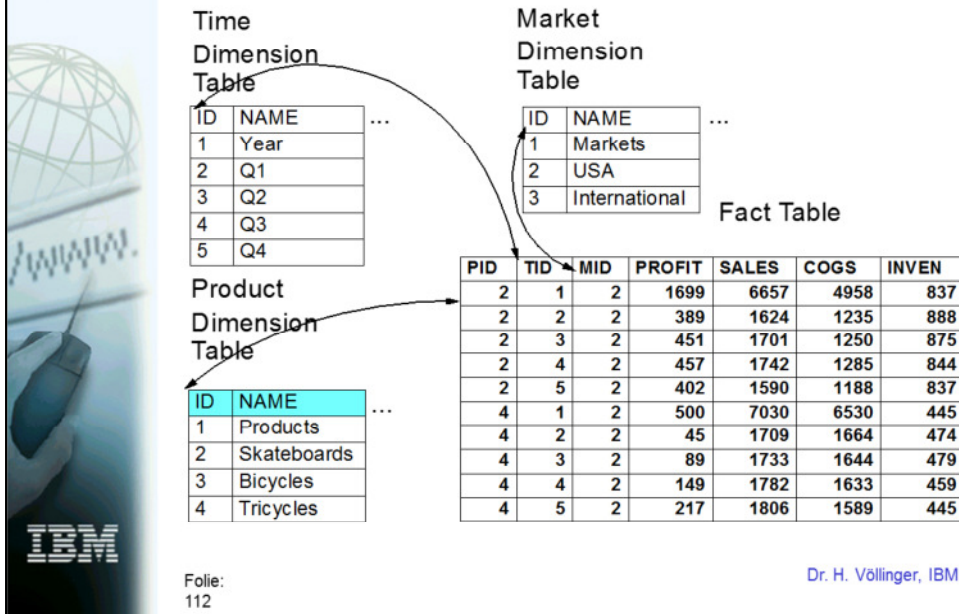
What is a Star Schema ?

- A star schema is a way to represent multidimensional data in a relational database
- *Dimension tables* store descriptive information about members and their relationships
- *Fact table* stores business data
 - Generally several orders of magnitude larger than any dimension table
 - One key column joined to each dimension table
 - One or more data columns
- Multidimensional queries can be built by joining fact and dimension tables
- Some products use this method to make a relational OLAP (*ROLAP*) system



This is a repetition of the chapter about multi-dimensional data modeling (MDDM).

Star Schema Example



Folie:
112

Dr. H. Völlinger, IBM

This is an example of a simple 3-dim. STAR SCHEMA.

If you use SQL, the downside is that joining and selecting is done each time a query is build. Especially with large DB, this will cost lot of processing and make the system slow ----→ **Killer Query**



Demo (20 Minutes) ErWin Data Modeling Tool

Exercise to MDDM (Part1)



Compare ER Modelling (**ER**) with multidimensional data models (**MDDM**), like **STAR** or **SNOWFLAKE** schemas (see appendix page):

Compare in IBM Redbook 'Data Modeling Techniques for DWH' (see DWH lesson homepage) Chapter 6.3 for ER modeling and Chapter 6.4 for MDDM

Build a list of advantages and disadvantages for each of these two concepts, in the form of a table:

ER Model	MDDM Model
Criteria1 ++	Criteria5 ++
Crit.2 +	Crit.6 +
Crit.3 -	Crit.7 -
Crit.4 --	Crit.8 --

The BI logical data models are compared here:

Entity-relationship: An entity-relationship logical design is data-centric in nature. In other words, the database design reflects the nature of the data to be stored in the database, as opposed to reflecting the anticipated usage of that data.

Because an entity-relationship design is not usage-specific, it can be used for a variety of application types: OLTP and batch, as well as business intelligence.

This same usage flexibility makes an entity-relationship design appropriate for a data warehouse that must support a wide range of query types and business objectives.

Star schema: The star schema logical design, unlike the entity-relationship model, is specifically geared towards decision support applications.

The design is intended to provide very efficient access to information in support of a predefined set of business requirements. A star schema is generally not suitable for general-purpose query applications.

A star schema consists of a central fact table surrounded by dimension tables, and is frequently referred to as a multidimensional model. Although the original concept was to have up to five dimensions as a star has five points, many stars today have more than five dimensions. The information in the star usually meets the following guidelines:

- A fact table contains numerical elements
- A dimension table contains textual elements
- The primary key of each dimension table is a foreign key of the fact table
- A column in one dimension table should not appear in any other dimension table.

Exercise to MDDM (Part2)

Compare MDDM Model schemas **STAR** and **SNOWFLAKE**::

Compare in IBM Redbook'Data Modeling Techniques for DWH' (see DWH lesson homepage) Chapter 6.4.4.

Build a list of advantages and disadvantages for each of these two concepts, in the form of a table:

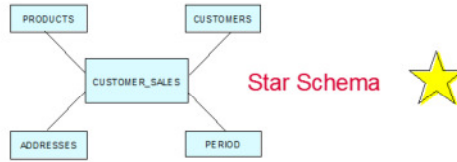
STAR Model	SNOWFLAKE Model
Criteria1 ++	Criteria5 ++
Crit.2 +	Crit.6 +
Crit.3 -	Crit.7 -
Crit.4 --	Crit.8 --

Snowflake Schema The snowflake model is a further normalized version of the star schema. When a dimension table contains data that is not always necessary for queries, too much data may be picked up each time a dimension table is accessed. To eliminate access to this data, it is kept in a separate table off the dimension, thereby making the star resemble a snowflake.

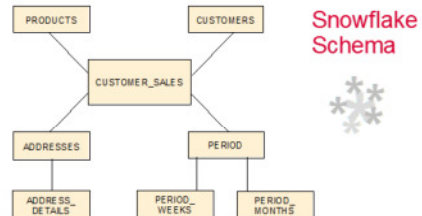
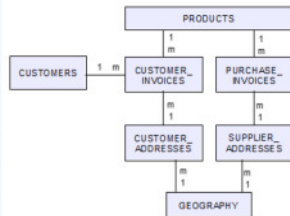
The key advantage of a snowflake design is improved query performance. This is achieved because less data is retrieved and joins involve smaller, normalized tables rather than larger, denormalized tables. The snowflake schema also increases flexibility because of normalization, and can possibly lower the granularity of the dimensions.



The disadvantage of a snowflake design is that it increases both the number of tables a user must deal with and the complexities of some queries. For this reason, many experts suggest refraining from using the snowflake schema. Having entity attributes in multiple tables, the same amount of information is available whether a single table or multiple tables are used.

Appendix to MDDM Lesson Exercises

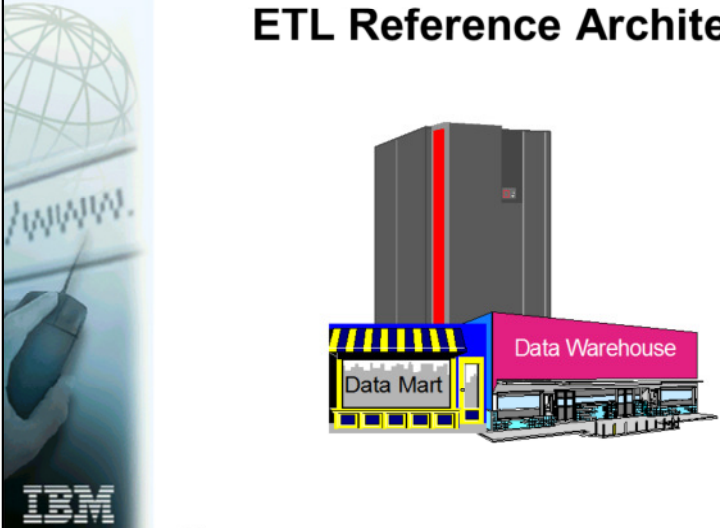


Entity-Relationship



 **Lecture – DWH & DM** 

Lesson 6
ETL Reference Architecture



IBM

Folie:
117

Dr. H. Völlinger, IBM

The following chapter gives an introduction into the ETL Reference Architecture as part of the overall architecture of a data warehouse. It explains the architectural ideas behind an ETL solution. It shows in detail the different components of an ETL Reference Architecture. It also explains the three steps of a successful ETL Strategy.

In especially the following topics are covered:

- Three steps of a successful ETL Strategy
 1. Discover
 2. Prepare
 3. Transform
- ETL components PSS and Pipe
- Metadata Layer
- Warehouse Management
- The different Data Layers
 1. Source Systems of a DWH
 2. Core Data Warehouse (CDW)
 3. Data Mart (DM)
- CDW Archive (Data and Process)

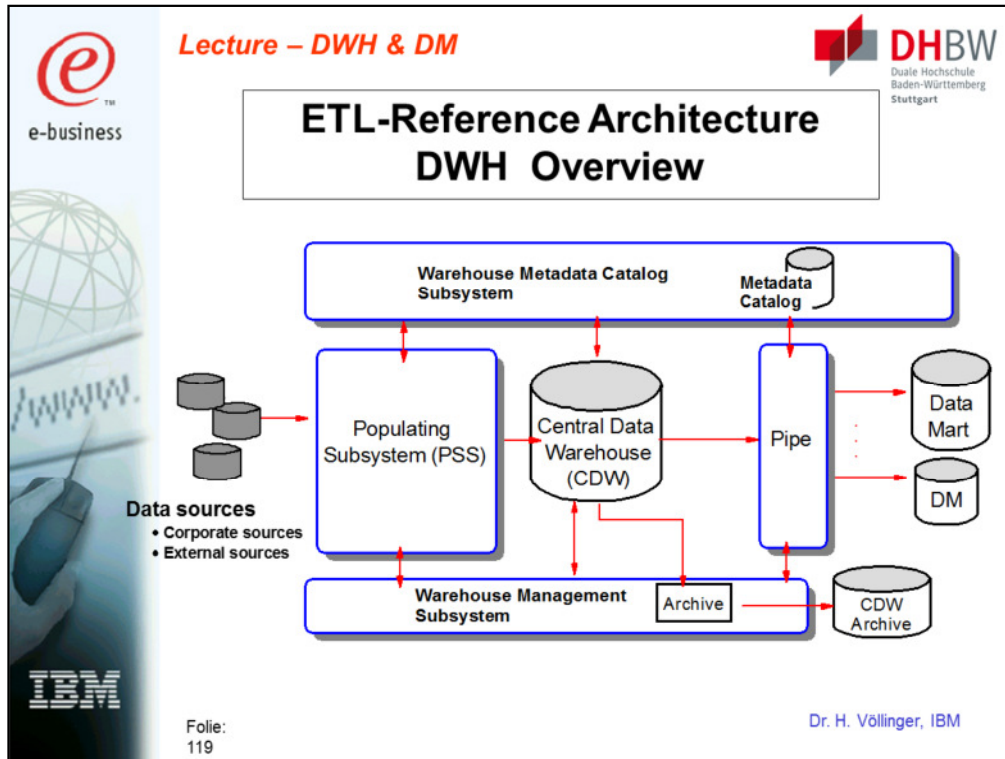


The process of data population is not only the Transformation part, but also the steps of DISCOVER and PREPARE:

DISCOVER: make an analysis of all data sources, i.e. run a data profiling process, which

will find for example inconsistencies and also bad data quality

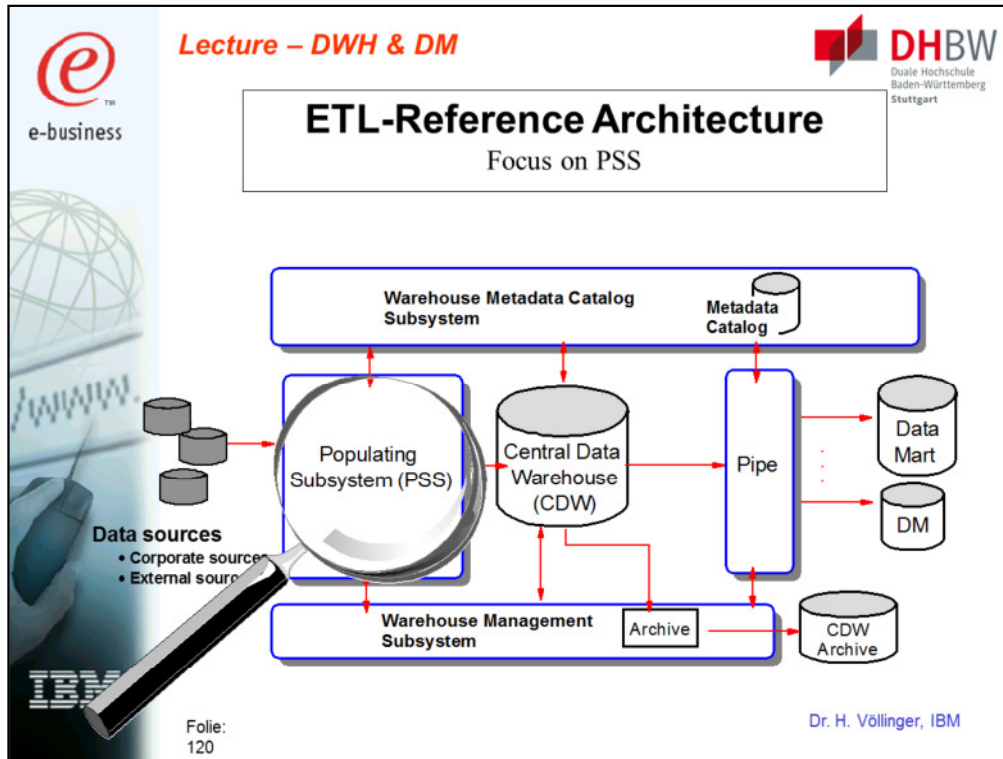
PREPARE: cleanse the data and correct the bad or corrupted data structures



The foil show the major components of a DWH Reference Architecture:

1. Warehouse Metadata Layer
2. Warehouse Management Layer
3. Populating Subsystem -PSS
4. Central Data Warehouse -CDW
5. Pipe: CDW → DM (this is the ETL process from CDW to DM)
6. Data Marts - DM
7. CDW Archive

The ETL Reference Architecture describes in detail the PSS component. So we will also show in the next foils in more detail this component.

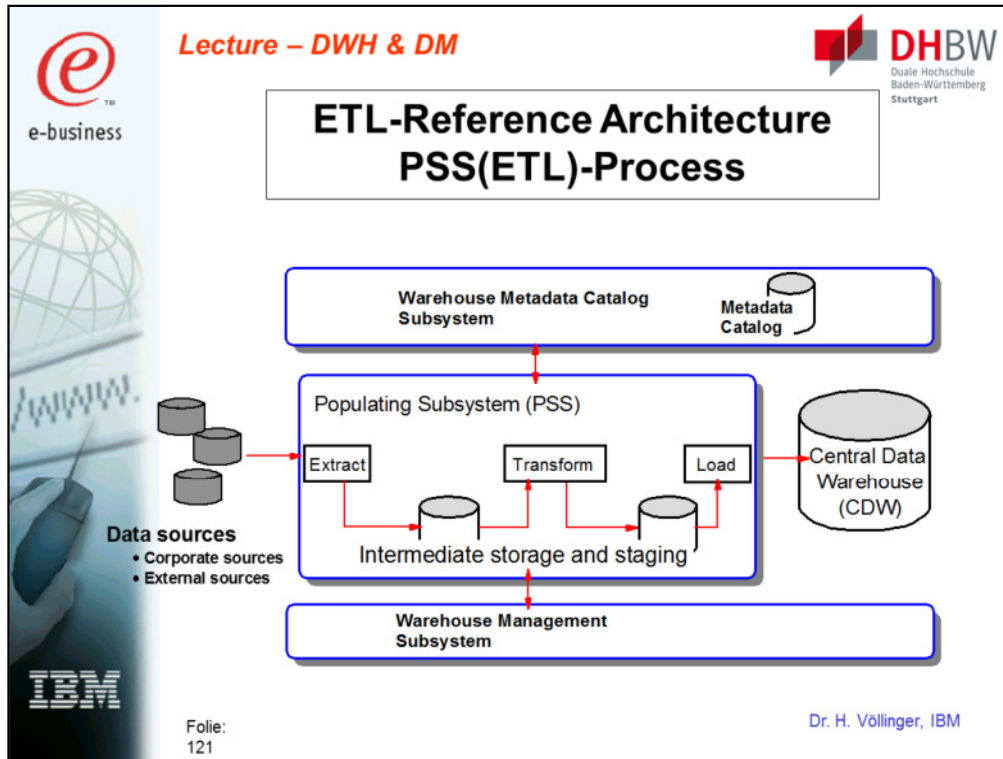


Consider now the PSS in more detail:

This is the ETL process from the source systems to the CDR.

Main Functions are:

- Extract data from sources (IMS, SAP, VSAM,
- Filter the data for valid data records
- Repair data inconsistency
- Cleanse bad data
- Run plausibility Checks
- Transform
- Load the data into DBMS (use also DB functionality)

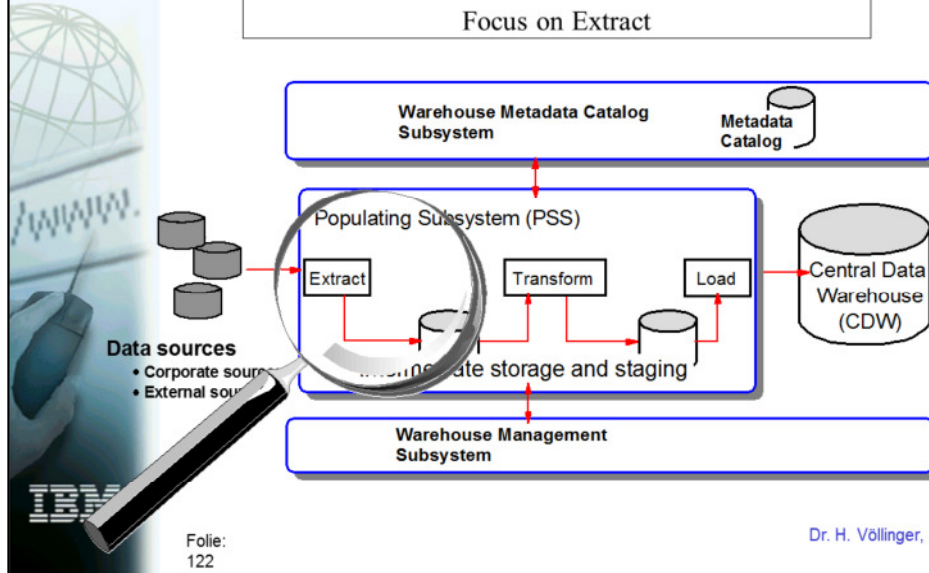


This is the first part of our reference architecture:

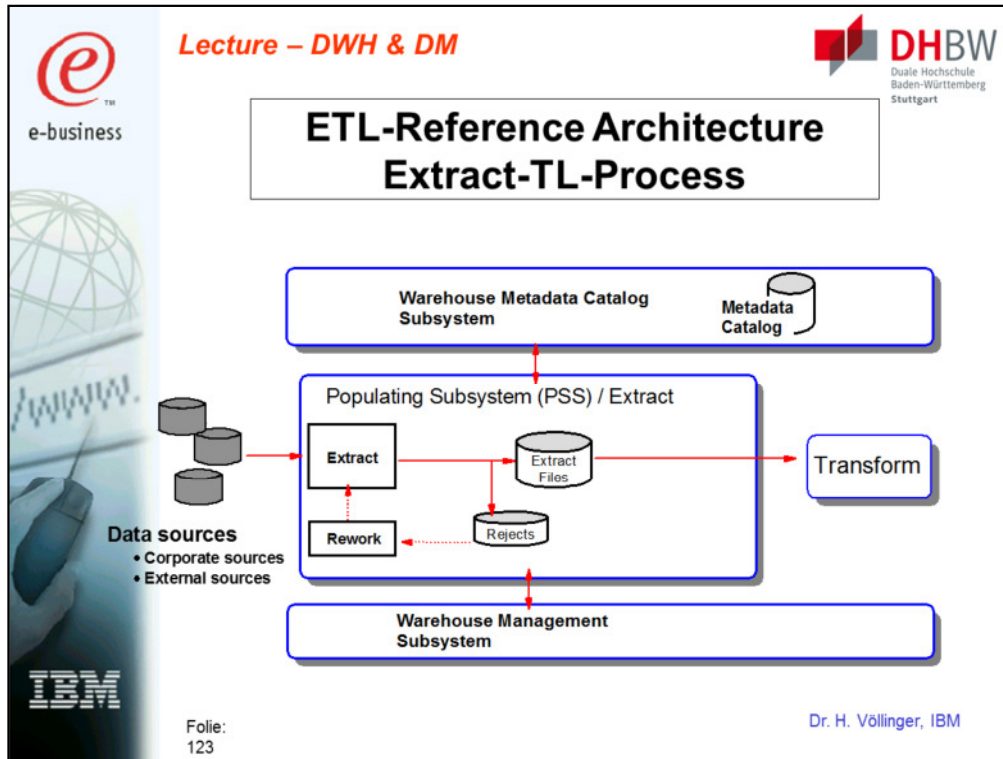
- **Extract**
 - Capture & Copy data from Operational Systems
 - Many different data structures, for example: hierarchical DBMS, Flat Files, Legacy Systems, ERP systems or other external sources
- **Transform**
 - Filter and Check (Plausibility)
 - Cleanse bad data
 - Manipulate operational data to conform with CDW standards
 - Build CDW rows
- **Load**
 - Fast load new data into CDW tables
 - Use for Load DBMS Functions if applicable

ETL-Reference Architecture ETL-Process

Focus on Extract



Data sources
• Corporate sources
• External sources



Extract services

Making all required source data available for the populating subsystem

- Full or partial snapshot of the data source
- Record of changes that occurred on the data source

Building and managing Extract files (EF)

- Static Capture (Snapshot) EF
- Incremental Capture EF

Filtering of inappropriate data and records (rejects)

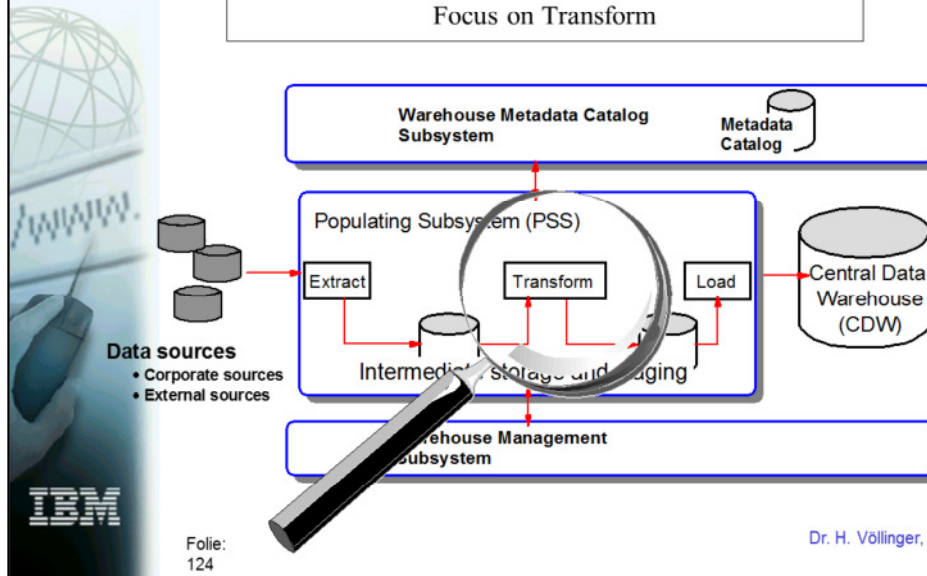
- Minimal
- Provide rework capabilities

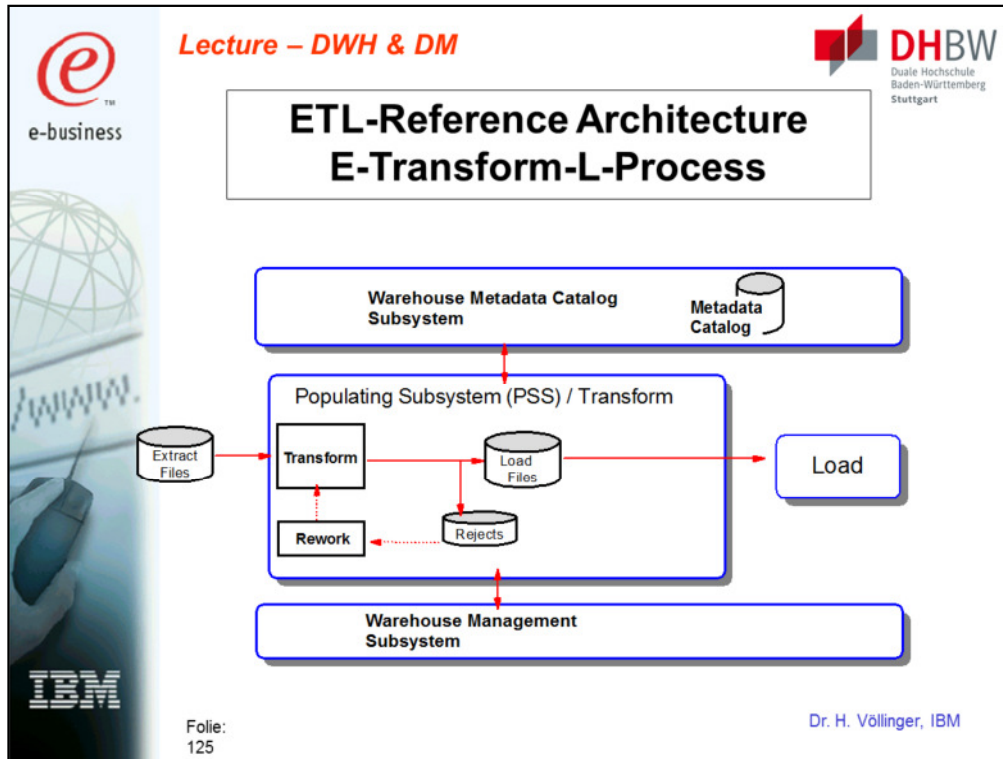
Provide Metadata

- Statistics gathering

ETL-Reference Architecture ETL-Process

Focus on Transform





Transform Component

Transform Control Manager

- Recognize "common format" extract files and call appropriate transformation services

Transformation services

- Structural transformations
- Content transformations
- Functional transformations

Building and managing Load Files

Filtering of inappropriate records (rejects)

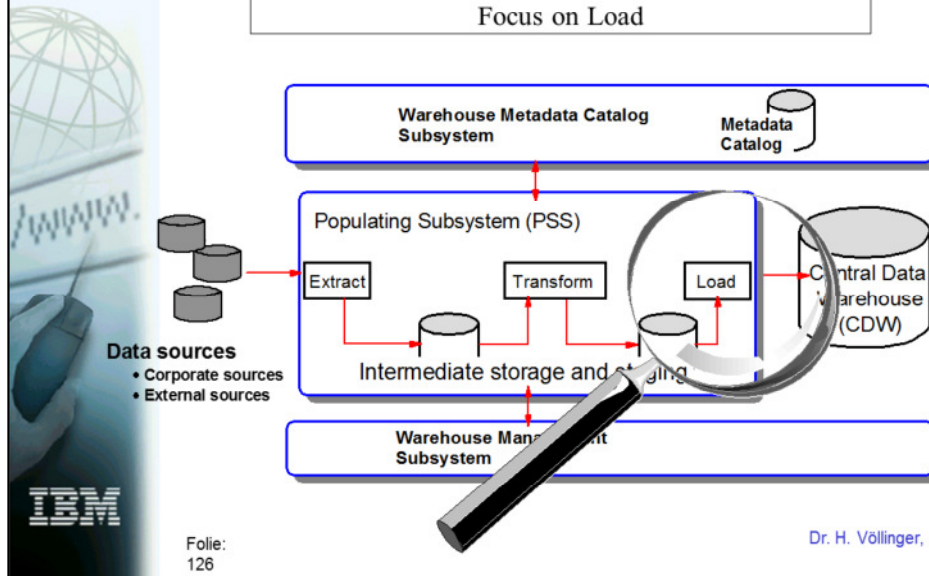
- Provide rework capabilities

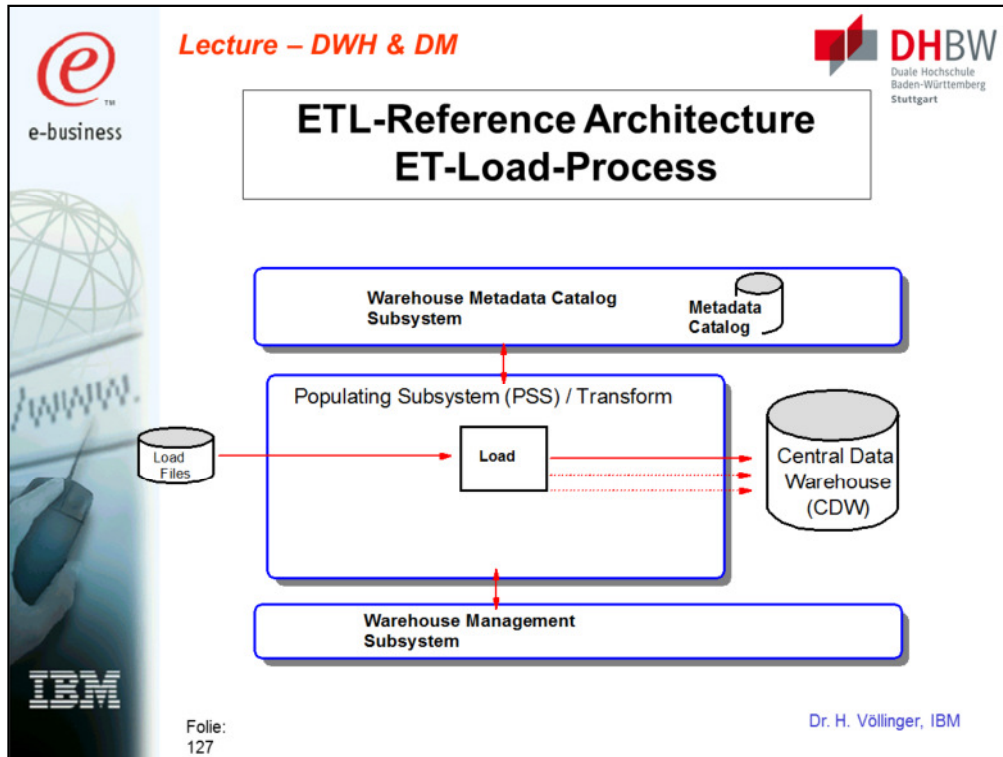
Provide Metadata

- Statistics gathering

ETL-Reference Architecture ETL-Process

Focus on Load





Load Component

Load Control Manager

- Recognize load files and call appropriate load services
- Recognize status of the apply files and decide appropriate actions (start, restart, prune, ...)
- Manage cascading effects of the load
- Manage locking/unlocking of the CDW
- Split to parallel load processes

Load services

- Read load files and load them into relevant parts of the CDW

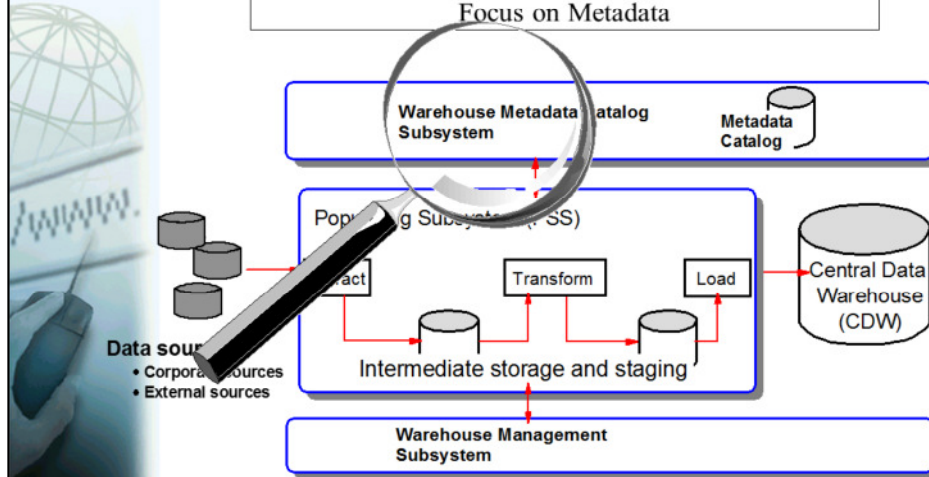
Provide Metadata

- Statistics gathering

ETL-Reference Architecture

ETL-Process

Focus on Metadata






Data sources
• Corporate sources
• External sources



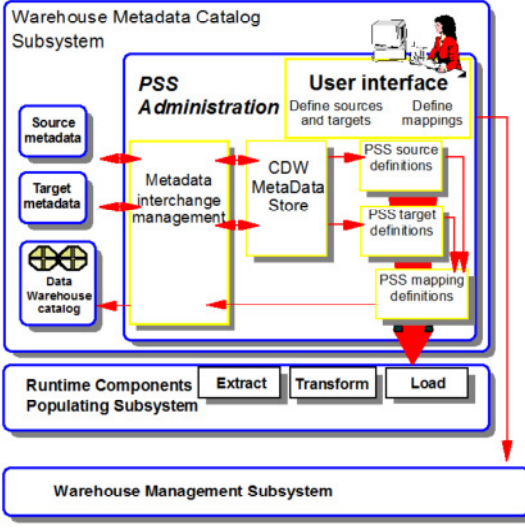
Lecture – DWH & DM

ETL-Reference Architecture
Warehouse Metadata Subsystem



- **Metadata sources**
 - Data modelling tools
 - Database catalogs
 - Record definitions in programs
 - Populating tools
- **Metadata Outputs**
 - PSS runtime statistics
 - Data Warehouse catalog
 - Process management
- **Issues**
 - Metadata access
 - Metadata synchronization
 - Metadata Interchange
 - CDW Metadata store
 - Today's tools provide little or no support



The diagram illustrates the Warehouse Metadata Subsystem architecture. It is divided into three main layers:

- Warehouse Metadata Catalog Subsystem:** This top layer contains:
 - Source metadata**, **Target metadata**, and **Data Warehouse catalog** on the left.
 - PSS Administration** (Metadata interchange management) in the center.
 - CDW MetaData Store** in the middle.
 - User interface** (Define sources and targets, Define mappings) on the right.
 - PSS source definitions**, **PSS target definitions**, and **PSS mapping definitions** on the far right.
- Runtime Components Populating Subsystem:** This middle layer includes **Extract**, **Transform**, and **Load** components.
- Warehouse Management Subsystem:** This bottom layer provides the foundational management for the entire system.

 Arrows indicate the flow of metadata and data between these components.

Dr. H. Völlinger, IBM

Folie: 129

This foil show the metadata layer, which is key to understand a good DWH strategy.

We have to distinguish between **technical** metadata and **business** metadata

It's important that the knowledge in the metadata is **published to the end users**, such that this knowledge is not lost.

Key for a good metadata strategy is the usage of **metadata standards**.

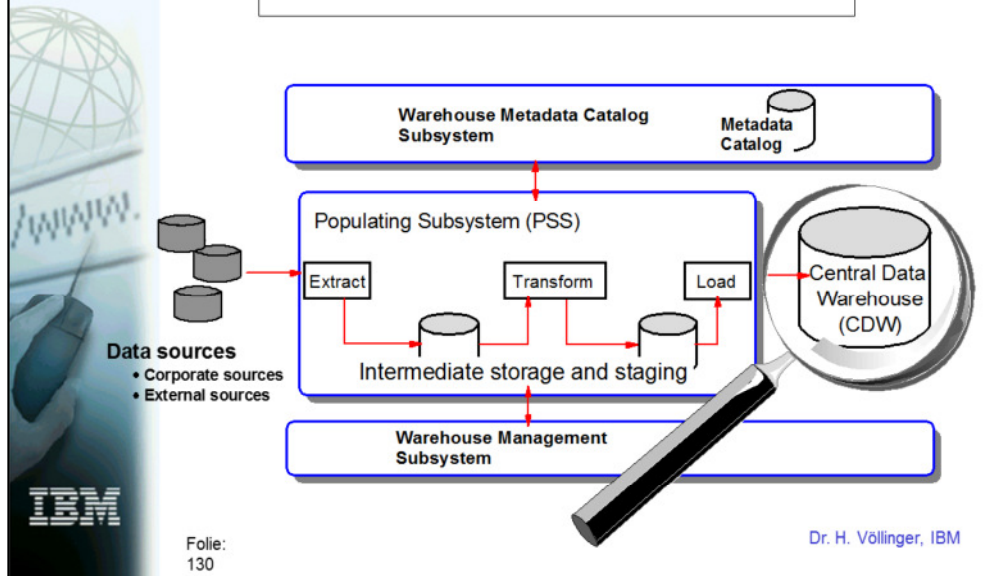
Remark: Just realize that common metadata standards are currently non-existent.

The OMG promoted a 'tagging' standard, **CWMI**, that most vendors, including IBM, have verbally endorsed but few other than IBM have actually implemented.

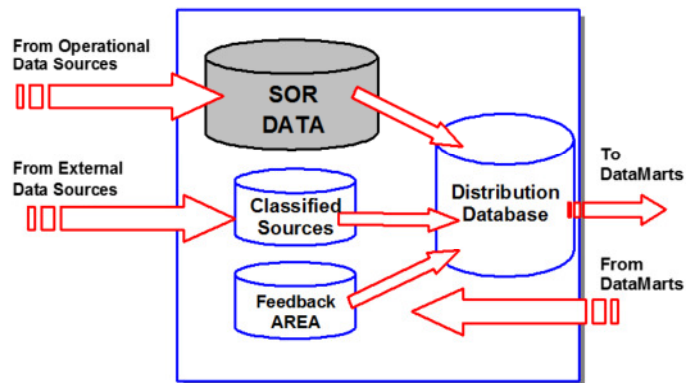
Microsoft recently has proposed a metadata standard, **OIM**, but this is viewed as proprietary due to the dependence on Microsoft SQL Server.

ETL-Reference Architecture

Focus on Central Data Warehouse



ETL-Reference Architecture CDW Data Feeds



Foils shows the main data layers of a CDW:

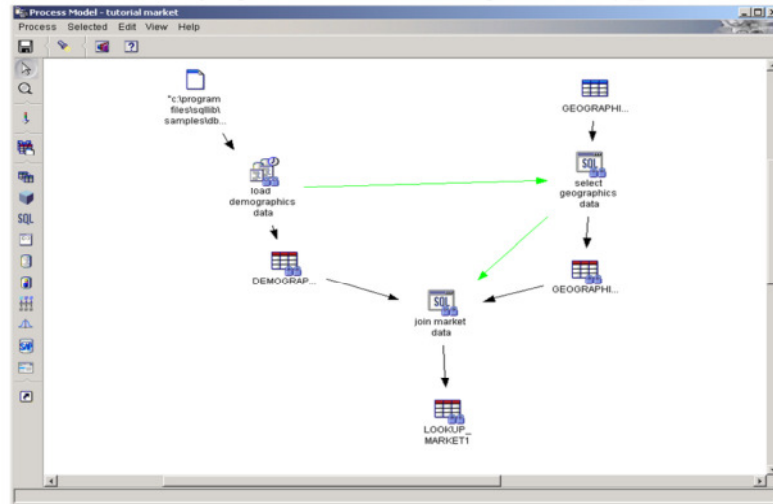
- SoR Data, which get the data out of the operational systems
- External data sources
- Feedback area
- Distribution Database (sometimes also called 'output layer')



Demo (20 Minutes) DB2 Warehouse Manager

Exercise1 to ETL - DB2 Warehouse Manager (Part1)

Define the underlying SQL for the loading of Lookup_Market table:



Folie:
133

Dr. H. Völlinger, IBM

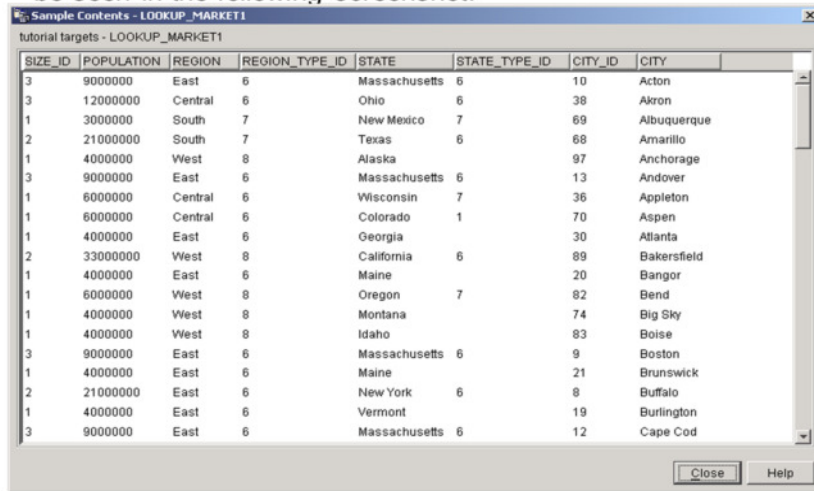
Join the demographics & geographics tables

Remark: DB2 Warehouse Manager uses SQL SELECT statement to extract data from warehouse sources and generates an INSERT statement to insert the data into the warehouse target table.

For more details see handouts

Exercise1 to ETL - DB2 Warehouse Manager (Part2)

The structure of the target table Lookup_Market1 table can be seen in the following screenshot:



SIZE_ID	POPULATION	REGION	REGION_TYPE_ID	STATE	STATE_TYPE_ID	CITY_ID	CITY
3	9000000	East	6	Massachusetts	6	10	Acton
3	12000000	Central	6	Ohio	6	38	Akron
1	3000000	South	7	New Mexico	7	69	Albuquerque
2	21000000	South	7	Texas	6	68	Amarillo
1	4000000	West	8	Alaska		97	Anchorage
3	9000000	East	6	Massachusetts	6	13	Andover
1	6000000	Central	6	Wisconsin	7	36	Appleton
1	6000000	Central	6	Colorado	1	70	Aspen
1	4000000	East	6	Georgia		30	Atlanta
2	33000000	West	8	California	6	89	Bakersfield
1	4000000	East	6	Maine		20	Bangor
1	6000000	West	8	Oregon	7	82	Bend
1	4000000	West	8	Montana		74	Big Sky
1	4000000	West	8	Idaho		83	Boise
3	9000000	East	6	Massachusetts	6	9	Boston
1	4000000	East	6	Maine		21	Brunswick
2	21000000	East	6	New York	6	8	Buffalo
1	4000000	East	6	Vermont		19	Burlington
3	9000000	East	6	Massachusetts	6	12	Cape Cod

Folie:
134

Dr. H. Völlinger, IBM

Conditions: see handouts

Exercise2 to ETL – Tools for the first two of the „Three Steps of Data Population“

In the lecture to this chapter we have seen 3 steps -“Discover” + “Prepare” + “Transform”- for a successful data population strategy.

Please present for the first two steps examples of two tools. Show details like functionality, price/costs, special features, strong features, weak points, etc.

You can use the examples of the lecture or show new tools, which you found in the internet or you know from your current business....

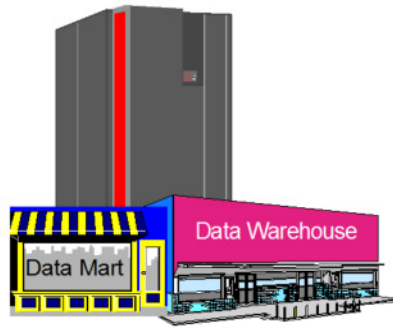
1. **DISCOVER:** Evoke-AXIO (now Informatica), Talend - Open Studio, IBM Inform. Sever (ISS) – ProfileStage, or ????
2. **PREPARE:** HarteHanks-Trillium, Vality-Integrity, IBM Inform. Server (IIS) – QualityStage, or ??????



Conditions: see handouts

Lesson 7

ETL Techniques & ETL Tools

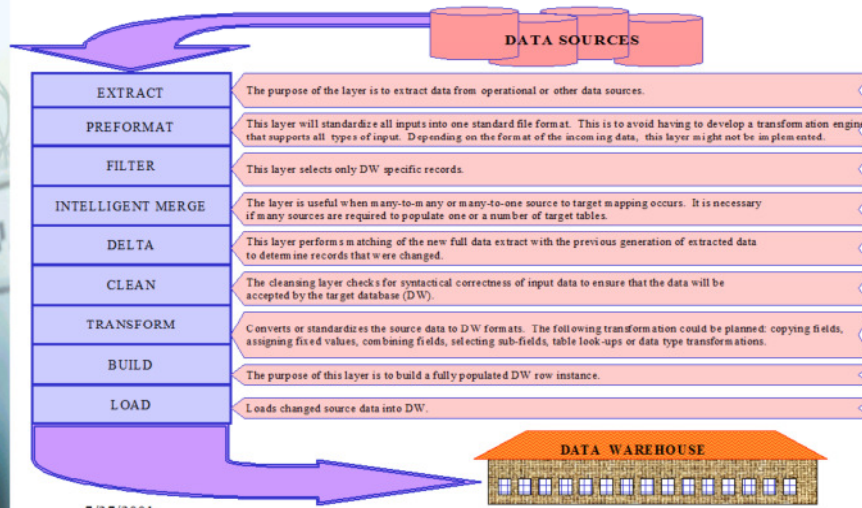


5 Highlights to ETL Techniques

1. ETL Process Layer Concept
2. Framework / Control of Processes
3. Scalability & Parallel Processing
4. Integration of ETL and DB
5. Special ETL Techniques



Generic ETL Process Layers



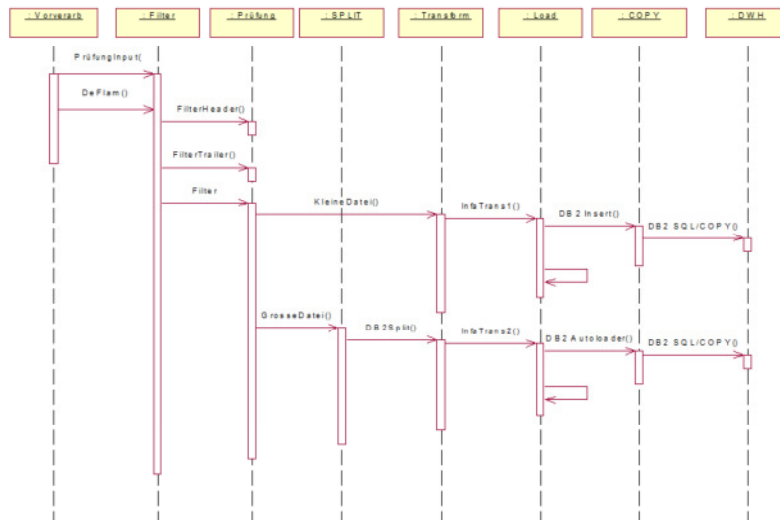
7/27/2001

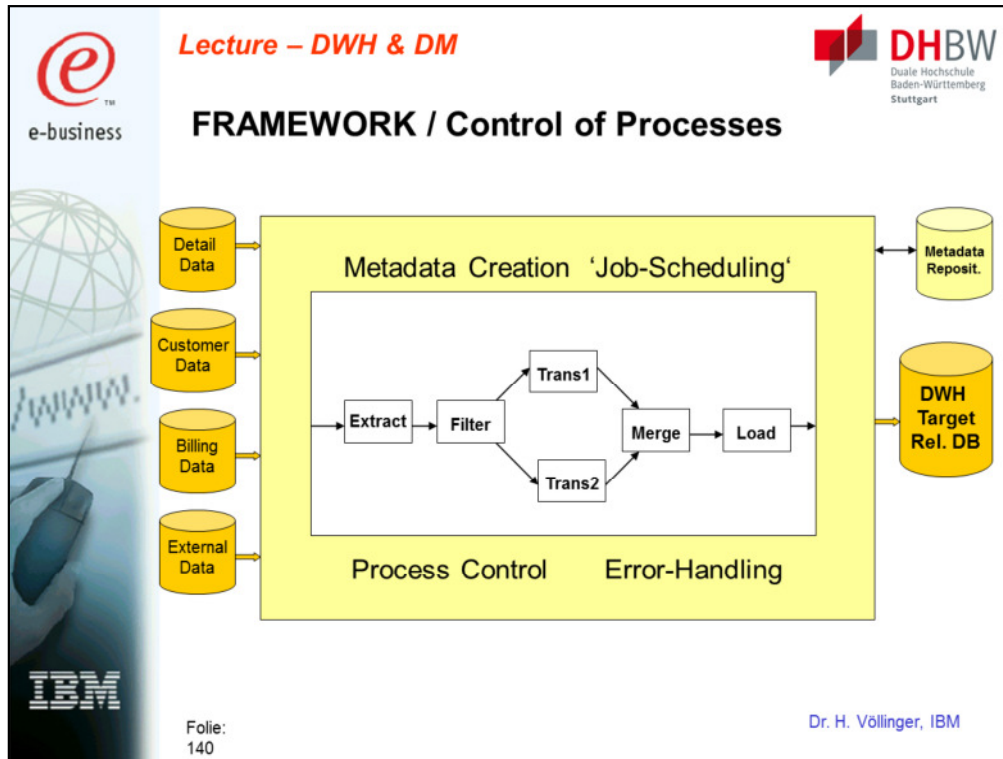


Folie:
138

Dr. H. Völlinger, IBM

ETL Layer Konzept (Beispiel)





The whole ETL process will be controlled by the Framework including the following functions:

- Initialization of ETL process (check completeness of data sources)
- Start/Stop of the ETL process
- Job-Scheduling (using triggers)
- Definition and control of parallel ETL processes
- Control all processes (through process parameters out of metadata)
- Error handling and protocol errors as metadata
- Creation of technical metadata (protocol and ETL statistics)
- Control the interface to the database system (DB2)
- Delete temporary files

Lecture – DWH & DM

DHBW
Duale Hochschule
Baden-Württemberg
Stuttgart

Scalability & Parallel Processing

Dr. H. Völlinger, IBM

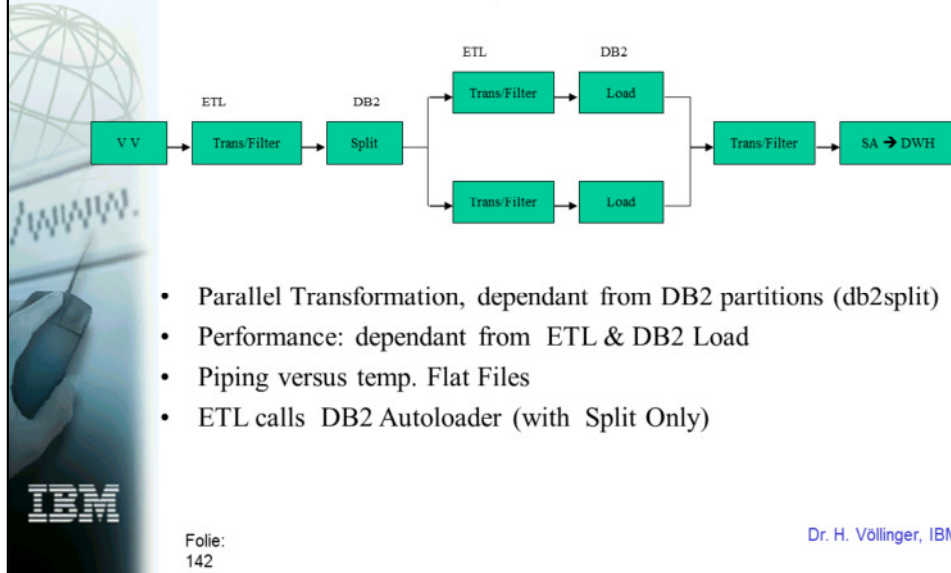
Folie:
141

The foil shows an example for parallel processing:

We see the following main steps:

1. Split of the input data in our example with DB2 Autoloader (SPLIT ONLY)
2. Distribution on parallel jobs (dependent on number of CPU's)
3. Parallel Transformation
4. Parallel Load into DB2 partitions

Integration of ETL & Database (Variante N)



- Parallel Transformation, dependant from DB2 partitions (db2split)
- Performance: dependant from ETL & DB2 Load
- Piping versus temp. Flat Files
- ETL calls DB2 Autoloader (with Split Only)

We see one possible integration scenario.

Special ETL Techniques

- ‘Piping’
- Combination: ‘Piping’ & Parallel Processing
- ‘Sequential’ Design
- ‘Piped’ Design



Lecture – DWH & DM

ETL Technique – ‘Piping’

- Manage workload, optimize data flow between parallel tasks
- Reduce I/Os

STEP 1 write

STEP 2 read

TIME

STEP 1 write

Pipe

STEP 2 read

TIME

Folie: 144

Dr. H. Völlinger, IBM

In this technique we see a **pipelined execution** for running batch jobs:

Pipes increase parallelism and provide data in memory. They allow job steps to overlap and eliminate the I/Os incurred in passing data sets between them. Additionally, if a tape data set is replaced by a pipe, tape mounts can be eliminated.

The left part of our figure shows a traditional batch job, where job step 1 writes data to a sequential data set. When step 1 has completed, step 2 starts to read the data set from the beginning.

The right part of our figure shows the use of a batch pipe. Step 1 writes to a batch pipe which is written to memory and not to disk. The pipe processes a block of record at a time. Step 2 starts reading data from the pipe as soon as step 1 has written the first block of data. Step 2 does not need to wait until step 1 has finished. When step 2 has read the data from pipe, the data is removed from pipe.

Using this approach, the total elapsed time of the two jobs can be reduced and the need for temporary disk or tape storage is removed.

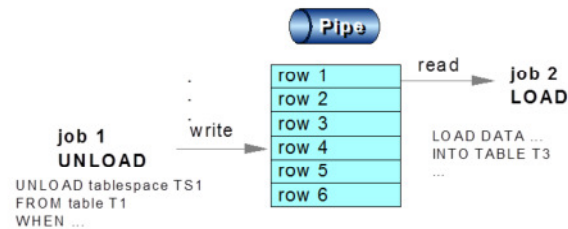
ETL Technique – ‘Piping’ Example

■ UNLOAD

- Provides fast data unload from DB2 table or image copy data set
- Samples rows with selection conditions
- Selects, order and form ats fields
- Creates a sequential output that can be used by LOAD

■ LOAD

- With SmartBatch, the LOAD job can begin processing the data in the pipe before the UNLOAD job completes.



Here we show how to **parallelize the tasks** of data extraction and loading data from a central data warehouse to a data mart.

The data is extracted from the data warehouse using the UNLOAD utility of the database. The UNLOAD writes the unloaded records to a batch pipe. As soon as one block of records has been written to the pipe database LOAD utility can start to load data into the target table in the data mart. At the same time, the unload utility continues to unload the rest of the data from the source tables.

Lecture – DWH & DM

ETL Technique – Compare Runtime

Traditional processing

Build the data with UNLOAD utility | Load the data into the tablespace

Processing using SmartBatch

Build the data with UNLOAD utility | Load the data into the tablespace

Two jobs for each partition; the load job begins before the build step has ended

Processing partitions in parallel

Build the part 1 data with the UNLOAD utility | Load the part.1 data

Build the part 2 data with the UNLOAD utility | Load the part.2 data

Two jobs for each partition

Processing partitions in parallel using SmartBatch

Build the part 1 aggregate data with DSN TIAUL | Load the part.1 data

Build the part 2 data with the UNLOAD utility | Load the part.2 data

Two jobs for each partition; each load job begins before the appropriate build step has ended

Folie: 146

Dr. H. Völlinger, IBM

Compare Runtime for combined parallel and piped executions:

Based on the characteristics of the population process, parallel and piped processing can be combined.

The first scenario shows the traditional solution. The first job step creates the extraction file and the second job step loads the extracted data into the target DB2 table.

In the **second scenario**, we consider those two jobs as two units of work executing in parallel. One unit of work reads from the DB2 source table and writes into the pipe, while the other unit of work reads from the pipe and loads the data into the target DB2 table.

In the **third scenario**, both the source and the target table have been partitioned. Each table has two partitions. Two jobs are created and both jobs contain two job steps. The first job step extracts data from a partition of the source table to a sequential file and the second job step loads the extracted sequential file into a partition of the target table. The first job does this for the first partition and the second job does this for the second partition.

In the **fourth scenario**, we manage the jobs as four units of work:

1. Unload data from partition 1 to pipe 1
2. Read data from pipe 1 and load into 1. partition of target DB
3. Unload data from part. 2 to pipe 2
4. Read data from pipe 2 and load into 2. partition of target DB

Lecture – DWH & DM

ETL Technique – ‘Sequential Design’

The diagram illustrates the ETL Sequential Design process. It starts with an input 'Claims For: Year, State' which is split into multiple parallel paths. Each path goes through a 'Key Assign.' step, then a 'Cim Key Assign.' step (which reads from 'Cim Tbl' and 'Link Tbl' tables), and finally a 'Transform' step. The transformed data is then loaded into multiple parallel 'Load' utilities, each receiving 'Records For: Claim Type, Year / Qtr, State, Table Type'. Below the diagram is a 'CPU Utilization Approximation' bar chart showing low utilization during the sequential 'Split', 'Key Assign.', and 'Transform' phases, and a significant increase in utilization during the parallel 'Load' phase.

Dr. H. Völlinger, IBM

Folie: 147

We show here an example of an **initial load with a sequential design**:

In a sequential design the different phases of the process must complete before the next phase can start. The split must, for example, complete before the key assignment can start.

In this example of an insurance company, the input file is split by claim type, which is then updated to reflect the generated identifier, such as claim ID or individual ID.

This data is sent to the transformation process, which splits the data for the load utilities. The load utilities are run in parallel, one for each partition. This process is repeated for every state for every year, a total of several hundred times.

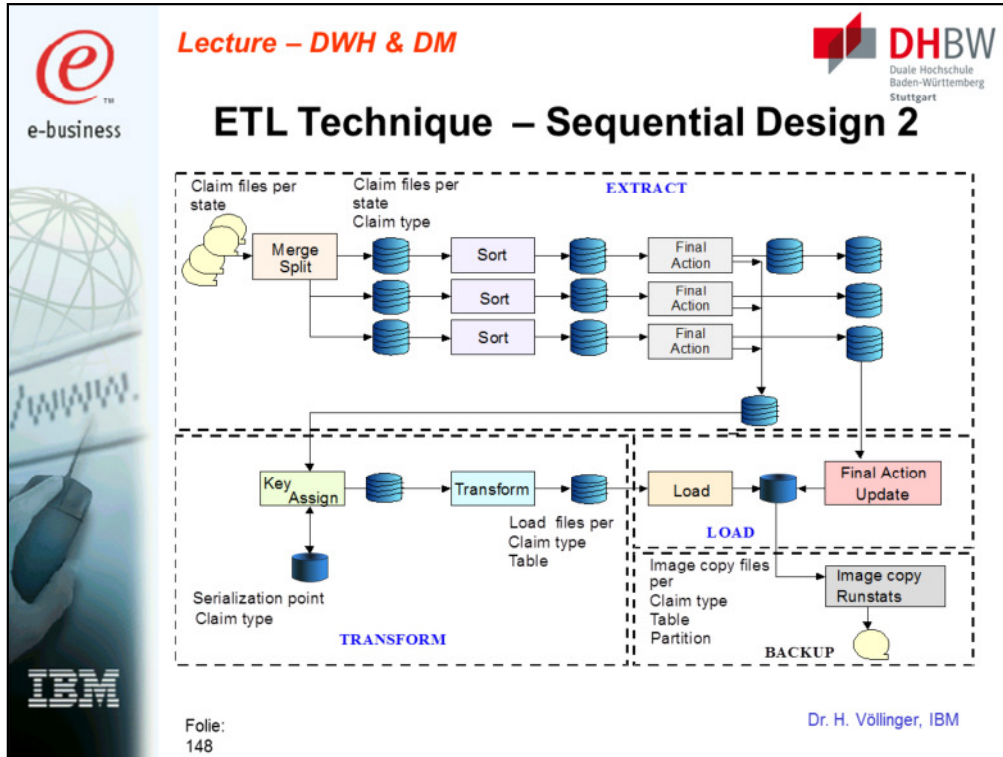
As can be seen at the bottom of the chart, the CPU utilization is very low until the load utilities start running in parallel.

The advantage of the sequential design is that it is easy to implement and uses a traditional approach, which most developers are familiar with.

This approach can be used for initial load of small to medium-sized data warehouses and refresh of data where a batch window is not an issue.

In a VLDB environment with large amounts of data a large number of scratch tapes are needed: one set for the split, one set for the key assignment, and one set for the transform. A large number of reads/writes for each record makes the elapsed time very long. If the data is in the multi-terabyte range, this process requires several days to complete through a single controller. Because of the number of input cartridges that are needed for this solution when loading a VLDB, media failure may be an issue. If we assume that 0.1 percent of the input cartridges fail on average, every 1000th cartridge fails.

This solution has a low degree of parallelism (only the loads are done in parallel), hence the inefficient usage of CPU and DASD. To cope with multi-terabyte data volumes, the solution must be able to run in parallel and the number of disk and tape I/Os must be kept to a minimum.



Here we show a design similar to the one in the previous figure. This design is based on a customer proof of concept conducted by the IBM Teraplex center in Poughkeepsie. The objectives for the test are to be able to run monthly refreshes within a weekend window.

As indicated here, DASD is used for storing the data, rather than tape. The reason for using this approach is to allow I/O parallelism using SMS data striping. This design is divided into four groups:

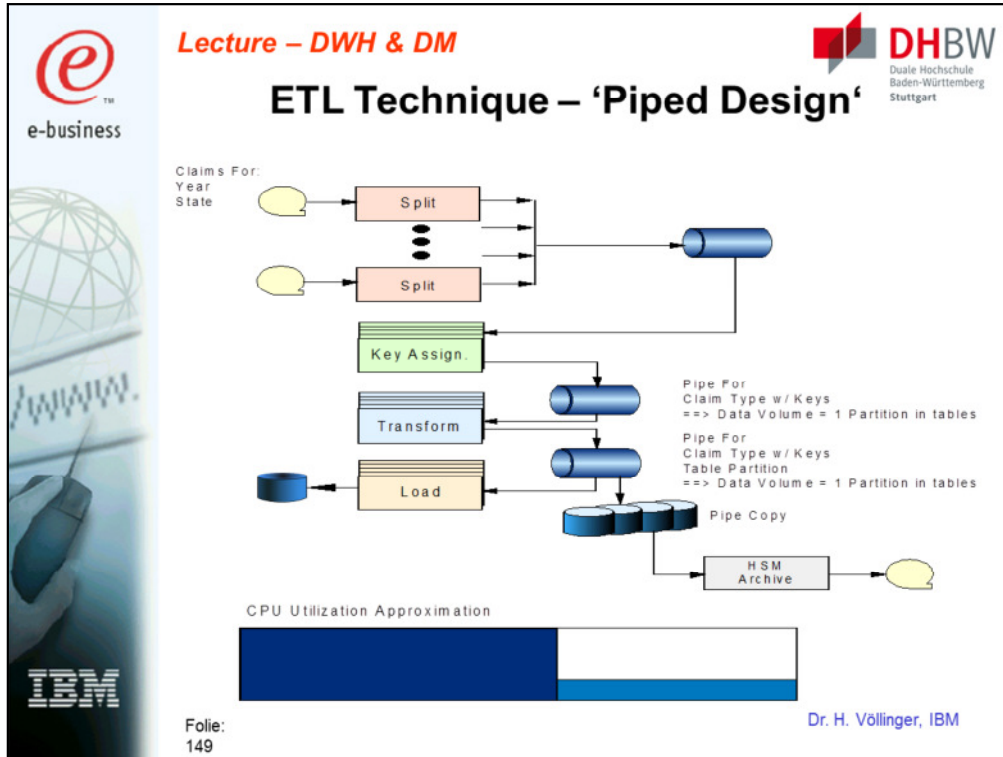
- Extract
- Transform
- Load
- Backup

The extract, transform and backup can be executed while the database is online and available for end user queries, removing these processes from the critical path.

It is only during the load that the database is not available for end user queries. Also, Runstats and Copy can be executed outside the critical path. This leaves the copy pending flag on the table space. As long as the data is accessed in a read-only mode, this does not cause any problem.

Executing Runstats outside the maintenance window may have severe performance impacts when accessing new partitions. Whether Runstats should be included in the maintenance must be decided on a case-by-case basis. The decision should be based on the query activity and loading strategy. If this data is load replaced into a partition with similar data volumes, Runstats can probably be left outside the maintenance part (or even ignored).

Separating the maintenance part from the rest of the processing not only improves end-user availability, it also makes it possible to plan when to execute the non-critical paths based on available resources.



The **piped design** shown here uses pipes to avoid externalizing temporary data sets throughout the process, thus avoiding I/O system-related delays (tape controller, tape mounts, and so forth) and potential I/O failures.

Data from the transform pipes are read by both the DB2 Load utility and the archive process.

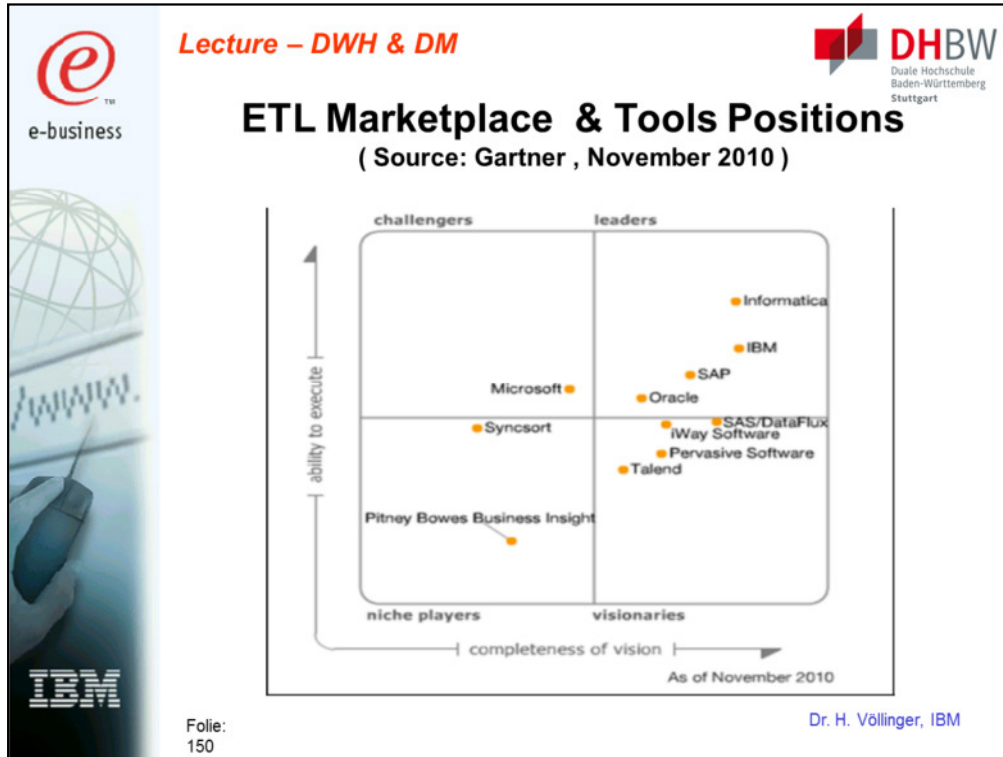
Since reading from a pipe is by nature asynchronous, these two processes do not need to wait for each other. That is, the Load utility can finish before the archive process has finished making DB2 tables available for end users.

This design enables parallelism—multiple input data sets are processed simultaneously.

Using batch pipes, data can also be written and read from the pipe at the same time, reducing elapsed time.

The throughput in this design is much higher than in the sequential design. The number of disk and tape I/Os has been drastically reduced and replaced by memory accesses. The data is processed in parallel throughout all phases.

Since batch pipes concurrently process multiple processes, restart is more complicated than in sequential processing. If no actions are taken, a piped solution must be restarted from the beginning. If restartability is an important issue, the piped design must include points where data is externalized to disk and where a restart can be performed.



For more details see:



http://www.virtualtechtour.com/assets/GARTNER_DI_MQ_2010_magic_quadrant_for_data_inte_207435.pdf

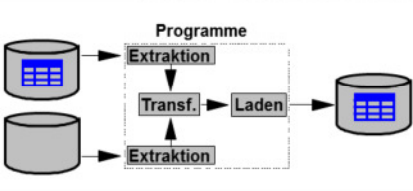
Gartner has defined multiple classes of functional capabilities that vendors of data integration tools must possess to deliver optimal value to organizations in support of a full range of data integration scenarios:

- Connectivity/adaptor capabilities (data source and target support).
- Data delivery capabilities.
- Data transformation capabilities.
- Metadata and data modeling capabilities.
- Design and development environment capabilities.
- Data governance capabilities (data quality, profiling and mining).
- Deployment options and runtime platform capabilities.
- Operations and administration capabilities.
- Architecture and integration.
- Service-enablement capabilities.

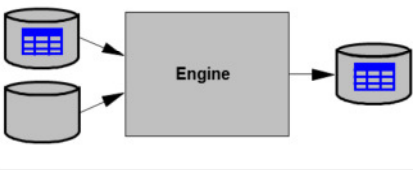
Lecture – DWH & DM

The 3 ETL Tool Architectures

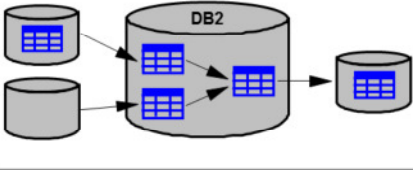





- ETL Code Generator
 - 3GL Programs (C, COBOL, ...)
 - Load Balancing on several CPUs & Systems
 - Debugging possible
- f.ex. ETI*EXTRACT, DataStage/390



- ETL Engine
 - Transformation on UNIX / NT System
 - Central ETL Management
- f. ex. Informatica, DataStage



- ETL with Database Utilities
 - SQL, Stored Procedures, UDF's
 - Database Scalability
 - DB-Transaction Security
- f.ex. DB2 Warehouse Manager
Oracle Warehouse Builder (OWB)

Folie: 151

Dr. H. Völlinger, IBM

SQL Engines, like DB2 Warehouse Manager first extracts data from the source systems and then loads this data into DB2 tables. It then uses DB2 database technology (SQL, stored procedures, user defined functions) to transform the data inside DB2. With this approach all DB2 features like parallelization, scalability and transaction security are automatically available for the ETL process with no additional effort.

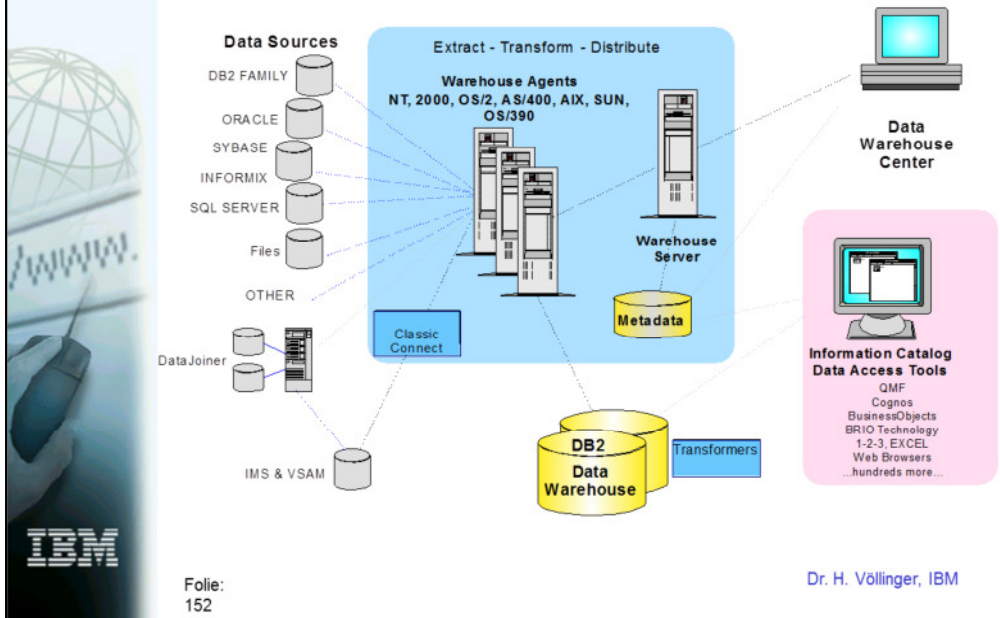
ETL Engines provide a centralized ETL management where all needed functionality is integrated transparently. Data is extracted from the source systems, transformed on a UNIX or Windows NT server and then loaded into the target system. An example for this approach is Ascential DataStage or Informatica.

ETL Code Generators generate 3GL programs (COBOL, C, ABAP, ...) for the extraction, transformation and loading of data. These programs can be run on different servers to achieve load balancing.

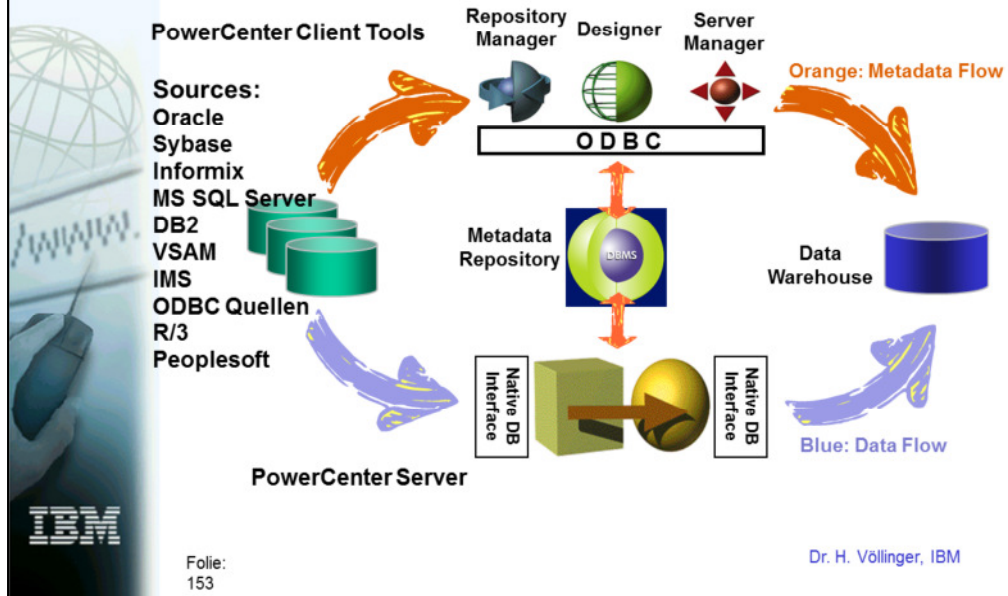
Debugging and performance tuning is easy because the program source code is available. DataStage/390 is an example for this architecture.

Each of these architectures has its advantages in certain environments.

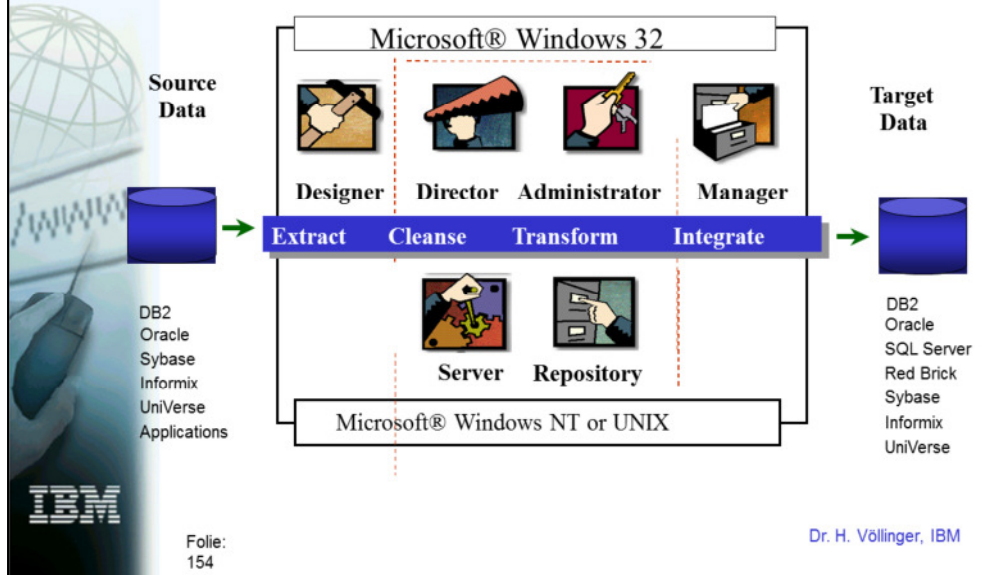
ETL Tool – DB2 Warehouse Manager



ETL Tool – Informatica PowerCenter



ETL Tool – Ascential DS XE/6000 (Ardent)



Exercise to Lesson 7: ETL Tool Evaluation

Show the Highlights and build a Strengthens/Weakness Diagram for the following three ETL Tools.

Use the information from the internet:

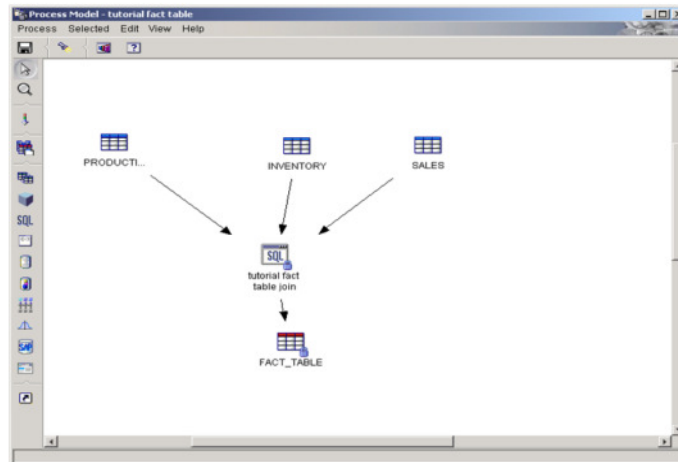
1. Informatica – PowerCenter --→ www.informatica.com
2. Ascential – DataStage (Infosphere Inform. Serv. - DataStage)
---→ www.ascentialsoftware.com oder www.ibm.com
3. IBM – DB2 Warehouse Manager
---→ www-3.ibm.com/software/data/db2/datawarehouse/

Show the three tools in competition to each other



Optional: Exercise to ETL - DB2 Warehouse Manager (Part1)

Define the underlying SQL for the following loading of the Fact_Table from the 3 tables: Production_Costs, Inventory & Sales:



Folie:
156

Dr. H. Völlinger, IBM

Join information from **SALES**, **PRODUCTION_COSTS**, and **INVENTORY** tables and writes the result in a new table: **FACT_TABLE**.

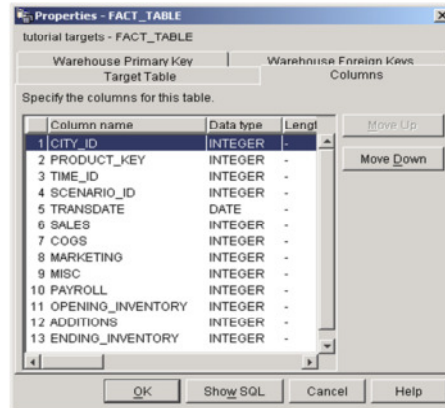
Compute calculated column to derive ending Inventory value.
Generates CITY_ID, TIME_ID, and SCENARIO_ID

Remark: DB2 Warehouse Manager uses SQL Select statement to extract data from warehouse sources and generates an INSERT statement to insert the data into the warehouse target table.

For more details see handouts

Optional: Exercise to ETL - DB2 Warehouse Manager (Part2)

The structure of the target table Fact_Table can be seen in the following screenshot:

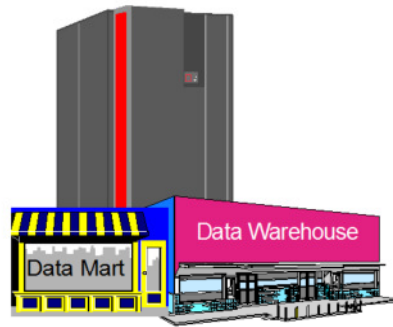


Column name	Data type	Length
1 CITY_ID	INTEGER	-
2 PRODUCT_KEY	INTEGER	-
3 TIME_ID	INTEGER	-
4 SCENARIO_ID	INTEGER	-
5 TRANSDATE	DATE	-
6 SALES	INTEGER	-
7 COGS	INTEGER	-
8 MARKETING	INTEGER	-
9 MISC	INTEGER	-
10 PAYROLL	INTEGER	-
11 OPENING_INVENTORY	INTEGER	-
12 ADDITIONS	INTEGER	-
13 ENDING_INVENTORY	INTEGER	-



Conditions: see handouts

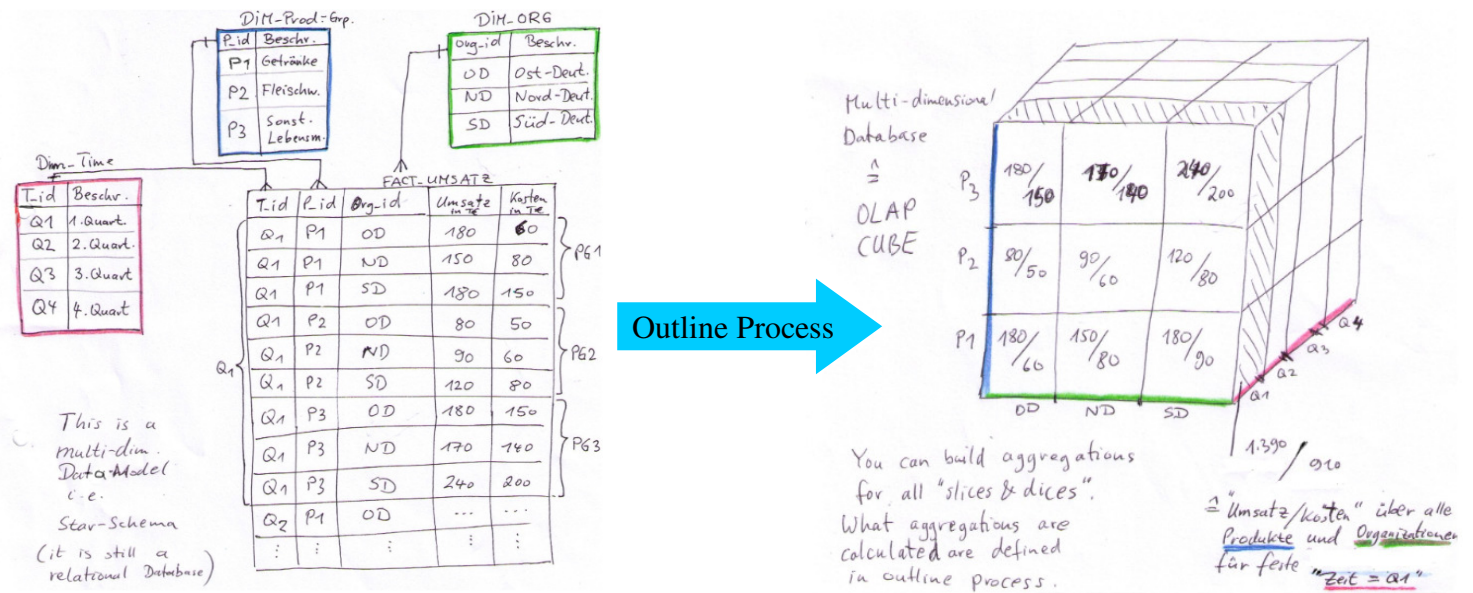
Lesson 8 Introduction to OLAP




Folie: 158


Dr. H. Völlinger, IBM

Motivation: In chapter 5 we learned about Multi Dim. Data Models (MDDM). Out of such a data model a OLAP Cube is build. This process is called “OUTLINE” process. See the following pictures:






e-business



IBM

Lecture – DWH & DM

What is OLAP



DHBW
Duale Hochschule
Baden-Württemberg
Stuttgart

- Stands for **OnLine Analytical Processing**
- A fast way of querying and reporting on data held in a data warehouse
- Business data is stored in a number of dimensions, so that the data can easily be analysed from many different viewpoints
 - Data is modelled to the business
 - The reshaped data is held in a special format
 - The data is viewed across, down and through the various dimensions
- Answers business questions and follow-on questions
 - How is that broken down?
 - Is that the same pattern every year?
 - Can we look at that another way?

Folie:
159

Dr. H. Völlinger, IBM

Compare also what we learned before above Multi Dimensional Data Modeling (MDDM).

How is broken down ? ---→ detail of hierarchy (drill down, drill up)

...every year ---→ time dimension

...in another way ? ---→ slice and dice

OLAP tools take you a step beyond query and reporting tools. Via OLAP tools, data is represented using a multidimensional model rather than the more traditional tabular data model.

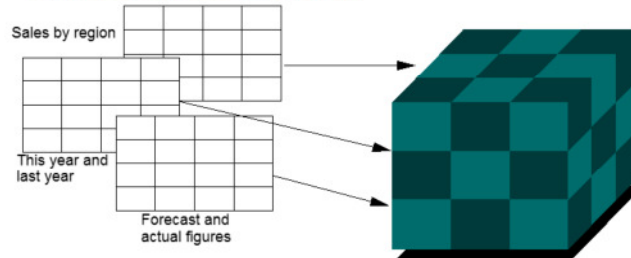
The traditional model defines a database schema that focuses on modelling a process of function, and the information is viewed as a set of transactions, each which occurred at some single point in time.

The multidimensional model usually defines a star schema, viewing data not as a single event but rather as the cumulative effect of events over some period of time, such as weeks, then months, then years.

With OLAP tools, the user generally views the data in grids or crosstabs that can be pivoted to offer different perspectives on the data. OLAP also enables interactive querying of the data. For example, a user can look at information at one aggregation (such as a sales region) and then drill down to more detail information, such as sales by state, then city, then store.

What is Multidimensionality

- The process of converting flat, row and column oriented data into a virtual cube
 - Business operations are modelled by organizing data in a multi-dimensional array
 - Each *dimension* describes an important point of view for business data (e.g., time, product, location, etc.)
 - Dimensions are composed of members, which describe the instances of the dimensions (eg. 4Q97, skateboards, Barcelona etc.)
- Supports simultaneous alternate views of sets of data
 - Time, accounts, products, markets etc.



Folie:
160

Dr. H. Völlinger, IBM

Multidimensionality is turning data from various sources, relational tables or flat files, into a structure where data is grouped into separate, heterogeneous dimensions.

This is often referred to as a cube. In reality cubes are three dimensional of course, but the term is used for a database holding data from more than three dimensions.

Dimensions are made up of **members**. In the example here, a dimension might be products and skateboards may be a member of that dimension.

Multidimensional Database

- A database specially designed to handle the organisation of data in multiple dimensions!
- Holds data cells in blocks that can be quickly built into a virtual cube depending on the query it is satisfying
- Optimised to handle large amounts of numeric data
 - Index of descriptive names held separately from block of numeric data
 - Often holds totals pre-calculated as well as base data
 - Not intended for textual data such as customer address lists



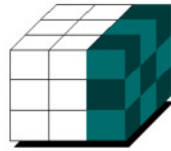
Some products store the data in a purpose-built multidimensional database, others hold it in a different form, such as relational tables, and construct the dimensions when the data is requested.

A multidimensional database stores all the “cube” data in a way that it can be regarded as multidimensional, not as separate files or tables. The data itself is numeric, the names of the dimensions and members are stored separately.

Another feature of an multidimensional database is that totals are often pre-calculated and stored with the data, not calculated when the data is fetched.

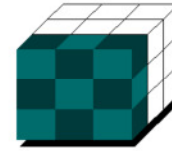
Multidimensional Views

Different selections give different ways of looking at the data

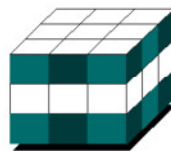


February for all products

		Jan		Feb	
Video		Jan	Feb	Actual	Budget
TV					
		Actual	Budget	Actual	Budget
Sales	Paris				
	Moscow				
	London				
	Total				
Costs	Paris				
	Moscow				
	London				
	Total				

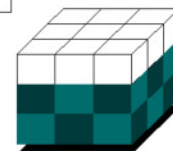


All TV information



Paris Sales and Costs

Viewing 5 dimensional database



All cost information

Once data is in a cube, it is easy to get all possible combinations of dimensions. These are multidimensional views of a database.

Here we see an example of a 5 dimensional database. Sales/Costs and Locations are **row** dimensions, budget/actual and time are **column** dimensions, while product (TV, Video, Audio) is a **page** dimension in this example.

They can of course be fetched in another combination of row, column and page.

Round the outside we can see the different selections can be made to show Paris only, all information for TV sales, cost information only, February budget information and so on.

Lecture – DWH & DM

Drill Down
Looks at components in greater detail down same dimension

Time	Market		Measures	Product
Year	Country		Profit	Category
Quarter	Region		Sales	Brand
Month	District		COGS	Package
Day	Town		Expense	Size

Folie: 163

Dr. H. Völlinger, IBM

A term commonly used with OLAP is **drill down**. It simply means going down to the next level of detail.

For example a drill down on year shows information by quarter, a drill down on quarter gives information for each month and so on.

The opposite is **drill up**.

Slice and Dice

Change row, column
and page dimensions



		Bud	Act	Bud	Act
1997	East				
	West				
1996	East				
	West				



		1994	1995	1996	1997
East	Food				
	Drink				
West	Food				
	Drink				

Slice and Dice is another concept you may hear when discussing OLAP.

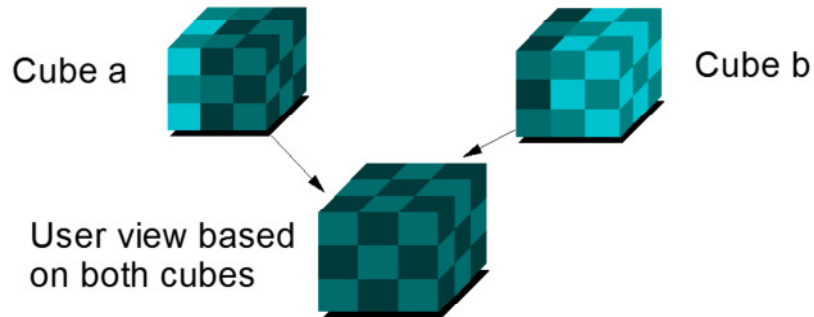
It is used to describe how multidimensional data can be shown in one way and then looked at in another way.

Effectively swapping page, column and row dimensions.

The action of moving an item from one position to another, for example moving Year displayed as a column dimension to display as a row dimension, is called **Pivot**

Multi-Cube Solutions

- Enhance Scalability
- Partition Applications for Parallel Load and Calculation
- Combine Similar or Dissimilar models in one user OLAP view



Some OLAP solutions might use more than one cube of data. For example, different product groups, countries, manufacturing plants or operating companies might have their own distinct cube. These may be integrated for reporting at a higher level. DB2 OLAP Server has facilities to partition applications across multiple cubes.

Multidimensional vs Relational

Multidimensional

- Optimised for query and report
- Restricted uses
- Fast, non-complex queries
- Data not dynamic - limited data update
- Database queries built by OLAP engine
- Cube must be rebuilt to refresh data and totals

Relational

- Optimised for transaction systems and query
- Many application areas
- Queries may be complex
- Easy to add/change data and structure
- Database queries written in SQL
- Data can be added and totalled interactively



IBM

Folie:
166

Dr. H. Völlinger, IBM

OLAP tools do not indicate how the data is actually stored. Given that, it's not surprising that there are multiple ways to store the data, including storing the data in a dedicated multidimensional database (also referred to as **MOLAP** or MDD). Examples include Arbors Software's Essbase and Oracle Express Server.


In a MOLAP environment, multi-dimensional "cubes" (often greater than 3 dimensions) are pre-calculated on the middle tier from one or more source systems and stored in a proprietary format.

End-user queries are run against the cubes and not the underlying databases/RDBMS's.

The other choice involves storing the data in relational databases and having an OLAP tool work directly against the data, referred to as relational OLAP (also referred to as **ROLAP** or RDBMS).


Examples include MicroStrategy's DSS server and related products, Informix's Informix-MetaCube, Information Advantage's Decision Suite, and Platinum Technologies' Plantinum InfoBeacon. (Some also include Red Brick's Warehouse in this category, but it isn't really an OLAP tool. Rather, it is a relational database optimised for performing the types of operations that ROLAP tools need.)

In a ROLAP environment, SQL queries are sent to a database on an RDBMS, the result is returned to a middle-tier server for final cross-tabbing and formatting, and then sent to the client that initiated the query.



e-business


Lecture – DWH & DM



DHBW
Duale Hochschule
Baden-Württemberg
Stuttgart

MOLAP vs ROLAP

Similarities	Differences
<ul style="list-style-type: none"> ● Both work with numeric data, not textual ● Output results the same ● Both can provide drill down and slice & dice ● Both provide information to end users 	<ul style="list-style-type: none"> ● Totals usually already calculated in MD OLAP ● MD cube must be recalculated ● ROLAP joins data tables for each query ● MD cube size limited by architecture, ROLAP size limited by database



Folie: 167

Dr. H. Völlinger, IBM

The foil shows the most important similarities and differences between MOLAP and ROLAP concept:

Usually, a scaleable, parallel database is used for the large, atomic. Organizationally structured data warehouse, and subsets or summarized data from the warehouse are extracted and replicated to proprietary MDDs.

Because MDD vendors have enabled drill-through features, when a user reaches the limit of what is actually stored in the MDD and seeks more detail data, he/she can drill through to the detail stored in the enterprise database. However, the drill through functionality usually requires creating views for every possible query.

As relational database vendors incorporate sophisticated analytical multidimensional features into their core database technology, the resulting capacity for higher performance saleability and parallelism will enable more sophisticated analysis.

Proprietary database and non-integrated relational OLAP query tool vendors will find it difficult to compete with this integrated ROLAP solution.

Both storage methods have strengths and weaknesses -- the weaknesses, however, are being rapidly addressed by the respective vendors.

Currently, data warehouses are predominantly built using RDBMSs. If you have a warehouse built on a relational database and you want to perform OLAP analysis against it, ROLAP is a natural fit.

This isn't to say that MDDs can't be a part of your data warehouse solution. It's just that MDDs aren't currently well-suited for large volumes of data (10-50GB is fine, but anything over 50GB is stretching their capabilities).

If your really want the functionality benefits that come with MDD, consider sub-setting the data into smaller MDD-based data marts.

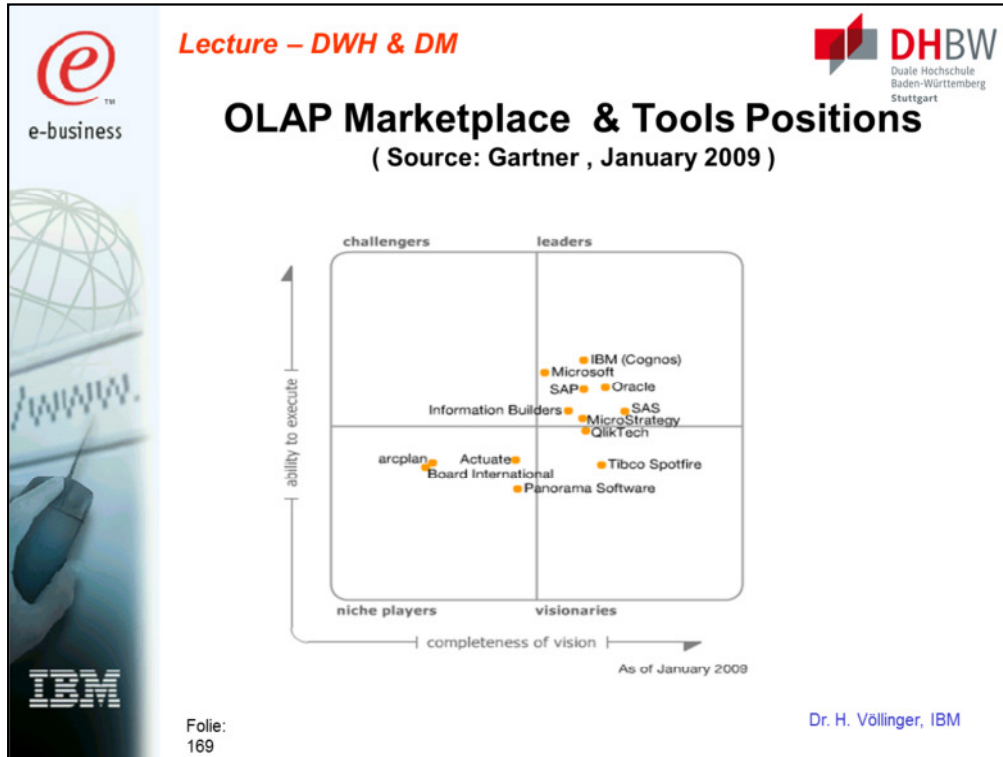
Benefits of MOLAP

- Makes many different analyses without constructing separate queries
 - All possible queries on the multidimensional data can be created by OLAP engine
 - Fast response to changing data requests
- Quick to deploy
 - Simple to report using spreadsheet or graphical tool
 - Many end user requirements satisfied once cube is built without building individual reports
- Quick to use
 - "Speed of thought" response
 - No contention from long-running queries
- Common Informational Database
 - Same information on server available to many users
 - Doesn't impact transaction systems



Although there is an initial effort in building a cube and calculating any stored totals, the main benefit is that once it is made, everything else is easy and quick.

All possible reports can on any selection and any combination without Developing new reports extractions, the queries are very fast and it is easy to add new users.



Market Overview

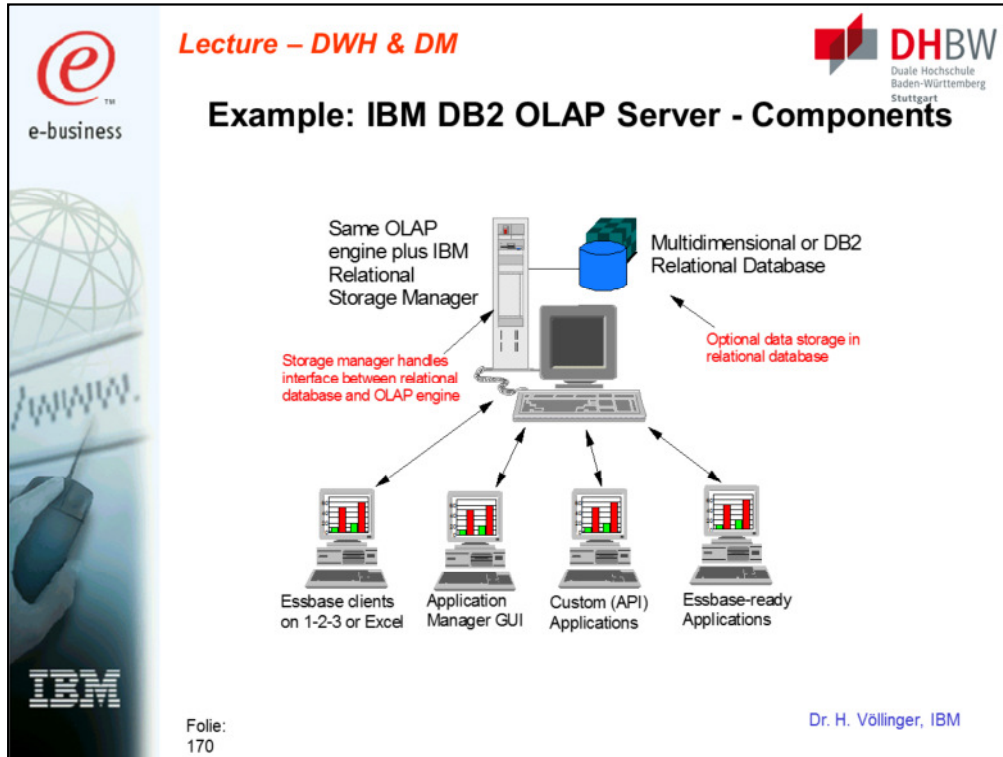
In January 2008, SAP completed its acquisition of Business Objects and IBM completed its acquisition of Cognos. The acquired firms were two of the largest in the market and had, arguably, been the defining suppliers in the space. Their absorption into larger entities (along with Hyperion into Oracle six months earlier) seemingly marked the end of BI platforms as a predominantly stand-alone, best-of-breed, buying decision. In 2008, BI platform investment decisions became tethered more closely to strategic sourcing and stack-led factors, and more influenced by organizational relationships with application and infrastructure vendors than before.

Conversely, however, based on the research conducted for this report and interactions with Gartner customers over the year, there clearly remains a demand for independent BI platforms.

To understand this duality, it is necessary to consider a number of factors at play that are driving the BI platform today:

See for more details:

<http://www.bi-strategy.co.uk/downloads/Gartner%20Magic%20Quadrant%20for%20BI.pdf>



The Hyperion Essbase product family includes the following feature sets:

- **Hyperion Essbase Application Manager**

A graphical environment for developing and maintaining Hyperion Essbase applications. Tasks include building outlines and dimensions, performing data loads and calculations, and defining security access.

- **Hyperion Essbase OLAP Server**

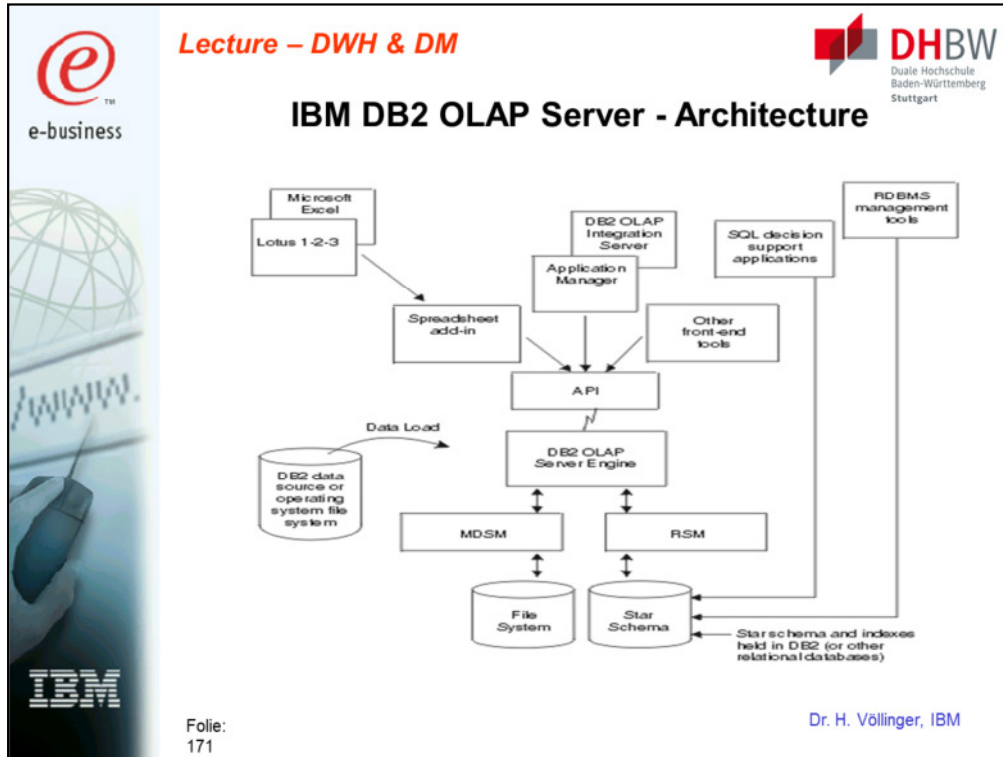
A multidimensional database for storing data with an unlimited number of dimensions, such as time, accounts, region, channel, or product. The Hyperion Essbase server manages analytical data models, data storage, calculations, and data security.

- **Hyperion Essbase Spreadsheet Add-in**

Desktop software enables analysis of the data stored in the Hyperion Essbase server. Hyperion Essbase Spreadsheet Add-in is seamlessly integrated with Microsoft Excel or Lotus 1-2-3 spreadsheets.

- **Hyperion Essbase application tools**

A suite of tools for extending Hyperion Essbase applications. These tools include Hyperion Essbase Currency Conversion, Hyperion Essbase SQL Interface, Hyperion Essbase Spreadsheet Toolkit, and Hyperion Essbase API.



The above figure shows the main components in the DB2 OLAP Server environment.

The IBM DB2 OLAP Server is an online analytical processing (OLAP) product that you can use to create a wide range of multidimensional planning, analysis, and reporting applications.

DB2 OLAP Server is based on the OLAP technology that was developed by Hyperion Solutions Corporation.

DB2 OLAP Server includes all of the capabilities of Hyperion Essbase.

In addition, it offers the option of storing multidimensional databases as sets of relational tables.

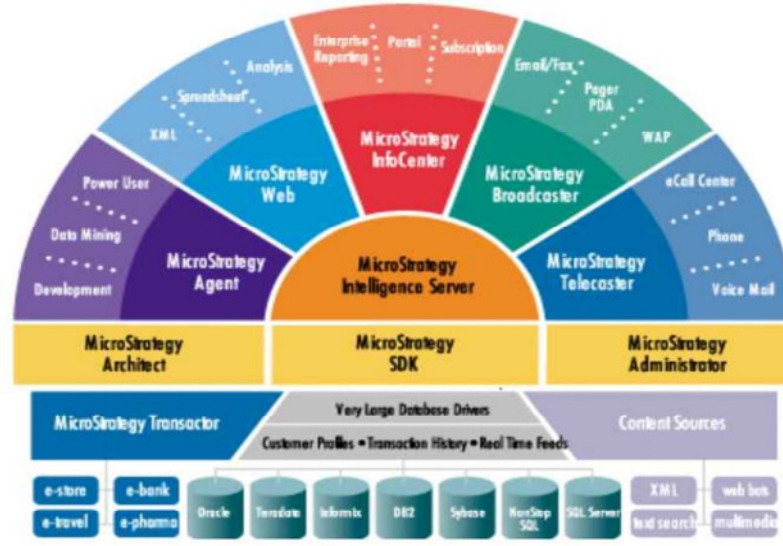
Regardless of the storage management option that you choose, you can use the Essbase Application Manager and Essbase commands to create an Essbase application and its associated databases.

You can also use over 70 Essbase-ready tools provided by independent software vendors that can access multidimensional databases transparently.



e-business

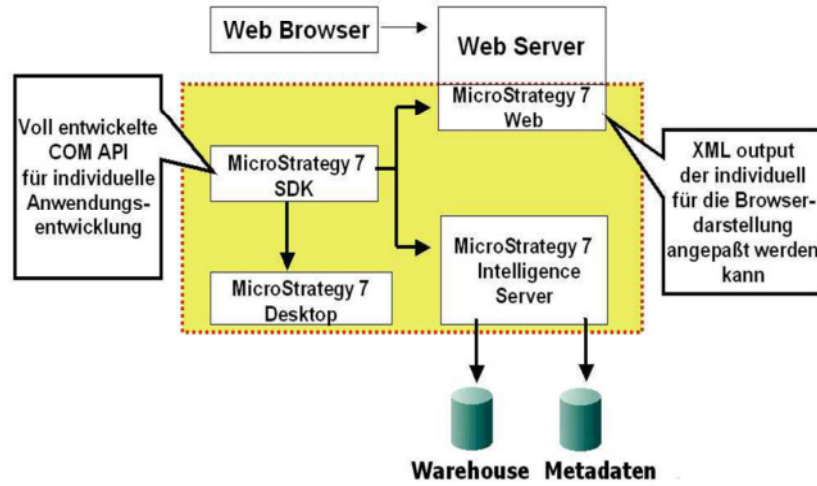
ROLAP / MicroStrategy: Components Overview



Folie:
172

Dr. H. Völlinger, IBM

ROLAP Ex.- MicroStrategy: Analytical Model

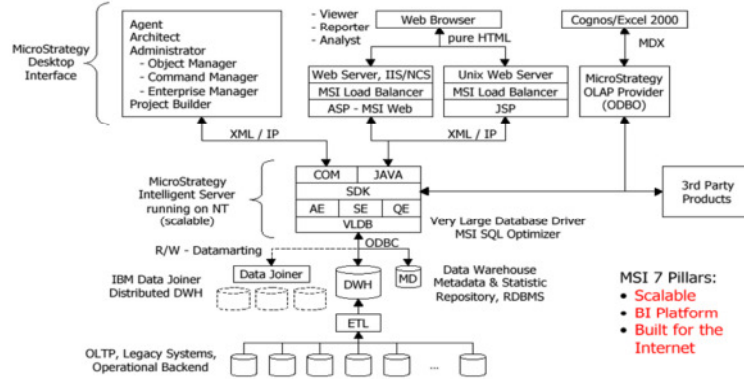




e-business

ROLAP Ex.- MicroStrategy: Big Picture

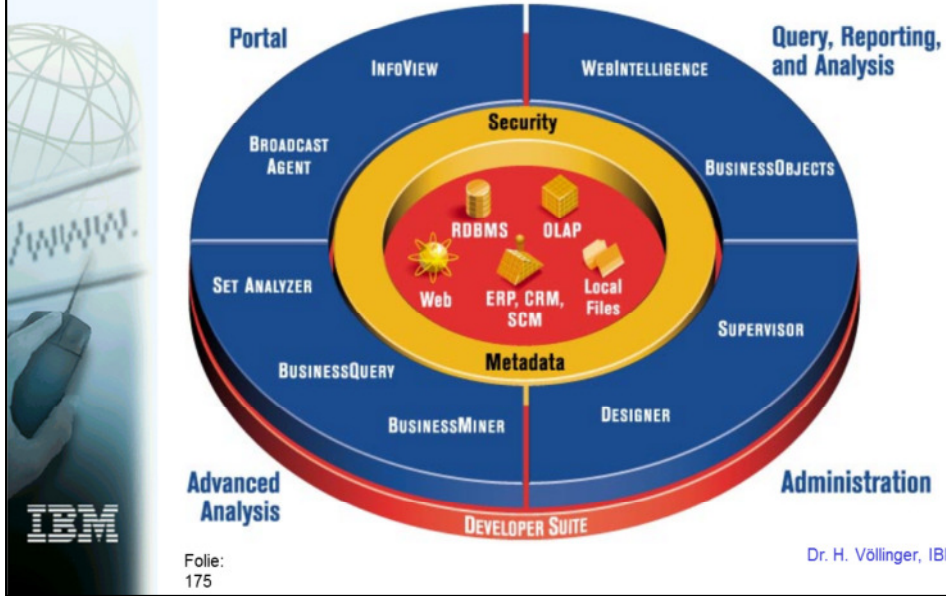
MicroStrategy Intelligence Server™



- MSI 7 Pillars:**
- Scalable
 - BI Platform
 - Built for the Internet

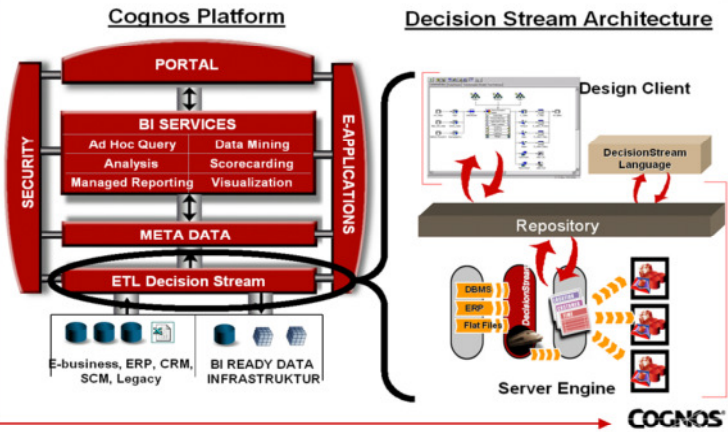


OLAP/Reporting Ex. - BusinessObject /Big Picture



OLAP/Reporting Ex. - Cognos / Big Picture

DecisionStream und die Cognos e-BI Lösung



Exercise1 to Lesson 8: MOLAP <--> ROLAP

Find and define the Benefits & Drawbacks of

- MOLAP
- ROLAP

Systems

Use the information of the lesson or use your own experience



Exercise2 to Lesson 8: OLAP/Reporting Tools

Show the Highlights and build a Strengthens/Weakness Diagram for the following three Reporting Tools.

Use the information from the internet:

1. MicroStrategy --→ www.MicroStrategy.com
2. BusinessObjects ---→ www.BusinessObjects.com
3. Cognos ---→ www.Cognos.com

Show the three tools in competition to each other



Solution to Exercise1 of Lesson 8: MOLAP

Benefits

- Faster query performance
- Little in-flight calculation time
- Can write back to database
- More sophisticated calculations possible

Drawbacks

- Size limited by architecture of cube
- Can't access data that is not in cubes
- Housekeeping/backups limited
- Can't exploit database parallelism



Solution to Exercise1 of Lesson 8: ROLAP



Benefits

- Full use of database security/integrity
- Scalable to larger data volumes
- Data can be shared with other SQL applications
- Data and structure more dynamic


Drawbacks

- Slower queries
- Expensive to build
- Indexes and summaries not maintained automatically
- Calculations may be limited to database functions
- Less "Open" - proprietary clients



 **Lecture – DWH & DM** 

Lesson 9
Data Mining 1 - Introduction & First Methods



IBM

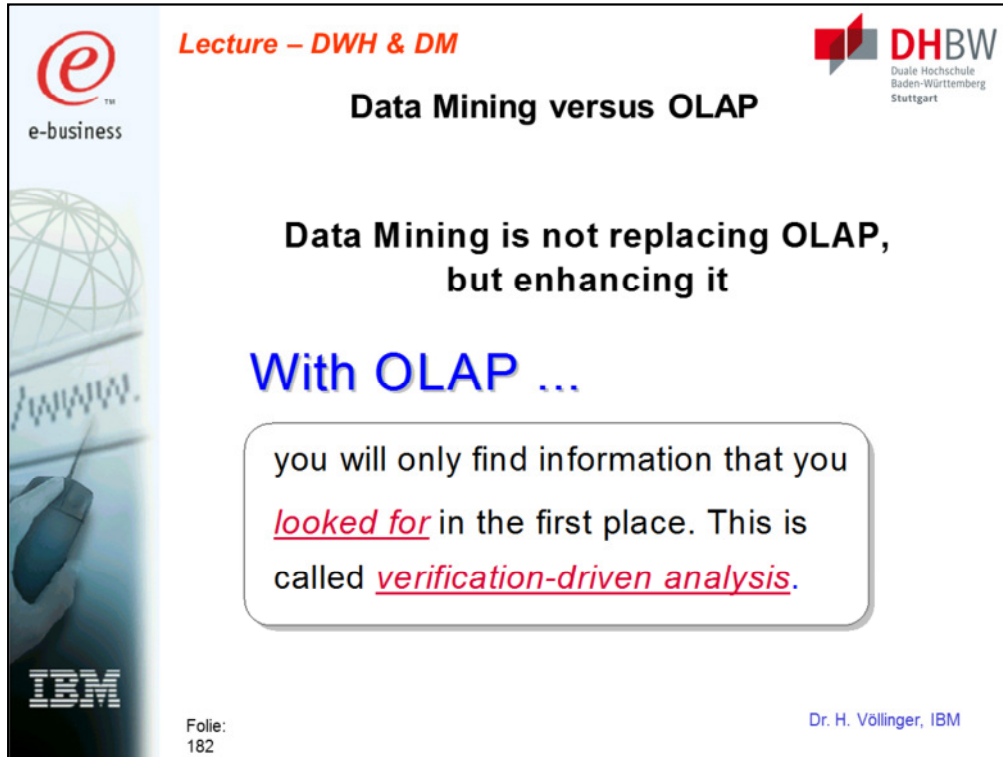
Folie:
181

Dr. H. Völlinger, IBM

The following two chapters gives an introduction to the methods and techniques of Data Mining as part of the overall architecture of a data warehouse. It explains the architectural ideas behind a Data Mining solution and shows for what industries and applications Data Mining can be used.

The best known applications, which use Data Mining techniques, are:

1. Market Basket Analysis (MBA)
2. Cross Selling
3. Customer Retention
4. Fraud Detection
5. Campaign Management

The slide features a vertical sidebar on the left with an 'e-business' logo, a globe, a mouse, and the IBM logo. The main content area includes a title 'Data Mining versus OLAP', a central statement 'Data Mining is not replacing OLAP, but enhancing it', and a blue heading 'With OLAP ...'. A rounded box contains the text 'you will only find information that you looked for in the first place. This is called verification-driven analysis.' Logos for 'Lecture - DWH & DM' and 'DHBW Duale Hochschule Baden-Württemberg Stuttgart' are at the top. Footer text includes 'Folie: 182' and 'Dr. H. Völlinger, IBM'.

OLAP and Data Mining are not replacing each other. Each of them can enhance the other technology.

While OLAP do analysis of existing data and facts, Data Mining creates new knowledge and new information for the decision process.

The slide features a vertical banner on the left with an '@' symbol, 'e-business', a globe, a mouse, and the 'IBM' logo. The top left corner has the text 'Lecture – DWH & DM'. The top right corner has the 'DHBW' logo with the text 'Duale Hochschule Baden-Württemberg Stuttgart'. The main title is 'Definition of Data Mining' in black, followed by 'Data Mining is...' in large blue font. A central box contains the definition: 'The process of extracting previously unknown, comprehensible, and actionable information from large databases and using it to make crucial business decisions.' The bottom left corner says 'Folie: 183' and the bottom right corner says 'Dr. H. Völlinger, IBM'.


Information technology has developed rapidly over the last three decades.

Many organizations store increasingly large volumes of data on their computer systems.


Useful information might be hidden in the data in the form of implicit patterns and connections that are not easy to discern using conventional data queries and statistical calculations.

Data mining is the process of discovering valid, previously unknown, and ultimately comprehensible information from large stores of data.

You can use extracted information to form a prediction or classification model, or to identify similarities between database records. The resulting information can help you make more informed decisions.




e-business



IBM

Lecture – DWH & DM



Who and where you need Data Mining

- **Telco, Insurance, Banks, Governments**
 - Fraud detection, Customer retention (Churn)
- **Retail industry**
 - Market-basket analysis
- **Manufacturing industry :**
 - Process and quality management
- **All industries (including Internet)**
 - Customer analysis and segmentation
 - Direct mailing optimization
 - Customer retention, pricing
 - Customer scoring

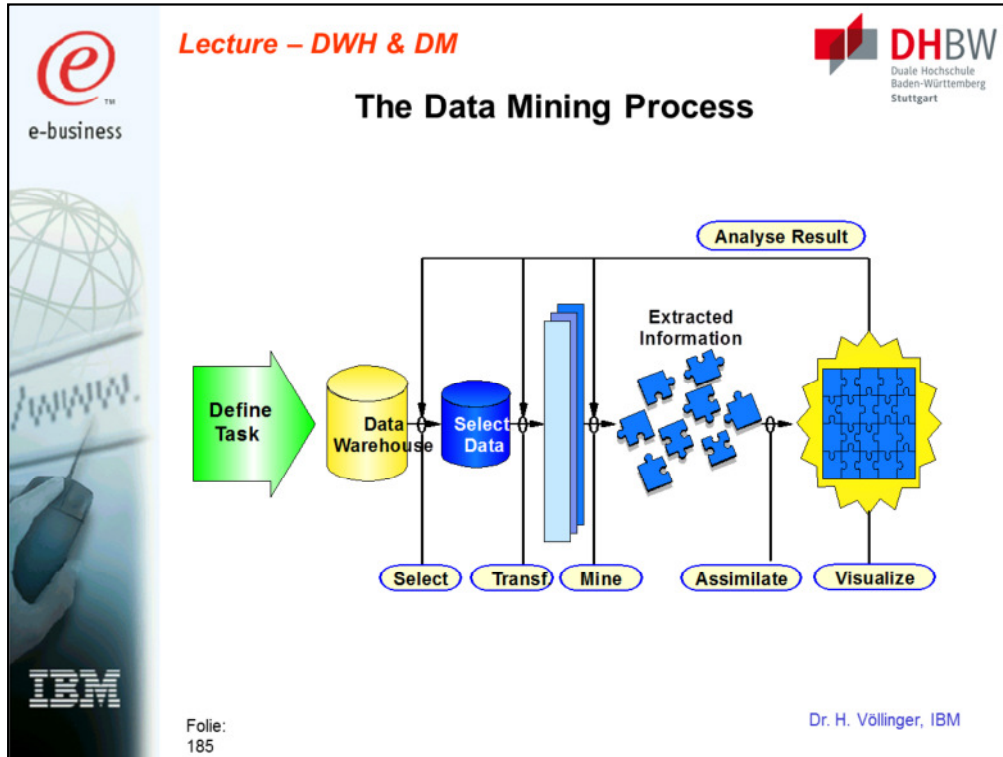
Folie:
184

Dr. H. Völlinger, IBM

Data Mining can be used in all industries.

The foil shows the most used business applications of Data Mining in the different industries.

The list is not complete, but gives a good impression of the importance of Data Mining for the process of getting information (and out of this decisions) out of raw data.



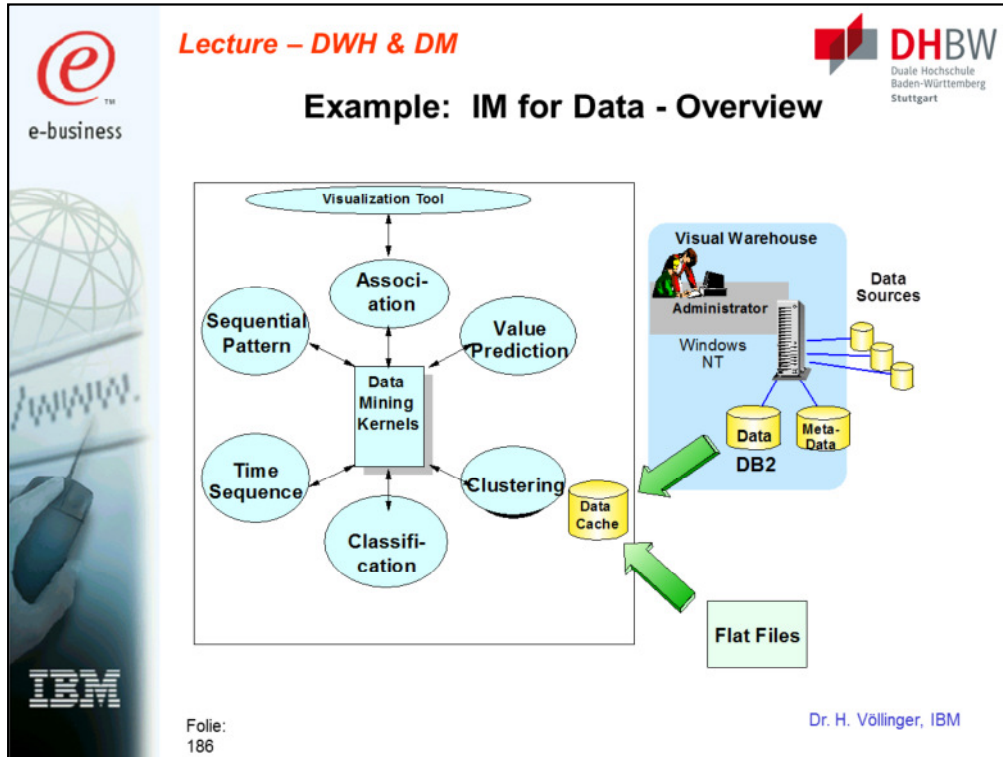
Data mining is an iterative process that typically involves:

- selecting input data,
- transforming it,
- running a mining function
- and interpreting the results.

The Data Mining tool assists you with all the steps in this process. You can apply the functions of the Data Mining tool independently, iteratively, or in combination.

Mining functions use elaborate mathematical techniques to discover hidden patterns in your data. After interpreting the results of your data-mining process, you can modify your selection of data, data processing and statistical functions, or mining parameters to improve and extend your results.

So the Data Mining process is a highly interactive and periodic process. This means in a concrete project you have to refine several times for mining models until you reach a satisfactory and valuable result.



We look here as an example on the IBM solution, which is *Intelligent Miner for Data (IM4D)*.



Intelligent Miner's mining functions use innovative techniques to efficiently discover previously unknown patterns in your business data.

The following mining functions are available:


- Associations mining function
- Clustering mining functions
- Sequential Patterns mining function
- Time Sequences mining function
- Classification mining functions
- Prediction mining functions

For clustering, classification, and prediction, different methods are available.

Each of these methods is suited to a different class of business problems.

Lecture – DWH & DM



Overview about Data Mining Applications

1. Market Basket Analysis
2. Cross Selling
3. Customer Retention
4. Fraud Detection
5. Campaign Management

<i>No.</i>	<i>Application</i>	<i>IM4D Technique</i>
1	Market Basket Analysis (MBA)	Associations, Sequential Patterns
2	Cross Selling (CS)	Associations, Classification, Clustering
3	Customer Retention (CR)	Clustering, Classification, Value Prediction
4	Fraud Detection (FD)	Associations, Sequential Pattern, Time Sequence
5	Campaign Management (CM)	Clustering, Classification, Value Prediction

Folie: 187 Dr. H. Völlinger, IBM

The best known applications, which uses Data Mining techniques are:

1. Market Basket Analysis (MBA)
2. Cross Selling (CS)
3. Customer Retention (CR)
4. Fraud Detection (FD)
5. Campaign Management (CM)

The link between the DM applications and IM4D methods is given with the following list:

1. Market Basket Analysis (MBA) ---→ Associations, Sequential Patterns
2. Cross Selling ---→ Associations, Classification, Clustering
3. Customer Retention → Clustering, Classification, Value Prediction
4. Fraud Detection ---→ Associations, Sequential Pattern, Time Sequence
5. Campaign Management ---→ Value Prediction, Classification, Clustering

Examples where this DM applications are used in real-life:

1. Market Basket Analysis (MBA) --→ Retail: How products are placed in the shelf in the store. Goods which are often sold together are placed together in the shelf. Special prices for bundles of goods.
2. Cross Selling CR --→ Automobile companies also offers now Financial Services (Mercedes Benz Bank, VW Bank). Coffee shops also offer clothes other goods for the kitchen or household. Telephone shops offer other articles together with a handy, like a handy envelope (leather).
3. CR --→ 'Miles & More' at Lufthansa or 'Happy Digits' at D. Telekom. Gold Status or Premium Status for customer cards.
4. FD --→ Insider Trading (stock exchange). Fraud Detection at Automobile-Insurance. Fraud detection with EC- or Master-Card.
5. CM --→ special campaigns for special customers which offers better pricing of tariff models. Campaigns to prevent silent attrition of valuable customers.

e-business

Lecture – DWH & DM

DHBW
 Duale Hochschule
 Baden-Württemberg
 Stuttgart

Market Basket Analysis – Business Idea

How can I maximize the per customer profit ?

Sequences

Associations

Analyse customer behaviour

IBM

Folie: 188

Dr. H. Völlinger, IBM

The first method is **Market Basket Analysis (MBA)**.

It shows the result for a retail scenario.

In the foil above one sees an example for the DM concept of Associations & Sequences.

Associations: This is possibly the best known mining technique, thanks to the well-known, but untrue, "diapers (nappies) and beer" anecdote.

As some sets of items will never appear together, and some only infrequently, statistics are used to calculate how often they occur together and decide how confident we can be that this is a real relationship.

Market Basket Analysis - Associations

- Search the table for all available combines and evaluate the frequencies

- **Results**

If a customer buys "product A", then he buys "product B" in Z% of the time. This association is present in X% of all bills



The foils shows how the **Associations** concept is working.

Associations: A relationship implies if one item is present, then the other will likely be present too. This is shown in the 4 concepts of Support, Confidence, Lift and Type reported by IM and the logic of this is fairly easy to understand.

1. Support represents the percentage of transactions in which items appear together. It shows how relevant this is compared to the total number of transactions.

2. Confidence gives the percentage of transactions that contain the first item and also the Second (the probability that the second item will be there). This defines the strength of the pattern, so if 90% of the time when a customer bought X they also bought Y, then X implies Y with 90% confidence).

3. Lift is the probability of both parts of the association occurring compared to how likely on of the parts is of occurring. (actual confidence factor divided by the expected confidence). A high lift means that the connection between the items is stronger. An item which is not expected to occur very often, which always occurs with another item has a strong lift, whereas an item which occurs nearly all the time probably does not have an association and will lead to a weak lift.

4. Type identifies where the lift is statistically significant, meaning that the appearance of one item does imply the appearance of another.

Hierarchies, called **Taxonomies**, can be set up to find associations at different levels. For example there may be no association with product A and a specific cola drink, but there may be one for all cola products or all soft drinks. The taxonomy effectively acts like an alias for the lower level items (children of the item in the hierarchy).

Market Basket Analysis - Sequential Patterns

- Search the table for all available sequences and evaluate the frequencies

- **Results**

If a customer buys "product A", then he buys later "product B". This sequence is present in X% of the total amount of sequences.

Customer 1	Day 1	Product 1
Customer 8	Day 1	Product 1
Customer 1	Day 4	Product 2



IBM

Folie:
190

Dr. H. Völlinger, IBM

The foils shows how the **Sequential Pattern** concept is working.

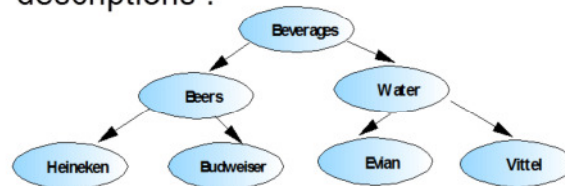
It shows the result for a retail scenario.

Sequential Patterns

Sequential Patterns are similar to associations, but are based on behavior over time, not a snapshot of activity at one point in time. Associations are centered on transactions (A+B happened together), Sequential Patterns on the sequence (A happened, then B, then C).

Market Basket Analysis - Associations Algorithm

- A reference for the whole industry. IBM Almaden research teams have been working for years on this subject
- The quickest algorithms
- Can use directly the transaction data
- Can take into account taxonomies descriptions :



Taxonomies define a hierarchy of relations between categories.

Categories usually represent grouped field values, but can also consist of only one value. When you use a taxonomy with a mining function, the Intelligent Miner treats the categories like values in your input fields.

The Intelligent Miner searches for relationships between categories, between field values, and between categories and field values.

Exercise1 to Lesson 9: Data Mining Techniques

Describe the following Data Mining techniques. Search this information in the internet, i.e. Wikipedia or other knowledge portals:

- **Clustering**
- **Classification**
- **Associations**



Clustering

Clustering is used to segment a database into subsets, the clusters, with the members of each cluster having similar properties. IM for Data can perform clustering by using either a statistical clustering algorithm (Demographic Clustering) or a neural network algorithm (Kohonen Clustering), depending on the type of the input data set. The neural clustering algorithm requires the user to specify the number of clusters required; the statistical clustering algorithm automatically determines the “natural” number of clusters.

When clustering is performed there are no preconceived notions of what patterns exist within the data; it is a discovery process. The results of the clustering process can be visualized to determine the composition of each cluster. Visualization graphically presents the statistical distributions of the characteristics of those records that compose the cluster in comparison with the data set as a whole. Tabular output is also provided to enable further analysis.

In addition to producing graphical and tabular output, a “cluster model” is also generated (Training Mode). It is also possible to generate a user-defined table, which can include selected information from the input records, together with the cluster number of the segment to which the record has been assigned. The output table can also include details on the next nearest cluster and a measure of the confidence in the degree of matching to the nearest and next nearest clusters for each record (Test Mode). An Application Mode is also provided, in which new data records are assigned to clusters and an output table generated. In the commercial environment clustering is used in the areas of cross-marketing, cross-selling, customizing marketing plans for different customer types, deciding on media approach, understanding shopping goals, and so forth.

Classification

Classification is the process of automatically creating a model of classes from a set of records that contain class labels. The induced model consists of patterns, essentially generalizations over the records that are useful for distinguishing the classes. Once a model is induced, it can be used to automatically predict the class of other unclassified records. IM for Data has two classification algorithms, a tree induction algorithm (modified CART regression tree) and a neural network algorithm (back propagation), to compute the classes.

The tree and neural network algorithms develop arbitrary accuracy. While neural networks often produce the most accurate classifications, trees are easy to understand and modify and the model developed can be expressed as a set of decision rules. Commercial applications of classification include credit card scoring, ranking of customers for directed mailing, and attrition prediction. One of the main uses of the tree algorithm is to determine the rules that describe the differences between the clusters generated by the clustering algorithm. This is achieved by taking the output table from the clustering algorithm and constructing the decision tree using the cluster label as the class.

Associations

The association algorithm, developed at the IBM Almaden Research Center in San Jose, California, compares lists of records to determine if common patterns occur across the different lists. In a typical commercial application the algorithm looks for patterns such as whether, when a customer buys paint, they also buy paintbrushes. More specifically, it assigns probabilities; for example, if a customer buys paint, there is a 20% chance that they will buy a paintbrush. The advantage of this approach is that it compares all possible associations. It also finds multiple associations, for example, if a customer buys paint and paint brushes, there is a 40% chance they will also buy paint thinner.

When the algorithm runs, it potentially creates hundreds or thousands of such rules. The user can however select a subset of rules that have either higher confidence levels (a high likelihood of B given A) or support levels (the percent of transactions in the database that follow the rule) or high lift (the ratio of measured to expected confidence for a rule). It is up to the user to read the rules and decide if the rules are:

- Chance correlations (for example, paint and hair rollers were on sale the same day and therefore were correlated by chance).
- Known correlations (for example, the paint and paint brush correlation is something that would have been known).
- Unknown but trivial correlations (for example, red gloss paint and red non gloss paint correlation may be something unknown, and is unimportant to know).
- Unknown and important correlations (for example, paint and basketballs, which may be something previously unknown and very useful in both organization of advertising and product placement within the store).

Association discovery is used in market basket analysis, item placement planning, promotional sales planning, and so forth.

The association algorithm also includes the capability to include a taxonomy for the items in the lists (for example, paint and a paintbrush are hardware) and the algorithm will discover associations across the taxonomy (for example, there is a 50% confidence that customers who buy hardware also buy soft furnishing).

Exercise2 to Lesson 9: Data Mining Techniques

Describe the following Data Mining techniques. Search this information in the internet, i.e. Wikipedia or other knowledge portals:

- **Sequential Patterns**
- **Value Prediction**
- **Similar Time Sequences**



Sequential Patterns

The purpose of discovering sequential patterns is to find predictable patterns of behavior over a period of time. This means that a certain behavior at a given time is likely to produce another behavior or a sequence of behaviors within a certain time frame. The rule generation method is a variation of the association technique. It analyzes the shopping behavior of customers, for example, over time. Instead of looking at 10,000 purchases, the algorithm looks at 10,000 sets of purchases.

These sets are, for example, lists of purchases from a sequence of shopping trips by a single customer. As a typical commercial example, one set of lists may be the purchases of computer:

- Computer in December
- Computer games and joy stick in January
- Additional computer memory and larger hard drive in March

If this sequence, possibly with different time scales but the same order, were repeated across a number of customers, then the sequential association algorithm would typically return a rule, such as:

If following the purchase of a computer, the customer purchases computer games, then there is a 30% chance that extra computer memory will be purchased in a subsequent visit to the store.

The algorithm also includes the capability to define minimum and maximum time periods between the items in the lists. This would, for example, enable the above rule to include the statement that computer memory will be purchased no earlier than one month and within three months of the purchase of the computer games. Sequential pattern detection can therefore be used to discover associations over time. This is especially useful in commercial applications, such as direct marketing, or the design special advertising supplements, and so on.

Value prediction

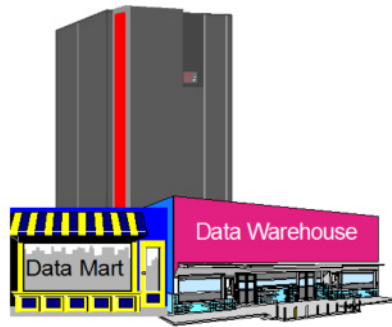
Value prediction is similar to classification; the goal is to build a data model as a generalization of the records. However, the difference is that the target is not a class membership but a continuous value, or ranking. IM for Data has two prediction algorithms: a neural network algorithm and a Radial Basis Functions (RBF) algorithm. The radial basis function is particularly efficient and is appropriate for value prediction with very large data sets.

Similar time sequences

The purpose of this process is to discover all occurrences of similar subsequences in a database of time sequences. Given a database of time sequences, the goal is to find sequences similar to a given one, or find all occurrences of similar sequences. The powerful alternatives afforded by multiple methods are enhanced by the fact that several of the methods are supported by more than one mining technique. Multiple techniques are often used in combination to address a specific business problem.

Lesson 10

DM 2 – Further Methods & Tool Example



Lecture – DWH & DM

Cross Selling – Business Idea

How can I increase the profit of my product lines ?

Associations
Segmentation

Increase Customer Loyalty



Folie: 195

Dr. H. Völlinger, IBM


Associations : The idea is to determine which things go together (e.g., retail chains). But, it can also be used to identify cross-selling opportunities and to design an attractive package or grouping of products and services.

Example:


- Policyholders who hold auto insurance also hold term life insurance product.

 **Lecture – DWH & DM** 

Cross Selling - Methods



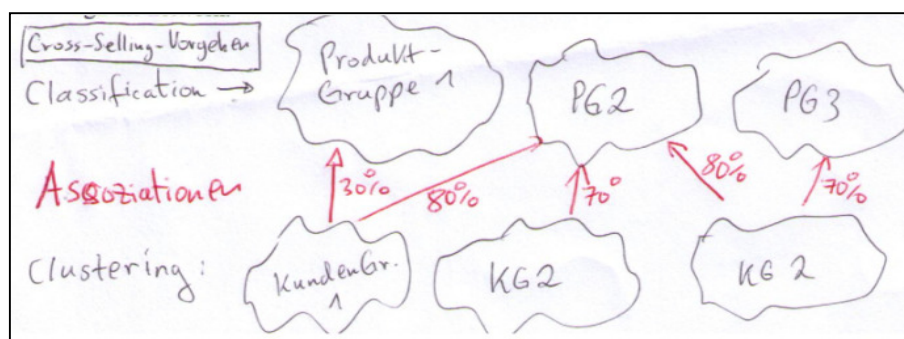
- **Analyse relation products - customer profiles**
 - Use IM Tree / Neural Classification
- **Create homogenous groups of customers, if customers can be identified**
 - Use IM Clustering techniques
- **Analyse products portfolios**
 - Use IM Associations or Sequential Patterns

 Folie: 196 Dr. H. Völlinger, IBM

IM Tree: **Tree Classification** is similar to clustering, in that it assigns records to groups, but different in that it assigns them to predefined groups, *Customer will stay/Customer will go* for example. It uses known data with a known outcome and decides what determined which group the record was in, i.e. the characteristics of each group. This can be used on unknown data to predict which group a new record is likely to belong to. There are various methods tree classification, three well-known names are CHisquare-Automatic-Interaction-Detection(CHAIID), Classification And Regression Trees (CART) and Quick, Unbiased, Efficient Statistical Tree (QUEST). CHAIID will split into two or more child nodes, CART and QUEST only into two child nodes.

Neural Classification: This, as you would expect, does the same job as Tree classification, but in a different way. The model is not a tree, but a network of 'neurons', the connection between each neuron has a strength or 'weight'. Each variable will be connected to some neurons and will have a weighting effect on whether the connection ends up as one outcome or another. The different values of the connection weights will give a different likely outcome depending on the path through the network.

Motivation for “Cross Selling”:





e-business

Lecture – DWH & DM



DHBW
Duale Hochschule
Baden-Württemberg
Stuttgart

Cross Selling - Goals



➤ **Goal :**

- Offer complementary products to existing customers
- Detect when a customer's behaviour changes to offer him new products
- Build promotion strategies
- Create new products

Increase Profit with your marketshare



Folie:
197

Dr. H. Völlinger, IBM

IM Clustering: **Demographic Clustering & Neural Clustering**

Clustering, takes records and fits them into groups (clusters), which should be as similar as possible within the group and as dissimilar to other groups as possible.

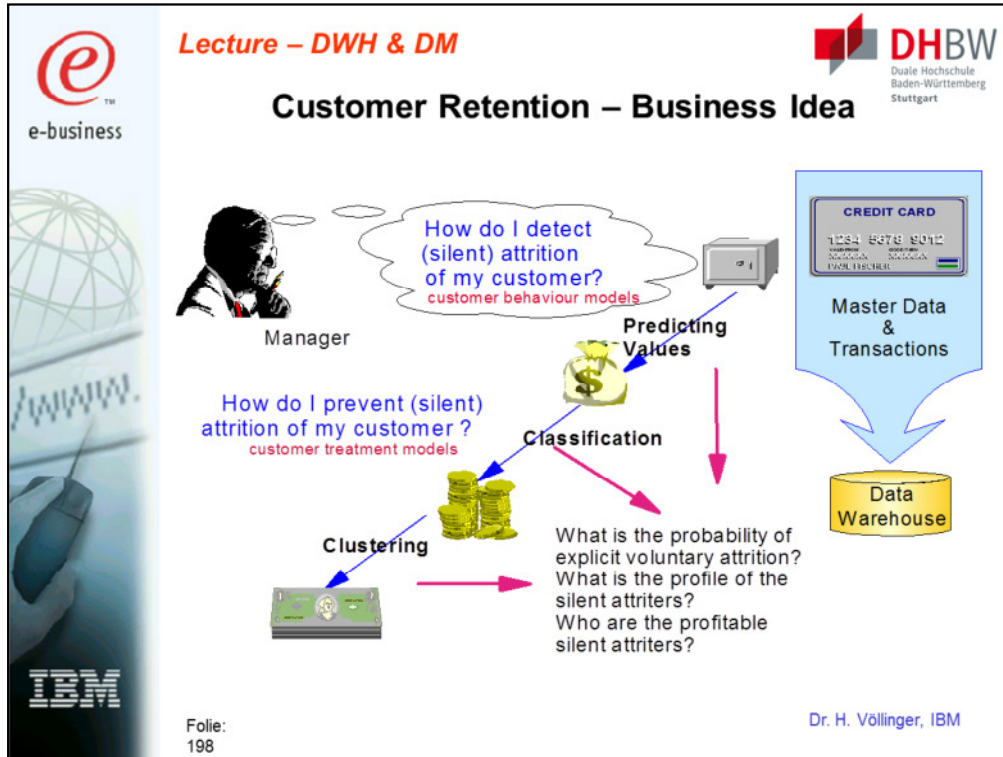
Demographic clustering uses something called the **Condorcet Criterion**, which it tries to maximize, to determine the cluster to which a record fits the best. This is easy for one variable, say sex, each record is either a boy or a girl, but when you have many variables, it is trickier to say how different a record is, but is similar enough to be in an existing group, or if it should be the first member of a new group.

For each variable, IM will class categorical variables as similar if they are the same, so sex=M is only similar to another variable where sex is also M. For numerical variables IM has to decide how far apart they can be, and still be classed as similar.

Neural clustering does the same job as Demographic clustering, but in a different way. IM uses a technique called Kohonen Feature Maps which use a process called self-organization to group similar input records together.

This is trickier to understand and more compute intensive than Demographic clustering.

Kohonen Feature Maps rely on a network of clusters (fixed number) and IM has to decide which cluster a record belongs to. This is done by some fairly serious statistics.



Prediction is the same as classification or estimation except that the records are classified according to some predicted future behavior or estimated future value. In a prediction task, the only way to check the accuracy of the classification is to wait & see.

Historical data is used to build a model that explains the current observed behavior.

Example: Predicting which customers will leave within the next 6 months

Classification consists of examining the features of a newly presented object (e.g., record in a database) and assigning it to one of a predefined set of classes on the basis of a model developed through training on pre-classified examples.

In a case of records in a database, classification consists of updating each record by filling in a field with a class code. It deals with discrete values: Yes/No, Accept/Reject, ...

Examples of classification tasks include:

- classifying credit applicants as low, medium and high risk;
- spotting fraudulent insurance claims.


Customer Retention – Business Goals

- Identify customers who left
- Build a training model
 - ▶ Create training and test data on historical basis
 - ▶ Learn the algorithm with training data
 - ▶ Check results with test data
- Run model against current customer data


■ **Result Analysis**

■ **Business Implementation**






Lecture – DWH & DM



Customer Retention - Methods

- Data Mining
 - ▶ Customer scoring
 - Classification Tree / Neural
 - Prediction RBF / Neural
 - ▶ Characterize Defectors
 - Clustering Neural / Demographic



Folie: 200

Dr. H. Völlinger, IBM

Value Prediction Radial Basis Functions does not try to fit a line to all values, but looks for groups of similar values. Radial Basis Functions represent the functions of the distance (or radius) from a particular point. RBF tries to find regions in the input space where the outputs are all similar and creates an RBF center for each group that predicts the average output of the group.

RBF is sophisticated and copes with out of line values (**noise**) by leaving a percentage of the data out and testing its model at the end of each pass against this 'held-out' data. If doing an extra pass did not help on this held-out data the radial basis functions stops.

Value Prediction Neural Networks

The process for Neural Networks Value Prediction is similar to that for Classification.

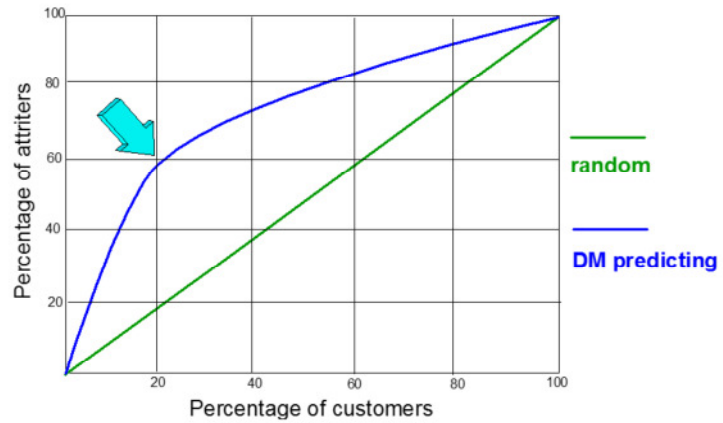
(Neural Value Prediction and RBF are fairly close too, but RBF is faster.)

As with RBF, Neural Value Prediction determines the value of a variable, based on known values of other variables.

If the value to be predicted is from a numeric variable, the result will also be a numeric, such as quantity or monetary value. If the value to be predicted is for a character variable, then the result is a number representing the probability of a specific value of the variable occurring.

Neural Value Prediction can predict values and time-series (multiple values, as a linear regression can predict a time series if you extend the line.) As with Neural Classification, it uses a back propagation algorithm, to see if it has passed the point where further improvements can be made.

Customer Retention – Attrition Response Model



Consider the following attrition model, which shows a high volume of customers which a candidates to leave the enterprise.

The goal of Data Mining is to reduce this number to an average number, which is shown in the random curve.

Customer Retention – Goal

Goal:

- Identify profitable customers with high probability of defection
- Execute campaign to target defectors
- Use model to be pro-active

Substantial cost saving

IBM

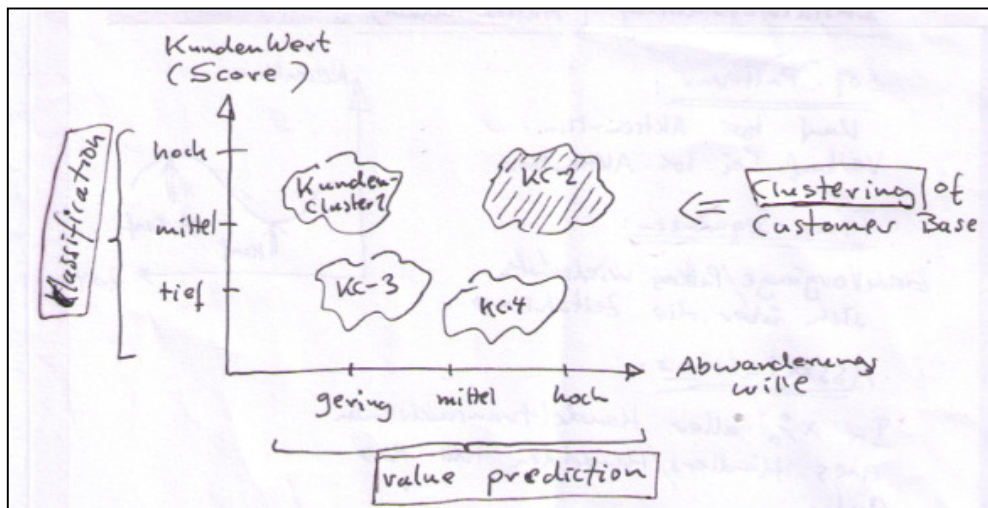
Folie:
202

Dr. H. Völlinger, IBM

The following picture gives a motivation to “Customer Retention”. In a first step you run Clustering to get Customer Clusters CC1, CC2, CC3 and CC4. The customers in each cluster have the two properties:

1. Value of Customer (“Classification”)
2. Probability of Defection = “Abwanderungswille” (“Value Prediction”).

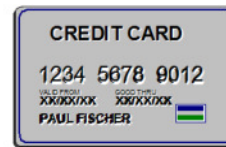
Now place the customer clusters (CC) in this two axes. You will then see CCs will be valuable to start special marketing activities with them, to prevent these customer to go away as customers. See picture:



Fraud Detection – Idea & Goal

Question :

How is it possible to avoid the damages caused by fraudsters ?



Goal :

- Detect quickly fraudulent transactions
- Identify potential frauders
- Stop immediately services to frauders

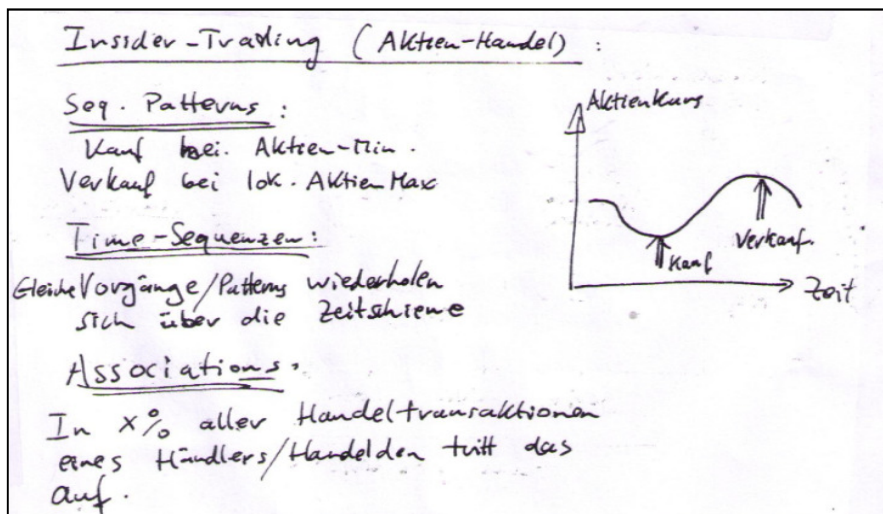
Reduces risks, saves money



The following picture gives a motivation to “Fraud Detection”. You use the 3 methods:

1. Sequential patterns
2. Time Sequences
3. Associations

See the example for “Insider Trading”:



Further examples of “Fraud Detection”:

- Fraud with credit cards
- Terroristic activities
- Crime cases
- Fraud in insurance. i.e. car insurance: examine all information/data about claims (look for pattern in a time sequences. Repeating = association)

Lecture – DWH & DM

Campaign Management – Business Idea

Can I be more effecient in my direct marketing strategy ?

Clustering
Scoring

Increase response
Save money

Folie: 204

Dr. H. Völlinger, IBM

The next data mining application is “Campaign Management”. The goal is to improve the success of marketing actions (i.e. marketing campaigns). We will use for this clustering and also scoring methods (see next foils).

Campaign Management – Methods

➤ *Build homogenous groups of customers*

- Use *automatic* multidimensional segmentations
- DM : two techniques :
 - Neural clustering
 - Demographic clustering
- Analyse segments profiles



IBM

Folie:
205

Dr. H. Völlinger, IBM

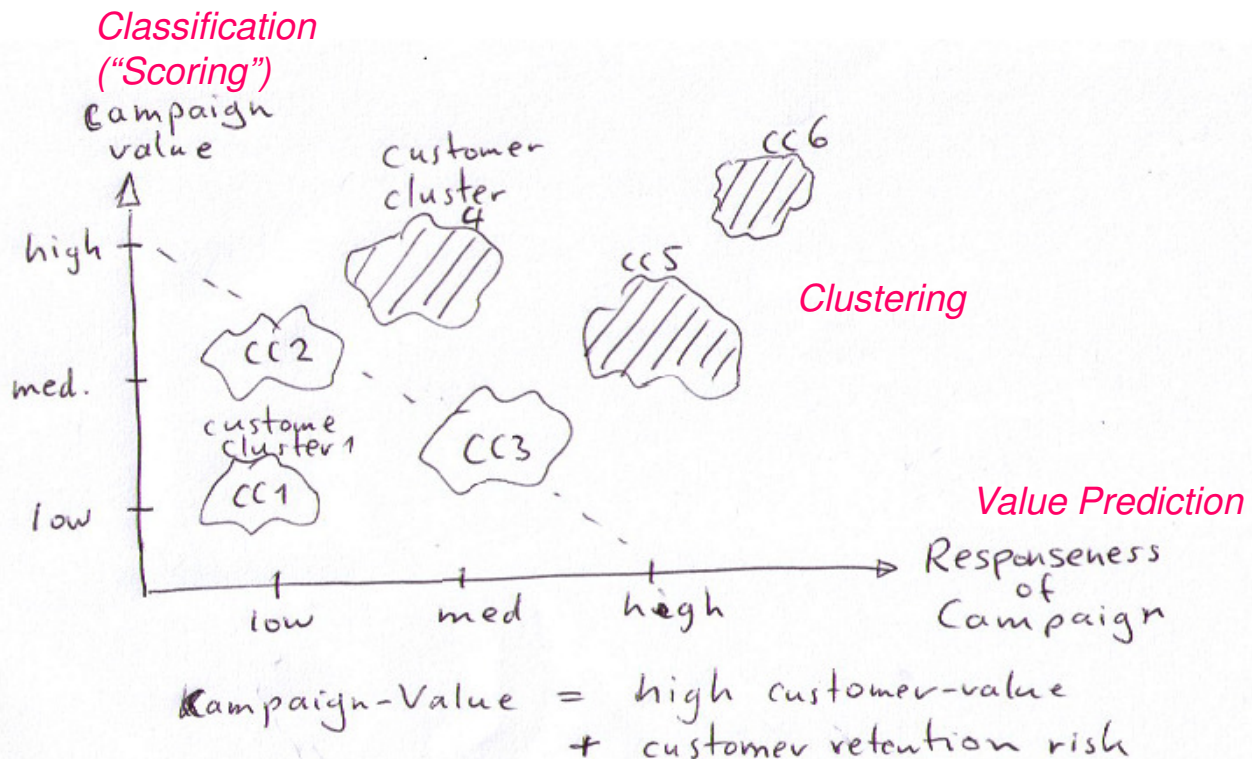
First Method: Build customer segments – “Clustering”

Campaign Management – Methods

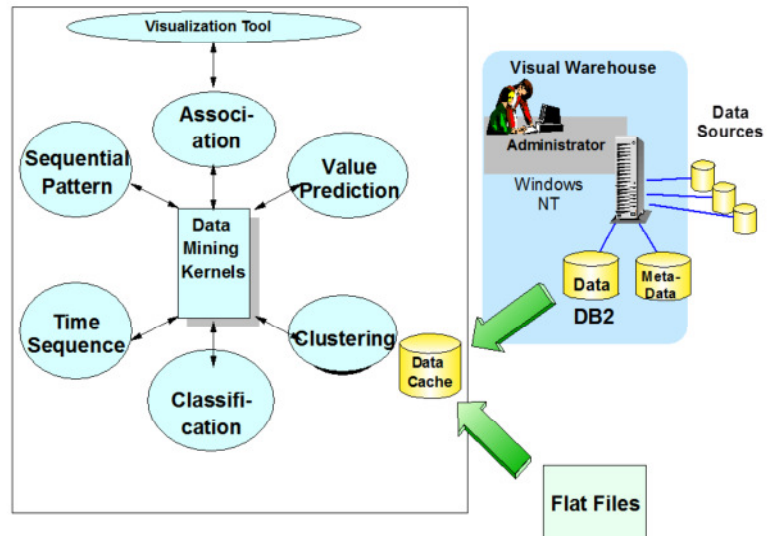
- **Choose the interesting segments**
- **Start the Campaign on a sample of people - adapt message to profile**
- **Analyse deeply the campaign results**
 - Build a model to explain why some replied and some did not
 - Use a scoring method
 - IM RBF Prediction
 - IM Neural Prediction
 - IM Tree/Neural Classification



For a motivation for “Campaign Management” see the following picture:



IM for Data - Overview

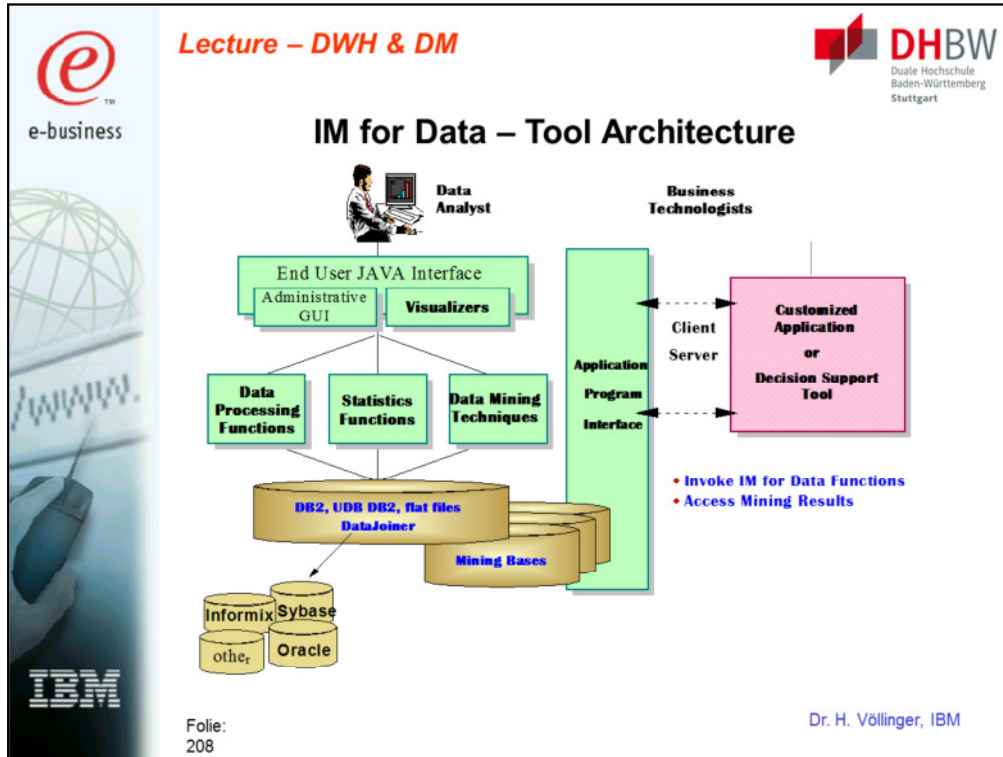


Folie:
207

Dr. H. Völlinger, IBM

IM4D consists of the following six basis Data Mining techniques:

1. Association
2. Sequential Pattern
3. Value Prediction
4. Clustering
5. Classification
6. Time Sequence



Next we will see a demo of Im4Data:

The business problem

Imagine that you work for a bank that sells several products, including Regular Checking, Premier Checking, and Exclusive Checking accounts and option packages for each account. The bank already knows that Premier Checking is their most profitable product, and wants to increase the number of customers who have this type of checking account. The marketing department wants to identify

different groups based on demographic data, such as age and income, within the Premier Checking customers so that the department can prepare different ad campaigns for each of the groups.

Additionally, the department wants to identify customers who are not currently Premier Checking customers who have similar demographics to the customers who are Premier Checking customers.

You have obtained some customer data from corporate headquarters to solve this business problem. This data is named **banking.txt**. It contains information about customers from all branches of the bank. You can use the Intelligent Miner to mine this data and provide demographic information to the marketing department. Your customer data includes information about customers who already have the Premier Checking account, so you can use the Demographic Clustering mining function to

identify different groups, based on demographic data, among customers who already have Premier Checking.

The slide features a vertical sidebar on the left with the 'e-business' logo (an '@' symbol) and the IBM logo at the bottom. The main content area is white with a black border. At the top right, the DHBW logo is present. The title 'IBM IM for Data - Life Demo Overview' is centered. Below the title, a text block states 'The demo will demonstrate the five phases of data mining tasks:' followed by a numbered list of five items: 1. Defining the data, 2. Building the model, 3. Applying the model, 4. Automating the process, and 5. Analyzing the results. At the bottom left, it says 'Folie: 209' and at the bottom right, 'Dr. H. Völlinger, IBM'.

Lecture – DWH & DM

IBM IM for Data - Life Demo Overview

The demo will demonstrate the five phases of data mining tasks:

1. **Defining the data**
2. **Building the model**
3. **Applying the model**
4. **Automating the process**
5. **Analyzing the results**

Folie: 209

Dr. H. Völlinger, IBM

This Intelligent Miner demo consists of an abbreviated data mining scenario with five phases: Defining data, building a model, applying the model, automating the process, and analyzing the results. By following the steps in this tutorial, you will learn how to use the Intelligent Miner wizards to define data objects, run mining functions, and view results in the Intelligent Miner.

Defining the data

Define a data object that points to a flat file containing your customer data file banking.txt. The data object will be named **Customers**.

You must specify which properties of your customers are contained in the data, their data types, and the columns in the flat file that they occupy.

The Intelligent Miner data objects simply point to the location of your data, so that the Intelligent Miner can process this data. You will not actually be changing the contents of the banking.txt file.

Building the model

Define a Demographic Clustering settings object named **Build model**. This settings object uses the Customers data object as the input data. It runs in clustering mode, and produces a results object named **Model**. This model contains information that describes the clusters identified during the mining run.

Lecture – DWH & DM

DHBW
Duale Hochschule
Baden-Württemberg
Stuttgart

IBM Intelligent Miner for Data - Life Demo

Folie: 210

Dr. H. Völlinger, IBM

Applying the model:

Define a Demographic Clustering settings object named **Apply model**. This settings object uses the Customers data object as the input data. It runs in application mode using the Model results object and produces an output data object named **Scored customers** and a flat file named **scored.txt**. This output file identifies the subgroup associated with a customer record.

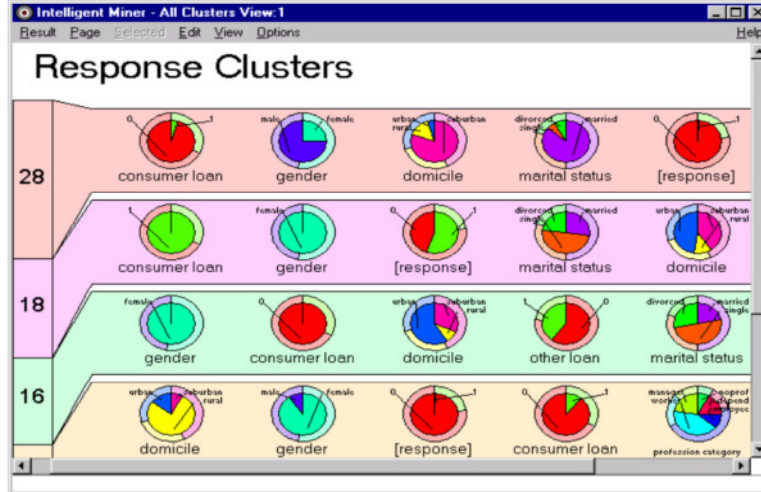
Automating the process:

To automate the process you create a sequence object **Target Marketing** containing the Build model settings object and the Apply model settings object. A sequence is an object containing several other objects in a specific sequential order. You can run a sequence, which runs each of the objects within the sequence in the order that you specified. This allows you to combine several mining tasks into one step.

Analyzing the results:

Define a Bivariate Statistics function named **Analyze**. This statistical function analyzes the data object **Scored customers** and produces an output data object **Target customers**, a flat file **target.txt**, and a result object **Target customer demographics**.

IBM Intelligent Miner for Data - Life Demo 2

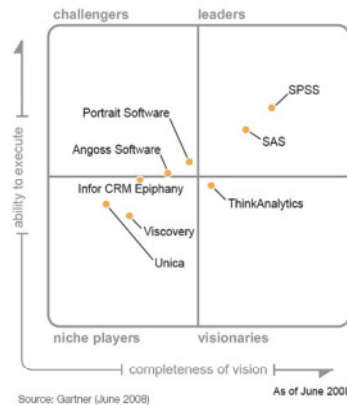


Folie:
211

Dr. H. Völlinger, IBM

Exercise 1 to Lesson 10: Data Mining Tools

Search for the actual “Gartner Quadrant” of DM tools.
Give detail description of two of the leading DM tools
in the quadrant:



Sources: Gartner (June 2008)

As of June 2008

Dr. H. Völlinger, IBM

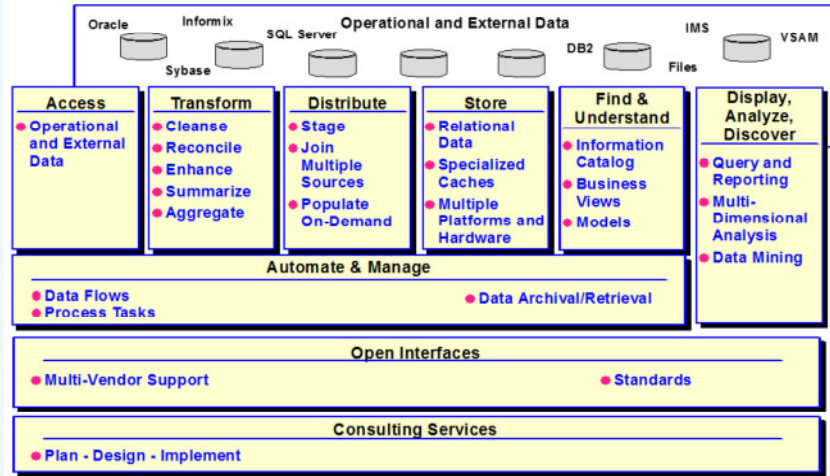


Anhang

BACKUP Folien



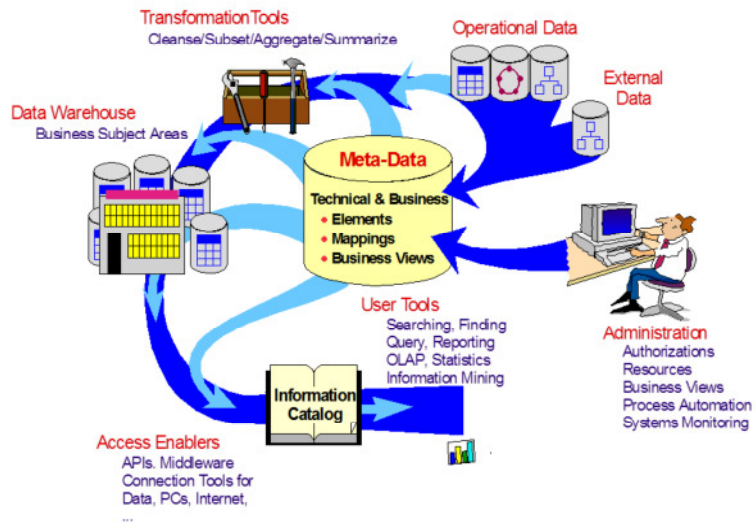
Components of a Data Warehouse



Enabling the Solution



DWH Architecture – Processes



Process Layers of the DWH

