Exercises (+Solutions) to BA Lesson DWH&DM – Chapter 1

Exercise 1.1:

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) (operational, analytical, collaborative)**

For Explanation of these 'catchwords' see also the foils of the lesson or search in the internet

Optional: Give an explanation also for the synonyms like: OLAP, OLTP, ETL, ERP, EAI

Solution:
DWH – Data Warehousing:

In vielen Organisationen sammeln sich in den operativen Systemen große, isolierte und meist unterschiedlich formatierte Datenmengen an. Durch Transformation dieser Daten und hinzufügen externer Daten wird es möglich, Informationen integriert im Data Warehouse – eine Art Warenlager für Daten – für Abfragen und weitergehenden Analysen bereitzustellen.

BI – Business Intelligence:

BI ist der Prozess, die angesammelten, rohen, operationalen Daten zu analysieren und sinnvolle Informationen daraus zu extrahieren, um auf Basis dieser integrierten Informationen bessere Geschäftsentscheidungen treffen zu können.

BI ist wenn Geschäftsprozesse anhand der aus dem Data Warehouse gewonnenen Fakten optimiert werden.

CRM – Customer Relationship Management:

CRM steht für kundenorientiertes Handeln, d.h. nicht das Produkt, sondern der Kunde ist Mittelpunkt aller Geschäftsentscheidungen. Durch besseren und individuelleren Service sollen neue Kunden gewonnen und bestehende Kundenkontakte gepflegt werden.

Operatives CRM:

Lösungen zur Automatisierung / Unterstützung von Abwicklungsprozessen mit Kunden (Online Shop, Call Center,…)

Analytisches CRM:

Lösungen, die auf Informationen des Data Warehouse zurückgreifen und auf aufgabenspezifische Analysen (Data Mining) beruhen.

Kollaboratives CRM:

Kommunikationskomponente, die die Interaktion mit dem Kunden ermöglicht.

Gewinnung von Erkenntnissen durch Zusammenarbeit mit dem Kunden. Diese können dann zur Optimierung der Geschäftsprozesse oder Personalisierung der Kundenbeziehung genutzt werden.

OLAP – Online Analytical Processing:

Der Begriff OLAP fasst Technologien, also Methoden, wie auch Tools, zusammen, die die Ad-hoc Analyse multidimensionaler Daten unterstützen. Die Daten können aus dem Data Warehouse, Data Marts oder auch aus operativen Systemen stammen.

(Abgrenzung Data Mining: Suche nach Mustern und bislang unbekannten Zusammenhängen (Neuronale Netze, Warenkorbanalysen,…))

OLTP – Online Transactional Processing:

Operative Softwaresysteme mit deren Transaktionsdaten. Heute analysiert man weniger diese operationalen Daten als vielmehr multidimensionale, navigierbare Daten (OLAP).

ETL – Extraction, Transformation and Loading:

Ein ETL – Tool ist dafür zuständig, um aus den operationalen Daten (real-time-data) gesäuberte und eventuell aggregierte Informationen sowie zusätzliche Metadaten zu erhalten.

ERP – Enterprise Resource Planning:

Unternehmensübergreifende SW-Lösungen, die zur Optimierung von Geschäftsprozessen eingesetzt werden. Dabei handelt es sich um integrierte Lösungen, die den betriebswirtschaftlichen Ablauf in den Bereichen Produktion, Vertrieb, Logistik, Finanzen und Personal steuern und auswerten.

EAI – Enterprise Application Integration:

EAI beschäftigt sich mit der inner- und über-betrieblichen Anwendungsintegration, um einen problemlosen Daten und Informationsaustausch zu gewährleisten.

Aktuelle Trends:

1) **Explodierendes Datenvolumen**
   - Stärkster Trend
   - Laut Gartner soll 2004 das Datenvolumen 30x so hoch wie 1999 sein.
   - Skalierbarkeit

2) **Integrierte 360° Sicht**
   - Der Kunde soll völlig transparent sein
     ⇒ Trotz verteilter Applikationen soll ein vollständiges Bild des Kunden vorhanden sein. → wichtig für CRM

3) **Komplexe Anfragen und Analysen**
   - Benutzeranforderungen an DWH- / BI- und CRM- Systeme steigen
   - Anfragen nehmen zudem zu

4) **Mehr Endbenutzer**
   - BI- und DWH- Systeme müssen zugänglicher werden
     ⇒ Benutzbarkeit „weniger ist mehr“

5) **Fussion von DWH und CRM**
   - Information (in den DWH's) ist die Basis, um Kunden zu versehen

6) **Active DWH**
   - Wettbewerbsdruck → Daten müssen schnell da sein
   - Aktive DWH sind eng an operationale Systeme gekoppelt → sehr aktuelle Daten + sehr detailliert

7) **Datenansammlungen ('Data Hubs') statt relationaler DBs**
   - Billiger + schneller, aber: kein SQL + nicht für jede Situation

8) **Outsourcing**
   - Zu Anfang Applikationen + Daten; zukünftig auch die Informationshaltung im DWH

9) **Starkes Anwachsen von Datenquellen (z.B. e-Business)**
   - Mehr Daten in unterschiedlichen Plätzen

10) **Re-Engeneering oder sogar Neuaufbau von Business- Systemen (DWH, …)**
   - Kunde war nicht Mittelpunkt oder wurde nicht vollständig betrachtet; Falschplanung (Größe, Geschwindigkeit, …)

Further Solution: (SS 2014):

Data Warehouse

Überblick und Trends

Agenda

- Motivation
- Überblick
- DWH Trends
- Zusammenfassung

Agenda

- Motivation
- Überblick
- DWH Trends
- Zusammenfassung

Motivation

- Wachsende heterogene Datenbestände in Unternehmen
- Erschwerte Entscheidungsfindung aufgrund zunehmender Komplexität
- Analyse und Auswertung muss gewährleistet sein

→ DWH unterstützt Lösung dieser Problemstellungen
→ "Turning Data into Information!"

[zitiert aus DWH Skript Dr. Hermann Völlinger]

Agenda

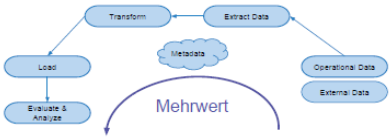
- Motivation
- Überblick
- DWH Trends
- Zusammenfassung

Überblick (I/II)

„A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process."

[Quelle: W.H. Inmon (1996), Seite 33]

Überblick (II/II)



Agenda

- Motivation
- Überblick
- DWH Trends
- Zusammenfassung

Trends (I / III)

1. In-Memory-Datenhaltung
   - Echtzeitanalyse
   - In-DB-Analyse
2. DWH Sicherheit
   - z.B. Banken, Versicherungen, etc.

Trends (II / III)

3. NoSQL DBMS
4. "Datafication" des Unternehmens
   - Mobile Geräte
   - Sensoren (RFID,etc.)
   - Soziale Netzwerke
5. DWH Appliances
   - vorkonfigurierte DWH Lösungen

Trends (III / III)

6. Analytics as a Service
   - dynamische Workloads
   - Anwendung z.B. Rapid Protoyping
7. In-Database Analysen
   - Data-Mining Algorithmen im DBMS
   - Integration von Statistischen Prog.Sprachen (z.B. R)

Agenda

- Motivation
- Überblick
- Trends
- Zusammenfassung

Zusammenfassung

Extraktion / Transformation / Laden von Daten aus verschiedenen Datenquellen mit den Zielen:

- Entscheidungsfindung zu unterstützen
- Datenqualität & -konsistenz sicherzustellen
- Businessprobleme zu lösen
- Einfachen, konsistenten & verständlichen Zugriff auf Daten für alle Beteiligten zu liefern

Business Intelligence

Überblick und Trends

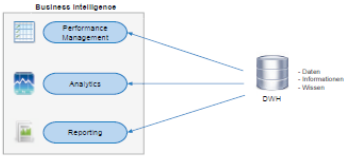
Agenda

- Überblick
- BI Trends
- Zusammenfassung

Agenda

- Überblick
- BI Trends
- Zusammenfassung

Überblick



Agenda

- Überblick
- BI Trends
- Zusammenfassung

Trends

Cloud Computing

Visual Data Discovery

Mobile First

Trends

Visual Data Discovery

- Agile Analyse
- Benutzerfreundlich
- Self-Service
- z.B. Tableau

Trends

Cloud Computing

- Marktreife von Cloud
- Infrastrukturelle Vereinfachung
- JasperSoft
  - Amazon-Cloud Host
  - Self-Service BI für jeden
- Microsoft Power BI, Oracle, ...

Trends

Mobile First

- HTML / Nativ
- Funktionen
- Zugriff / Authorisierung / Sicherheit
- Offline Support
- Nutzer interessiert an Mobile?
- Cognos Mobile

Trends

Weitere Trends, die BI unterstützen

- Informationsgewinn aus unstrukturierten Daten
- Schnellere Analyse von Datenmassen (strukturiert)
- Neue, bessere Arten von Suchen

Agenda

- Überblick
- BI Trends
- Zusammenfassung

Zusammenfassung

Nutzung und Auswertung von verteilten und inhomogenen Daten aus Data Warehouse Anwendungen mit den Zielen:

- Geschäftsabläufe, CRM profitabler machen
- Kosten senken
- Risiken minimieren
- Wertschöpfung vergrößern

→ Erfolgskritisches Wissen über Status, Potentiale und Perspektiven erzeugen

Customer Relationship Management

Überblick und Trends

Agenda

- Was ist CRM?
- Ausblick und Trends
- Zusammenfassung

Agenda

- Was ist CRM?
- Ausblick und Trends
- Zusammenfassung

Was ist CRM?

„Products come and go, but customers remain"
[Rust, Zeithaml, Lemon 2000, S. 6]

Was ist CRM?

→ Ganzheitliche, strategische Ausrichtung aller Geschäftsprozesse am Kunden

→ IT-gestützt

Was ist CRM?

Agenda

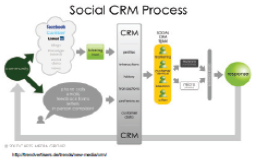
- Was ist CRM?
- Ausblick und Trends
- Zusammenfassung

Trendthemen CRM 2014

Social CRM

- Aufbau eigener CRM-Kanäle auf sozialen Plattformen
- Webmonitoring, Data Mining
- Verlagerung der "klassischen" Kommunikation

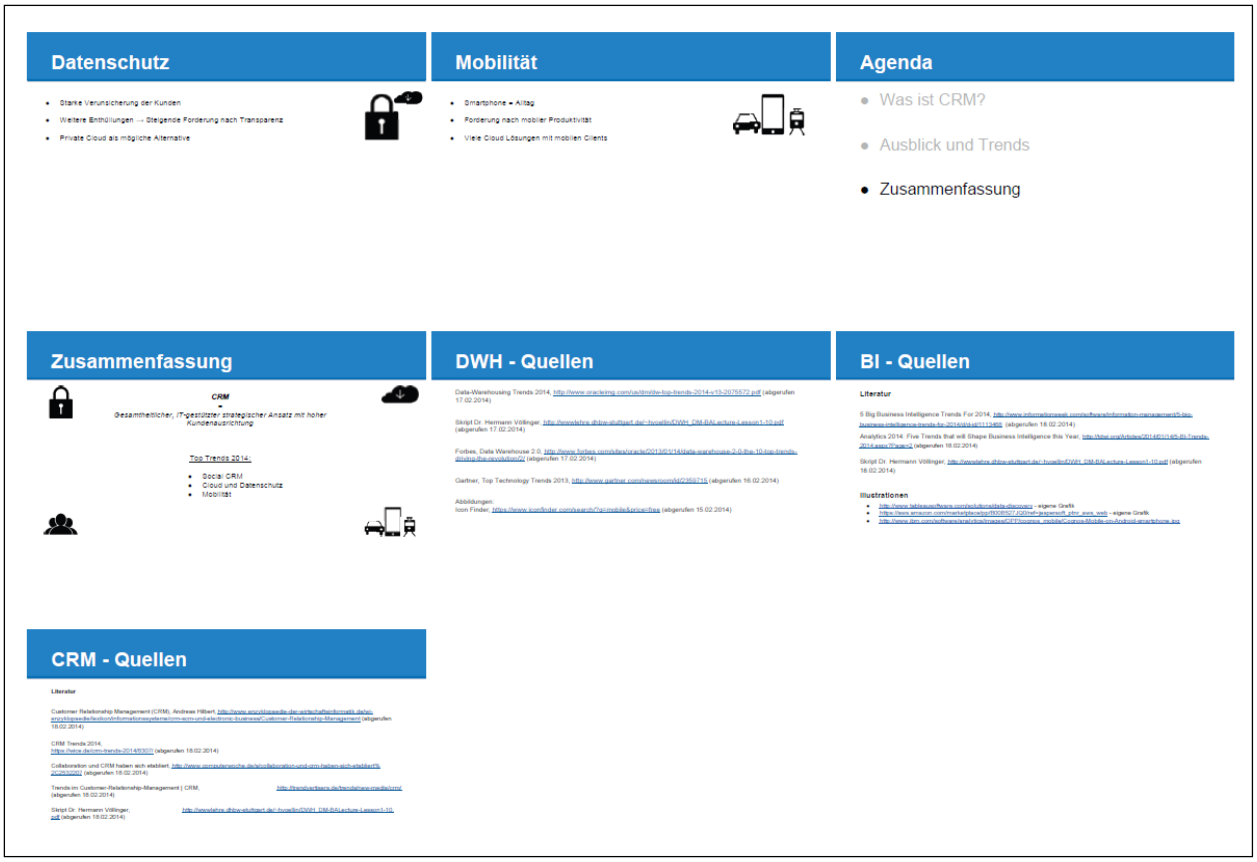
Social CRM



Cloud Computing

- CRM as a Service
- 40% CRM Erlöse in der Cloud

"In kaum einem anderen Bereich der Unternehmensapplikationen hat das SaaS-Modell eine schnellere Verbreitung gefunden als beim Kunden- und Kontakt-Management."



Exercise 1.2:

***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

Solution:

IBM

Basel II

Michael Illiger, Stefan Tietz, Steve Gebhardt, Thomas Dürr

Agenda

- **Warum Basel-Abkommen?**
- **Überblick Basel I + II**
- **Basel II Roadmap**
- **Basel II und Data Warehousing**
- **Tools**
- **Ausblick**

Warum Basel-Abkommen?

- **Risiko: Kreditausfall**
- **Geringe Eigenkapitalquote**
- **Keine einheitlichen Rating-Richtlinien**
- **→ Basel I (1988)**

Basel I

- **8% der Kreditsumme durch Eigenkapital abdecken**
- **Kunden-Rating anhand interner Prüfungen**
- **Grundlage: Bilanzen + bisherige Kreditwürdigkeit**
- **→ falsche Anreizsetzung, unabgedeckte Risiken**

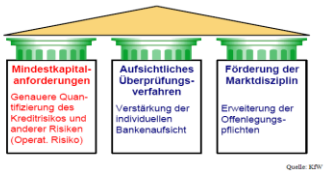
Kirch-Krise, Bankenkrise in Japan

→ Basel II

Basel II

- **Kundenrating intern <u>und</u> extern**
- **Reservebildung je nach Kreditrisiko**
- **Aufteilung in Qualitative und Quantitative Risikofaktoren**
- **erweiterte Offenlegung der Finanzsituation in Banken**
- ***Eigenkapital*** **=** *Kreditsumme* **x** *Risikogewicht* **x** *Kapitalquote*

Basel II - Säulenmodell



Basel II Roadmap

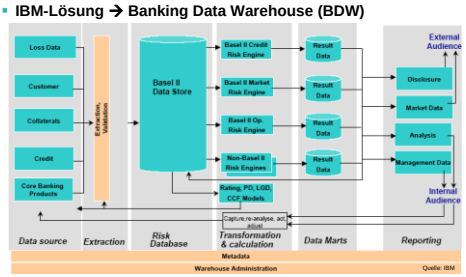
- **2003:** **Aufnahme von Basel II in die Strategie der Institute**
- **2004:** **Aufbau der DWH-Infrastruktur**
- **2005:** **Datensammlung + Auswertungsstrategie**
- **2006:** **... Parallel-Lauf von Basel I + II**
- **2007:** **Basel II wird bindend**

Basel II und Data Warehousing

- **grosse Datenmengen zur Analyse**
- **DWH werden benötigt von:**
  - Banken → Kunden-Rating
  - Rating-Agenturen → Service zur Verfügung stellen
  - Unternehmen → optimale Finanzsituation verringert Kreditkosten

Tools

- **IBM-Lösung → Banking Data Warehouse (BDW)**



Ausblick

- **Vorbereitungen laufen seit 2003**
- **Start: 01.01.2007**
- **verändert Verhältnis zwischen Kapitalgeber und Kapitalnehmer**
- **umfangreicher Absatzmarkt für DWH-Services entsteht (weltweit!)**

Eine weitere Lösung zu Basel2 und DWH ist wie folgt:

Basel II & DWH

Christian Schäfer, 28.10.2005

Agenda

- Sicherung der Stabilität im Finanzsektor
- Eigenkapitalvereinbarung von 1988 (Basel I)
- Von Basel I zu Basel II
- Gründe für Basel II
- Rating von Krediten nach Basel II
- Die Bank will uns kennen lernen
- Auswirkungen von Basel II
- Herausforderungen an Data Warehouse Systeme

Sicherung der Stabilität im Finanzsektor

- Umgang mit Kredit-, Markt, Liquiditäts- und anderen Risiken ist Aufgabe und Geschäftszweck von Kreditinstituten/Banken

Probleme:

- Der freie Umgang mit Risiken darf nicht zu Instabilitäten im Finanzsektor führen
- Freier Umgang bei der Sicherung von Risiken führt zu unterschiedlichen Wettbewerbsbedingungen im Bankwesen

Sicherung der Stabilität im Finanzsektor

Lösung:

- Sicherung einer angemessenen Eigenkapitalausstattung der Banken
- Schaffung einheitlicher internationaler Wettbewerbsbedingungen

Eigenkapitalvereinbarung von 1988 (Basel I)

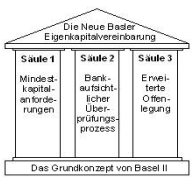
- Richtlinie zur Förderung und Sicherung eines funktionierenden Bankwesens.

Wichtigstes Merkmal:

- Eigenkapitalunterlegung seitens der Bank muss mindestens 8% der Kreditsumme betragen

Von Basel I zu Basel II

- Kritik an Basel I
  - Risiken der Kreditvergabe werden unzureichend abgebildet
  - Neue Finanzierungsmöglichkeiten werden nicht berücksichtigt
- Verbesserungen durch Basel II
  - Internes oder externes Rating von Kreditrisiken
  - Marktrisiko (Branche etc.)
  - Operationelles Risiko (Ausfall Von Mitarbeiter, Systeme etc.)
  - Bankaufsichtlicher Überprüfungsprozess
  - Erweiterte Offenlegung (Selbstkontrolle des Marktes)



Rating von Krediten nach Basel II

- Bewertung von Kreditnehmern durch unabhängige Ratingagenturen wie z.B. Standard & Poor's (externes Rating)
- Bewertung von Kreditnehmern durch den Kreditgeber selbst (auch IRB-Ansatz genannt). Genehmigungspflichtig durch Bankenaufsicht
  - Harte Faktoren
    - Branchenabhängiges Risiko
    - Geschäftsprofil und Wettbewerbsposition
    - Ertragskraft
    - Kapitalanlagen- und Ausstattung
    - Liquidität
  - Weiche Faktoren
    - Unternehmensstrategie
    - Personalpolitik
    - Managementqualitäten
    - Organisationsstrukturen

Erhebung großer Datenmengen

Die Bank will uns kennen lernen

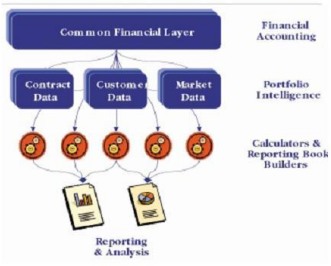
- Datenerhebung zur Berechnung des erwarteten Kreditausfalls (Expected Loss)
  - Adresse
  - Transaktionen
  - Laufende Kredite
  - Liquidität
  - Eigenkapital
  - ...
  - -> Differenzierte Kreditbewertung
  - -> Berechnung der Eigenkapitalquote

Auswirkungen von Basel II

Herausforderungen für Data Warehouse Systeme

- Daten der Kunden als Basis für Analysen und Statistiken
- DWH als Möglichkeit zur Steuerung der Qualität des Kreditportfolios
- Transaktionen, Kontostände, laufende Kredite zur schnellen Bewertung von Risiken
- Erkennen von externen Einflussfaktoren (Marktgeschehen, etc.)
- Daten müssen Konsistent sein und mindestens eine Historie von 3 Jahren aufweisen

DWH Konzept



Roadmap Basel II

Monat	Jahr	Ereignis
Juli	1988	Veröffentlichung der Baseler Eigenkapitalvereinbarung (Basel I)
Ende	1992	Inkrafttreten von Basel I
Januar	1996	Baseler Marktrisikopapier
Juni	1999	Erstes Konsultationspapier zur Neufassung der Eigenkapitalvereinbarung (Basel II)
Januar	2001	Zweites Konsultationspapier zu Basel II
Dezember	2001	Änderung des ursprünglich vorgesehenen Zeitplans für die Fertigstellung des neuen Akkords.
Mai	2003	Drittes Konsultationspapier zu Basel II
Mitte	2004	Veröffentlichung der neuen Eigenkapitalvereinbarung
Ende	2006	Inkrafttreten von Basel II

Quellen

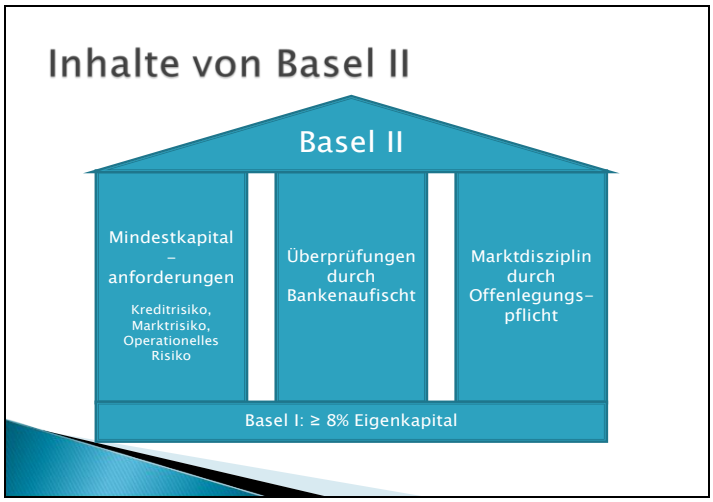
- http://de.wikipedia.org/wiki/Basel_II
- http://www.bundesbank.de/bankenaufsicht
- http://www.foerderland.de/353.0.html
- http://www.basel-ii.info/
- http://www.hp.com/

Eine weitere Lösung (dritte Lsg.) zu Basel2 und DWH finden Sie in der folgenden Darstellung:

Basel I: Kreditvergabepraxis limitiert durch Verknüpfung mit Eigenkapital
Vergabe von Krediten an Kunde mit mäßiger Bonität -> höhere Zinssätze

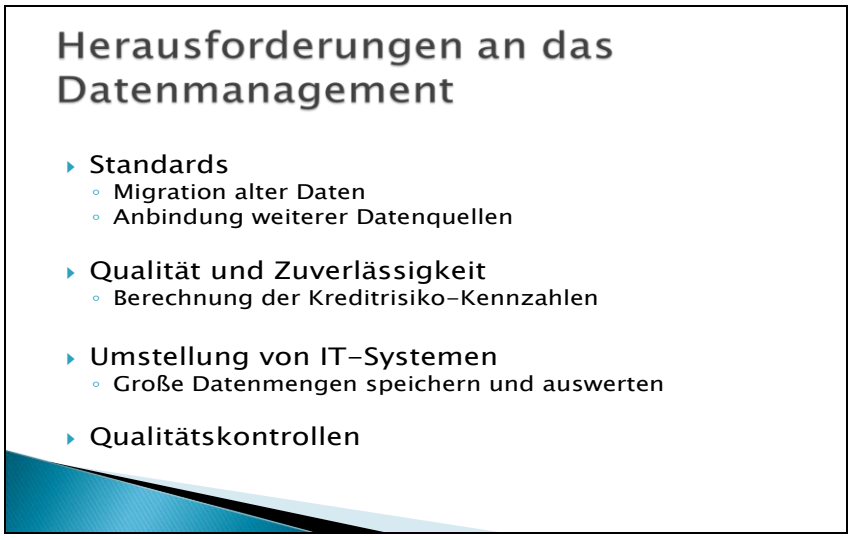
1974: Zusammenbruch Herrstatt-Bank
- Devisenspekulationen

1988: Eigenkapitalvereinbarung „Basel I“
- Kreditvergabepraxis



Basel II: nur Mindestkapital basierend auf Kredit- und Marktrisiken
Marktdisziplin: Verhalten, Öffentlichkeit über Kapital & Risiko zu informieren -> günstige Bedingung bei Beschaffung Fremdkapitals

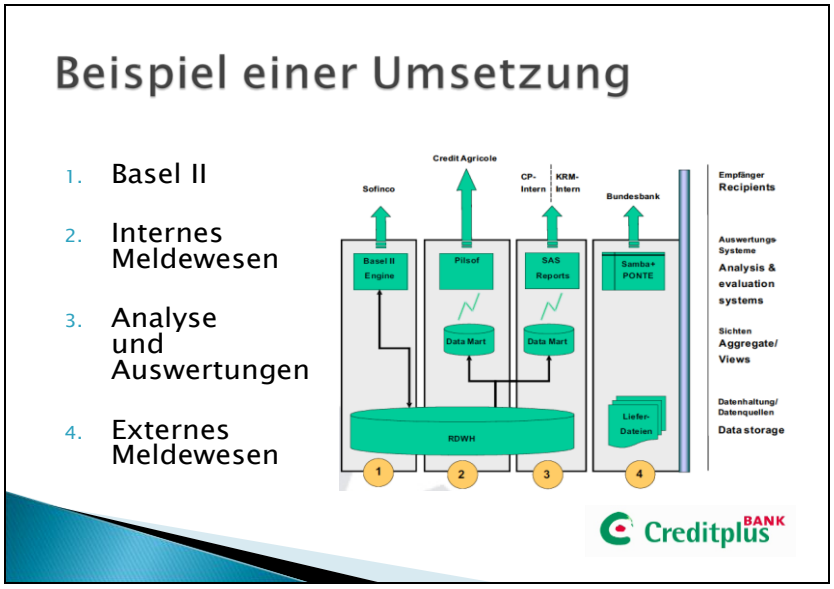
http://www.bundesbank.de/bankenaufsicht/bankenaufsicht_basel.php



DM strategy: Risk International
http://db.riskwaters.com/data/Risk__free_article_/basel.pdf

PD: Ausfallwahrscheinlichkeit, Verlustquote bei Ausfall, Höhe bei Ausfall -> erwarteter Verlust

http://www.it-observer.com/data-management-challenges-basel-ii-readiness.html
http://www.facebook.com/topic.php?uid=25192258947&topic=5725&_fb_noscript=1



CreditBank Plus AG, Stuttgart
www.information-works.de

http://www.finextra.com/community/fullblog.aspx?blogid=4988
frei verfügbare Anlagen hoher Qualität halten, welche auch in Krisenzeiten verkäuflich, Echtzeit -> data quality challenge

http://www.information-management.com/news/data_risk_management_Basel-10018723-1.html
http://www.pwc.lu/en/risk-management/docs/pwc-basel-III-a-risk-management-perspective.pdf

Eine Präsentation zu RFID & DWH sehen Sie hier:

R F I D

Radio Frequency Identifikation

Stefan Baudy, Max Nagel, Andreas Bitzer

Agenda

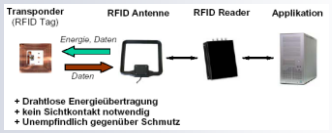
- Was ist RFID
- Anwendungsgebiete
- RFID & Data-Warehouse
- Ausblick

Was ist R F I D

- Kontaktlose Kommunikation über elektromagnetische Wellen
- Sillicon-Chip mit gespeicherter ID
- Abruf von Lesegerät über Aussenden von Wellen
- Chip sendet ID zurück
- Empfänger leitet Information weiter

Was ist RFID → Anwendungsgebiete → RFID & Data Warehouse → Ausblick

R F I D - Funktionalität



Was ist RFID → Anwendungsgebiete → RFID & Data Warehouse → Ausblick

RFID & Data-Warehouse

- Anforderungen an ein DWH
  - hohe Anzahl gleichzeitiger Transaktionen
  - extrem hohe Datenmengen
  - kurze Antwortzeiten
- Edge-Computing
- Dezentale Speicherung der Daten

Was ist RFID → Anwendungsgebiete → **RFID & Data Warehouse** → Ausblick

Anwendungsgebiete RFID

- Barcode: Ersatz, Erweiterung
  - Inventarüberwachung
  - Automatische Lagersysteme
- Sicherheitssysteme
  - Zugangskontrolle
  - Diebstahlschutz
  - Gepäckkontrolle

Was ist RFID → **Anwendungsgebiete** → RFID & Data Warehouse → Ausblick

RFID & Data-Warehouse

- Anforderungen an ein DWH
  - hohe Anzahl gleichzeitiger Transaktionen
  - extrem hohe Datenmengen
  - kurze Antwortzeiten
- Edge-Computing
- Dezentale Speicherung der Daten

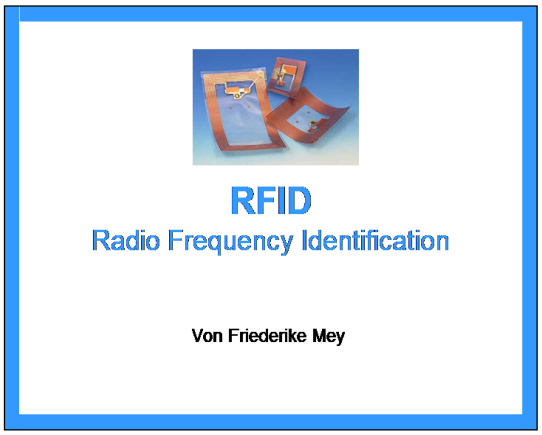
Was ist RFID → Anwendungsgebiete → **RFID & Data Warehouse** → Ausblick

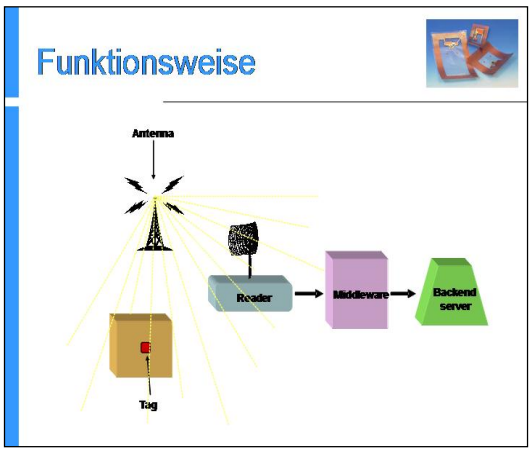
Ausblick

- Standards?
- Kosten vs. Nutzen (Barcodeersatz)
- Nutzen vs. Ausspionieren d. Kunden
- Höchst politisches Thema

Was ist RFID → Anwendungsgebiete → RFID & Data Warehouse → **Ausblick**

Eine weitere Lösung:





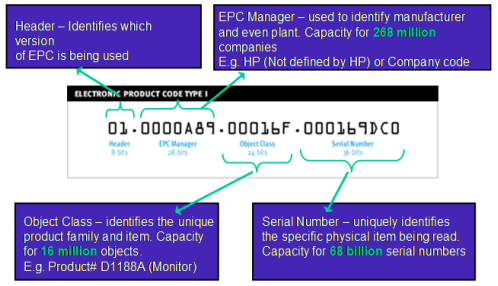
Herausforderungen

- Probleme bei speziellen Materialien (Flüssigkeiten, Metalle)
- Frequenzbereich
- Hohe Kosten
- Datenschutz

EPC Global

- Organisation um Standarts zu setzen
- Beispiele:
  - Electronic Product Code (EPC)

EPC Global



Komponenten

Middleware

- Säubert und filtert die Daten

Printer

- Druckt die Tags und schreibt Daten auf den Chip

EPC Global

- Organisation um Standarts zu setzen
- Beispiele:
  - Electronic Product Code (EPC)
  - Savant
  - Object Name Service (ONS)
  - Physical Markup Language (PML)

Projekte

- Metro Future Store in Rheinberg
  - RFID Innovation Center in Neuss

METRO Group
Future Store Initiative

RFID & Data Warehouse

- Riesige Datenmengen
- Hohe Anzahl gleichzeitiger Transaktionen
- Aufbereitung und Bereitstellung der Daten notwendig
- Kurze Antwortzeiten

Quellen

- Internet
  - www.future-store.org
  - www.hporaclectc.com
  - www.oracle.de
  - www.epcglobal.de
- Buch
  - RFID Handbuch von Klaus Finkenzeller

Exercises (+solutions) to BA Lesson DWH&DM – Chapter 2

Exercise 2.1:

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 hint: Use a table of the following form:

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

Solution:

Datawarehousing 5. Semester, IT00PMB, Marcel Petz
Projekt: Comparison of different DWH Architectures
Dokument: Comparison Matrix, 25.10.02
Seite: 1/ 1

Criteria	Datawarehouse only	Datamarts only	Datawarehouse & Datamarts	"Virtual" Datawarehouse	Description
Costs of Implementation	+	+	++	-	Costs of implementation describes the complexity of implementing a datawarehouse and its components. Complexity depends on general architecture (e.g. networked environment, database based, amount of hardware needed,...) of the datawarehouse.
Costs of administration	+	-	0	(--)	Costs of administration describes the costs for maintaining and run the datawarehouse.
Average data age	0	0	-	++	How old is the information presented to the frontend users of a datawarehouse
Performance		+	+	--	The Performance of the whole system and all involved components. Which architecture has a good performance.
Flexibility	-	0	+	++	Flexibility describes the ability of changing datastructures or parts of the datawarehouse programs. High flexibility means, that it is easy to make changes to the data structures and client applications.
Implementation-time	-	-	--	+	Describes the time-to-market. The time from begin of implementation until the system is activated for production use. Positive values mean short implementation time.
Data Consistency	+	+	+	--	
Quality of informations					Quality of data stored in the datawarehouse depend strongly on the quality of the ETL process. It is not possible to draw general conclusions at this point
History	++	++	++	--	The ability to look at certain points or periods of time in the past und gather information about it.

Implementation costs
The implementation of a Data Warehouse with Data Marts is the most expensive solution, because it is necessary to build the system including connections between Data Warehouse and its Data Marts.
It is also necessary to build a second ETL which manages the preparation of data for the Data Marts.
In case of implementing Data Marts or a Data Warehouse only, the ETL is only implemented once. The costs may be almost the same in building one of these systems. The Data Marts only require a little more hardware and network connections to the data sources. But due to the fact, that building the ETL is the most expensive part, these costs may be relative low.
The virtual Data Warehouse may have the lowest implementation costs, because e.g. existing applications and infrastructure is used.

Administration costs
The **Data Warehouse only solution** offers the best effort in minimizing the administration costs, due to the centralized design of the system. In this solution it is only necessary to manage a central system. Normally the client management is no problem, if using web technology or a centralized
client deployment, which should be a standard in all mid-size to big enterprises. A central Backup can cover the whole data of the Data Warehouse.
The solution with **Data Marts only** are more expensive, because of its decentralized design. There are higher costs in cases of product updates or maintaining the online connections, you also have to backup each Data Mart for itself, depending on his physical location.
Also the process of filling a single Data Mart is critical. Errors during update may cause loss of data. In case of an error during an update, the system administration must react at once.
Data Marts with a central Data Warehouse are more efficient, because all necessary data is stored in a single place. When an error during an update of a Data Mart occurs, this is normally no problem, because the data is not lost and can be recovered directly from the Data Warehouse. It may also be possible to recover a whole Data Mart out of the Data Warehouse.
Virtual Data Warehouses administration costs depend on the quality of the implementation. Problems with connections to the online data sources may cause user to ask for support, even if the problem was caused by a broken online connection or a failure in the online data source. End-users may not be able to realize whether the data source or the application on their computer cause a problem.

Average data age
The virtual Data Warehouse represents the most actual data, because the application directly connects to the data sources and fetches its information online. The retrieved information is always up to date.
Information provided by Data Mart only or Data Warehouse only solutions are collected to specific time. Generally each day by night. These times can vary from hourly to monthly or even longer. The selected period depends on the cost of the process retrieving and checking the information.
A solution with one central Data Warehouse and additional Data Marts houses less actual data then Data Warehouse only. The data of the Data Warehouse must be converted and copied to the Data Marts, which is time consuming.

Performance
A virtual Data Warehouse has the poorest performance all over. All data is retrieved during runtime directly from the data sources. Before data can be used, it must be converted for presentation. Therefore a huge amount of time is spent by retrieval and converting of data.

The Data Marts host information, which are already optimized for the client applications. All data s stored in an optimal state in the database. Special indexes in the databases speed up information retrieval.

Implementation Time

The implementation of a Data Warehouse with its Data Marts takes the longest time, because complex networks and transformations must be created. Creating Data Warehouse only or Data Marts only should take almost the same amount of time. Most time is normally spent on creating the ETL (about 80%), so the differences between Data Warehouse only and Data Marts only should not differ much.

Implementing a Virtual Data Warehouse can be done very fast because of its simple structure. It is not necessary to build a central database with all connectors.

Data Consistency

When using Data Warehouse or Data Mart technology a maximum consistency of data is achieved.

All provided information is checked for validity and consistency. A virtual Data Warehouse may have problems with data consistency because all data is retrieved at runtime. When data organization on sources changes, the consistency of new data may be consistent, but older data may not be represented in its current model.

Flexibility

The highest flexibility has a virtual data warehouse. It is possible to change the data preparation process very easy because only the clients are directly involved. There are nearly no components, which depend on each other.

In Data Warehouse only solution flexibility is poor, because there may exist different types of clients that depend on the data model of the Data Warehouse. If it would be necessary to change a particular part of the data model intensive testing for compatibility with existing applications must be done, or even the client applications have to be updated.

A solution with Data Marts, with or without a central Data Warehouse has medium flexibility due that client applications normally uses Data Marts as their point of information. In case of a change in the central Data Warehouse or the data sources, it is only necessary to update the process of filling the Data Marts.

In case of change in the Data Marts only the depending client applications are involved and not all client applications.

Data Consistency

Data consistency is poor in a virtual Data Warehouse. But it also depends on the quality of the process, which gathers information from the sources.

Data Warehouses and Data Marts have very good data consistency because the information stored in their databases have been checked during the ETL process.

Quality of information

The quality of information hardly depends on the quality of the data population process (ETL process) and how good the information are processed and filtered before stored in the Data Warehouse or presented to a user. Therefore it is not possible to give a concrete statement.

History

A virtual Data Warehouse has no history at all, because the values or information are retrieved at runtime. In this architecture it is not possible to store a history because no central database is present.

The other architectures provide a central point to store this information. The history provides a basis for analyzing business process and their efforts, because it is possible to compare actual information with information of the past.

Exercises (+Solutions) to BA Lesson DWH&DM – Chapter 3

Exercise 3.1:

Build 4 groups .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 …) (3.1.1)
2.Relational DBMS (3.1.2)
3.SQL Basics (3.1.3)
4. Normalization (3.1.4)

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 to 3.1.1 - Non-relational databases (IMS, VSAM …):

Datenmodell: Die zur Beschreibung von Daten und deren Beziehungen untereinander auf logischer Ebene zur Verfügung stehenden Datenstrukturen bezeichnet man zusammenfassend als **Datenmodell**.

Dient zur formalen Beschreibung des konzeptionellen (bzw. logischen) Schemas und der externen Schemata mit Hilfe entsprechender Datendefinitionssprachen.

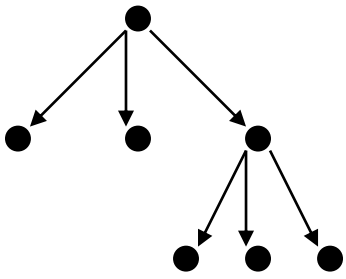
1. Das Hierarchische Datenmodell – HDM

- primär können nur hierarchisch-baumartige Beziehungen von Objekttypen dargestellt werden.
- Reale Beziehungen sind oft von netzwerkartiger Struktur, sodass Erweiterungen des Datenmodells erforderlich sind => z.B. bei IMS

Strukturelemente:

- Objekttypen
- Hierarchische unbenannte Beziehungen (Kanten haben keine Bezeichnungen)

Ergebnis: Baumstruktur



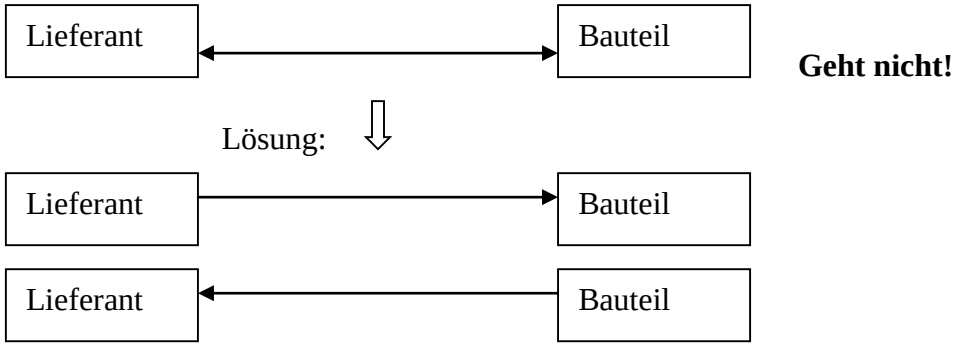
Wurzelbaum-Typ (Hierarchie-Typ) stellt Objekttypen und deren Beziehungen zueinander dar.

Hierarchische Datenbank ist eine Menge von disjunkten Wurzelbaum-(Hierarchie)-Typen.

Im hierarchischen Modell ist jedes Wurzel-Objekt über einen Primärschlüssel erreichbar, alle anderen Objekte gemäß der hierarchischen Ordnung. Der Zugriff auf Datenobjekte erfolgt also entlang den logischen Zugriffspfaden (durch Kanten dargestellt). Dies setzt seitens des Anwenders eine genaue Kenntnis der DB-Struktur voraus und bedingt eine prozedurale Beschreibung des Zugriffs. Man spricht bildlich von einem **Navigieren** durch die Datenbank.

Darstellung von Strukturen im HDM:

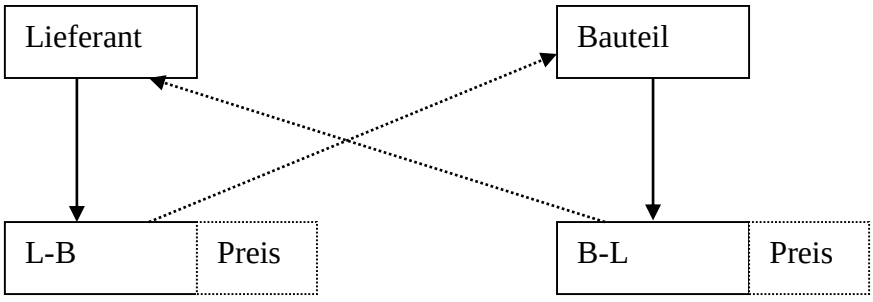
In einem (strengen) HDM können netzwerkartige Strukturen nicht dargestellt werden. Eine n:m Beziehung, wie z.B. die zwischen Bauteilen und Lieferanten, kann nur durch zwei getrennt Hierarchie-Typen dargestellt werden
=> Redundanz!
Bsp:



Problem: Lieferanten und Bauteile sind mehrfach gespeichert.

Problemlösung: **Pairing**

Abweichend vom strengen HDM werden zusätzliche logische Zugriffe eingeführt, damit n:m Beziehungen dargestellt werden können.



Lieferanten und Bauteile sind nun nur einfach vorhanden.

Problem: Preise, die als Attribute bei zusätzlich eingeführten Objekttypen B-L und L-B gespeichert werden, sind immer noch redundant.

IMS Information Management System:
- kennt keine genaue Unterscheidung zwischen den 3 Schemas (extern, konzeptionell. intern)

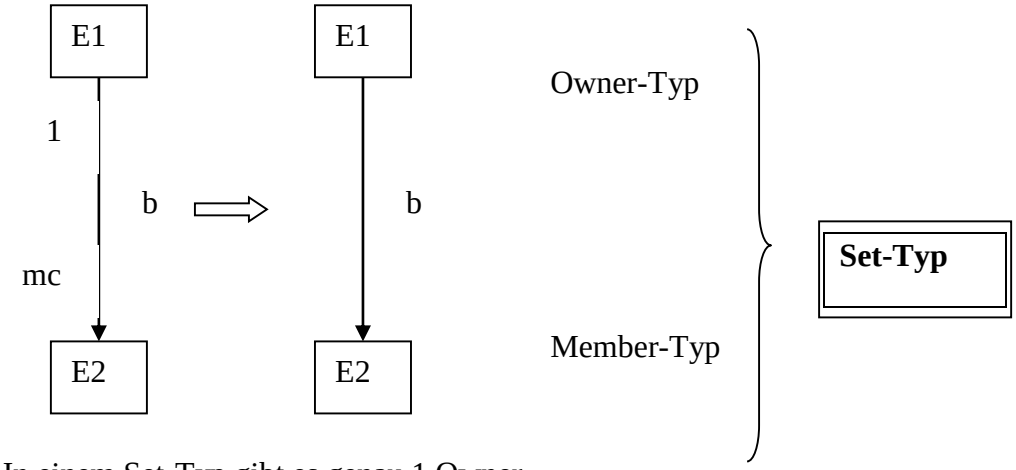
- logische Datenmodellierung und physikalische Datenorganisation ineinander verwoben
- Datendefinition erfolgt mit Hilfe der Sprache DL / I
- hierarchische Strukturen können über logische Zeiger miteinander verkettet werden
- Anwender-Sichten können definiert werden

2. Das Netzwerkmodell

- „Erweiterung“ des HDM um netzwerkartige Beziehungen

Strukturelemente:

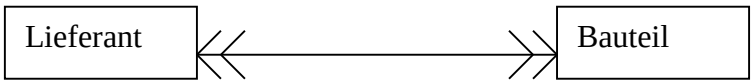
- Objekttypen
- hierarchische Beziehungen (1:mc), die als Set-Typen bezeichnet werden



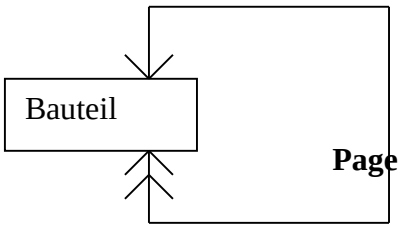
In einem Set-Typ gibt es genau 1 Owner.
1 Owner kann viele Members haben (0 ..*)
1 Owner kann Member sein (in einem anderen Set-Typ), 1 Member kann auch Owner sein

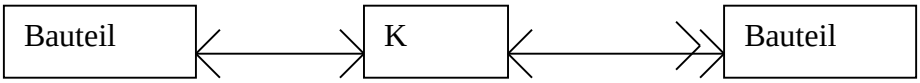
Darstellung von Strukturen im NDM:

1:m ist trivial
m:n durch K ett-Objekt-Typ (link entity type)



Objekt-Typen können auch mit sich selbst in Beziehung stehen, z.B. kann ein Bauteil ein Bauteil eines anderen Bauteils sein.





VSAM: Virtual Storage Access Method

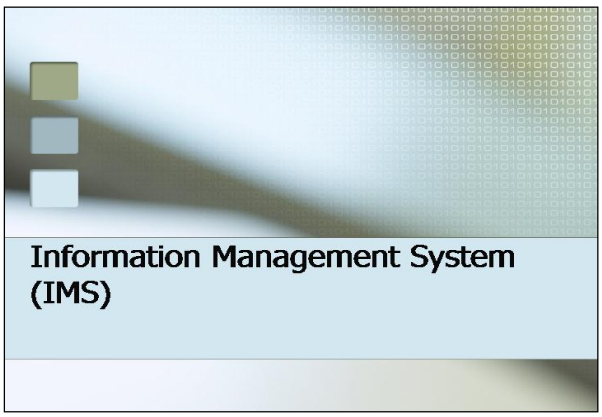
Virtuel: Hardware-Unabhängigkeit, d.h. bei der Dateiorganisation wird primär kein Bezug auf die physische Speicherorganisation (z.B. Zylinder und Spuren der Magnetplatte) genommen.

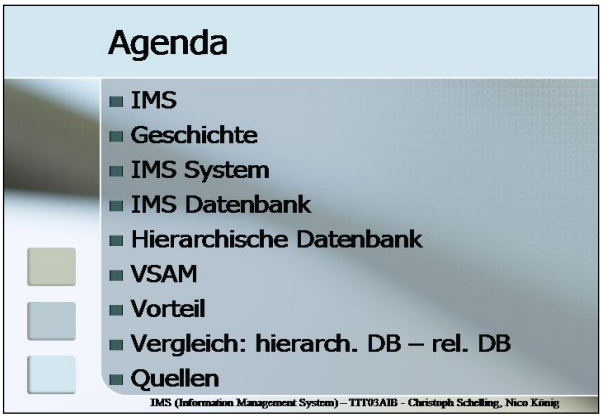
Die auch den B- und B^+-Bäumen zugrunde liegenden Prinzipien, nämlich

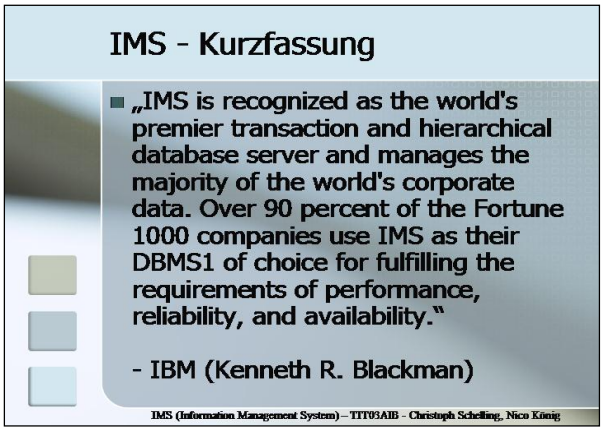
- in Speicherbereichen fester Größe (Knoten) verteilten freien Speicherplatz zur Aufnahme einzufügender Datenobjekte vorzusehen
- durch „Zell-Teilung“ (cellular splitting) neuen Speicherplatz zu schaffe, falls der Platz beim Einfügen nicht ausreicht,

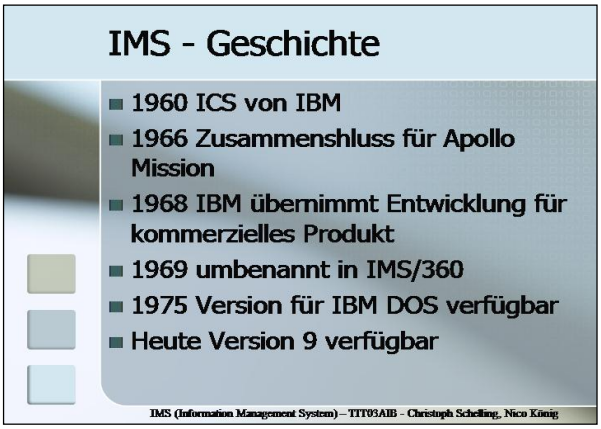
werden hier auch auf die Speicherung der Datensätze selbst (Primärdaten) angewendet und als Index ein B^+-Baum verwendet, dessen Blätter gekettet sind, so dass eine logisch fortlaufende Verarbeitung nach aufsteigenden und absteigenden Schlüsselwerten und auch der (quasi-) direkte Zugriff möglich ist.

Eine weitere Lösung (2. Lösung):

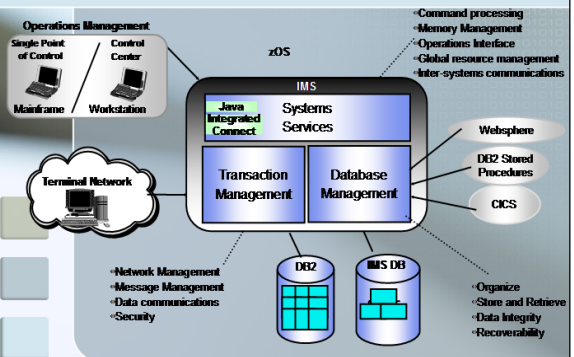








IMS - heute

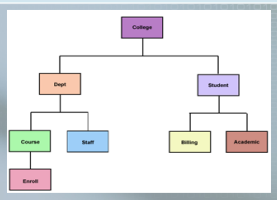


IMS Datenbank

- Hierarchische Datenbank
- Abfragesprache DL/I
- Access Methods(System, IMS)
- Control Blocks (DBD, PSB, ACB)
- Data Communication
- Secondary Indexes
- Logical Relationships

Hierarchische Datenbank

- Beispiel Struktur:

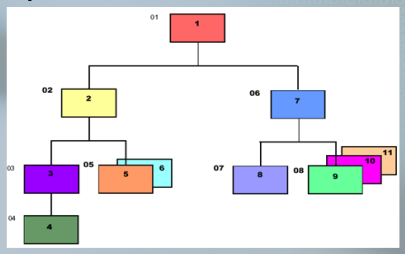


- Segment:

Prefix			Data		
Segment code	Delete byte	counters and pointers	Size field	seq. (key) field	data length varies,
1 byte	1 byte	4 bytes per element	2 bytes		based on a minimum and maximum size

Hierarchische Datenbank

- Suchpfad:



01... = Typ, 1... = Reihenfolge der Suche

VSAM – Virtual Storage Access Method

- Zugriffsmethode auf Dateien in auf IBM Großrechnersystemen (z/OS)
- keine Rücksicht auf physikalische Eigenschaften der Speichermedien mehr nötig
- Speicherung in Cluster

VSAM – Virtual Storage Access Method

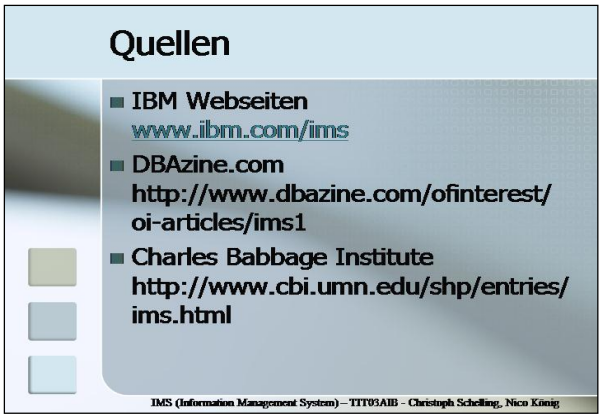
- Clusterformen
  - KSDS (Key sequential DataSet):
    - Datenzugriff über einen Index oder sequentiell
  - ESDS (Entry sequential DataSet):
    - sequentiell Zugriff
  - RRDS (Relative Record DataSet):
    - Der Zugriff mit Hilfe von logischen Satznummern

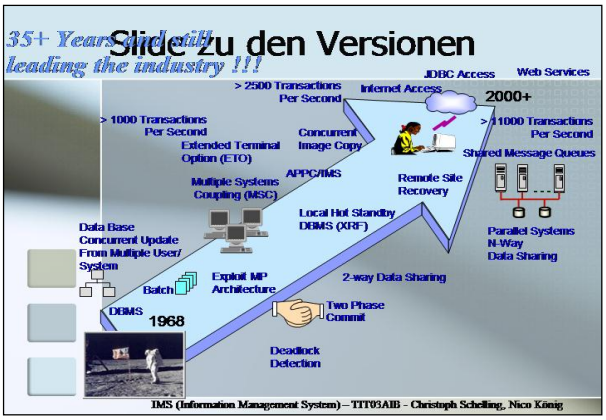
VSAM – Virtual Storage Access Method

- Master-Catalog:
  - jedes System, das VSAM nutzt hat genau einen Master Katalog
  - enthält Informationen zu Datensätzen und Strukturen, um die VSAM Operationen zu steuern
- User Catalog:
  - enthält Einträge zu Anwendungsspezifischen Daten
  - Informationen zur Beschreibung des User Catalog sind im Master Catalog gespeichert

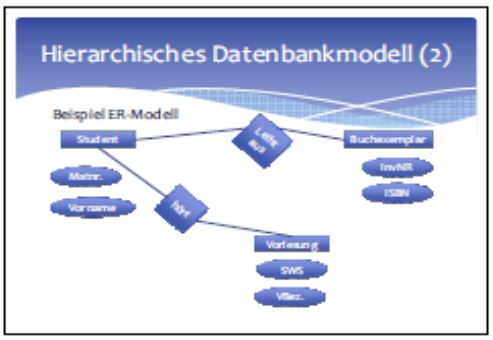
Vergleich hier. DB - rel.DB

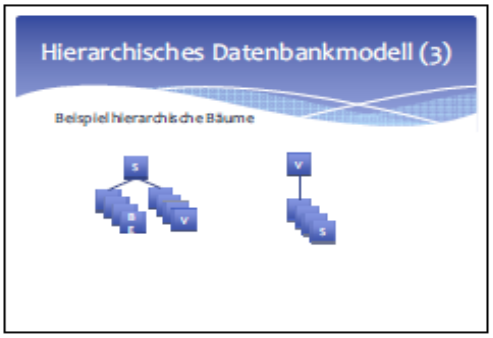
- Vorteile:
  - Leistungsstärker (keine techn. Metadaten Verwaltung)
  - Sehr große Datenmengen mit vielen Transaktionen effizienter verwalten
  - Komplexe Abfragen schnell abrufbar
- Nachteile:
  - Hohe Komplexität der Entwicklung
  - Lange Entwicklungsdauer

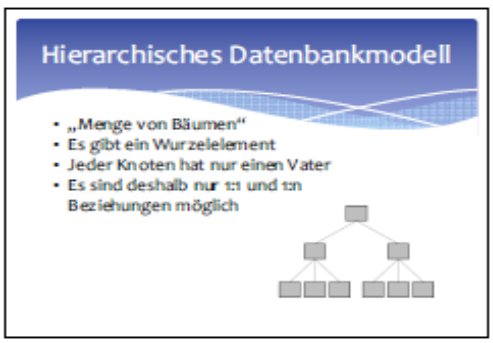


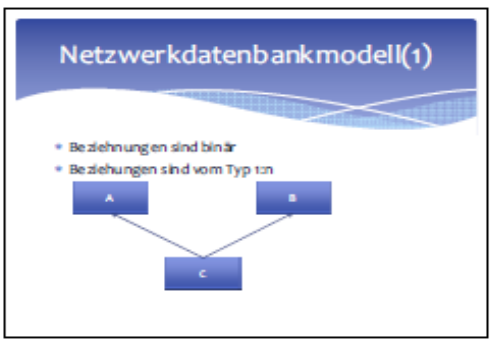


Eine dritte Lösung zu 3.1.1 finden Sie hier:









Netzwerkdatenbankmodell(2)

VSAM

- Master-Catalog:
  - jedes System, das VSAM verwendet, hat genau einen
  - Informationen zu Datensätzen und Strukturen, um VSAM-Operationen zu steuern
- User-Catalog:
  - Einträge zu Anwendungsspezifischen Daten
  - Informationen zur Beschreibung dieses sind in Master-Catalog

VSAM

- virtual storage access method
- Zugriffsmethode auf Dateien auf IBM Großrechnern, Speichermethode der Daten in DB2
- keine Rücksicht auf physik. Eigenschaften eines Speichermediums
- technisch: Einträge in einem Katalog (VSAM Volume Data Set) mit Metadaten
- mindestens eine physischen Datei
- →Cluster

VSAM

- Vorteile:
  - sehr große Datenmengen mit vielen Transaktionen verwaltbar
  - komplexe Abfragen schnell aufrufbar
- Nachteile:
  - Hohe Komplexität der Entwicklung
  - Lange Entwicklungsdauer

VSAM

- Arten von Clustern:
  - KSDS (*key sequential dataset*): Zugriff auf Daten über Index (in eigenem Dataset gespeichert)
  - ESDS (*entry sequential dataset*): sequentieller Zugriff auf Daten
  - RRDS (*relative record dataset*): Zugriff mittels logischer Satznummern
  - LDS (*linear dataset*): Daten werden als unstrukturierter Bytestrom verwaltet

mongoDB

- dokumentorientiertes Speichern (JSON)
- Indizes werden unterstützt
- Replikation →hohe Verfügbarkeit
- horizontale Fragmentierung

Project Voldemort

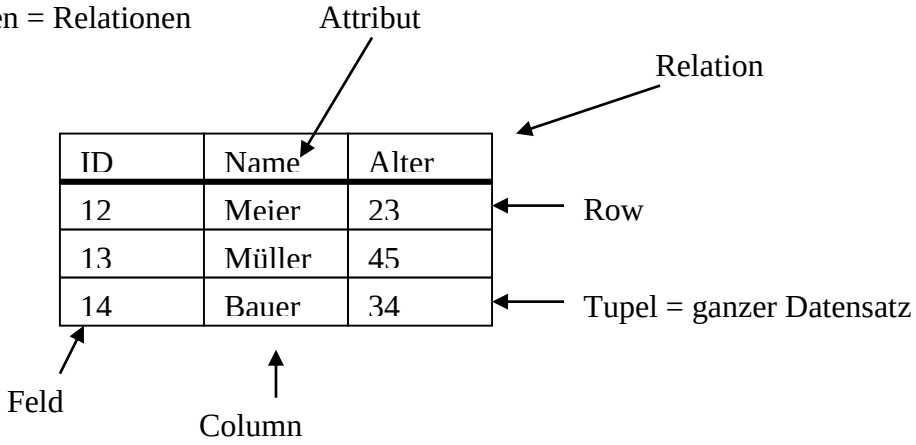
- key-value storage system
- Daten werden automatisch über mehrere Server repliziert / verteilt
- Daten werden versioniert um Integrität zu gewährleisten
- 3 Operationen: value=get(key), put(key,value),delete(key)
- effiziente Operationen→vorhersehbare Performance

Solution to 3.1.2 - Relationale Datenbanken:

→ Relation = Beziehung/Abhängigkeit von Objekten und Daten zueinander

→ Definition:
- rel. DB-Modell 1970 von Codd
- Datenspeicherung in Tabellen (Relationen) mit einer festen Anzahl an Spalten und einer flexiblen Anzahl an Zeilen
- Durch das Verteilen der Informationen auf einzelne Tabellen werden Redundanzen vermieden.
- Mit Schlüsselfeldern können Verknüpfungen zw. den Tabellen erstellt werden.

→ Tabellen = Relationen



→ Eine Menge von miteinander verbundenen Relationen bildet eine Datenbank.

→ In einer Tabelle gibt es keine zwei Tupel, die für alle Attribute die gleichen Werte haben.

→ Schlüssel = identifizierende Attributmenge

→ Primärschlüssel
= eine Spalte der Tabelle, durch deren Werte jeder Datensatz der Tabelle eindeutig identifiziert wird.
Der Wert eines Primärschlüsselfeldes einer Tabelle darf nicht doppelt vorkommen.
Jede Tabelle kann nur einen Primärschlüssel haben.
Er kann sich aus mehreren Datenfeldern zusammensetzen und darf nicht leer sein.

→ Fremdschlüssel
= eine Spalte einer Tabelle, deren Werte auf den Primärschlüssel einer anderen Tabelle verweisen.
Eine Tabelle kann mehrere Fremdschlüssel enthalten.
Er kann aus mehreren Feldern der Tabelle bestehen, er kann leer sein.
Für jeden Wert eines Fremdschlüssels muss es einen entsprechenden Wert im Primärschlüssel der korrespondierenden Tabelle geben (Integrität)

→ Basisoperationen: (siehe SQL-Anweisungen)
- Selektion
- Verbund
- Projektion

→ Weitere Regeln der relationalen Datenbank:
- Transaktionen müssen entweder vollständig durchgeführt werden oder, bei einem Abbruch, vollständig zurückgesetzt werden.
- Der Zugriff auf die Daten durch den Benutzer muss unabhängig davon sein, wie die Daten gespeichert wurden oder wie physikalisch auf sie zugegriffen wird.
- Ändert der Datenbankverwalter die physikalische Struktur, darf der Anwender davon nichts mitbekommen.

Solution to 3.1.3- SQL Basics:

Compare standard books about SQL language

Solution to 3.1.4 - Normalization:

Ziel von Normalformen

- Update-Anomalien innerhalb einer Relation vermeiden
- Update-Anomalien: Redundanzen in Datenbanken, die einerseits unnötigen Speicherplatz verbrauchen und andererseits dazu führen, dass sich Änderungsoperationen nur schwer umsetzen lassen (Änderung bei allen Vorkommen einer Information)
- Ziel: Redundanzen entfernen, die aufgrund von funktionalen Abhängigkeiten innerhalb einer Relation entstehen

Abhängigkeiten

a) funktional abhängig
zu einer Attributkombination von A gibt es genau eine Attributkombination von B
B ist funktional abhängig von A: A -> B

b) voll funktional abhängig
A und B als Attributkombination der gleichen Relation R
B ist voll funktional abhängig von A, wenn es von der gesamten Attributkombination von A funktional abhängt, aber nicht schon von einem Teil: A => B

c) transitiv abhängig
B ist abhängig von A und C ist abhängig von B: A -> B -> C
C darf dabei nicht Schlüsselattribut sein und nicht in B vorkommen

Anomalien:

Prüfungsgeschehen

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**

a) Einfüge-Anomalien
Wo fügt man in dieser Relation einen Studenten ein, der noch nicht an einer Prüfung teilgenommen hat?

b) Lösch-Anomalien
Mit Löschung des Studenten Pitt, geht auch die Information über den Dekan vom Dachbereich BWL verloren.

c) Änderungs-Anomalien
Zieht ein Student um, der an mehreren Prüfungen teilgenommen hat, so muß die Adressänderung in mehreren Tupeln vollzogen werden

Erste Normalform

- Nur atomare Attribute, also Elemente von Standard-Datentypen und nicht Listen, Tabellen oder ähnliche komplexe Strukturen

Prüfungsgeschehen

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**

- Bsp. enthält eine weitere Relation
- 1. Lösung: jede Zeile um die ersten drei Attribute erweitern, dann entstehen aber Redundanzen
- 2. Lösung: Auslagerung in eine neue Tabelle Prüfung

PNR	**Fach**	**Prüfer**
3	**Elektronik**	**Richter**
4	**Informatik**	**Schwinn**
5	**TMS**	**Müller**

Prüfling

PNR	**MATNR**	Name	Geb	Adr	Fachbereich	Dekan	Note
3	**123456**	**Meier**	**010203**	**Weg 1**	**Informatik**	**Wutz**	**1**
3	**124538**	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**	**2**
4	**245633**	**Kunz**	**021279**	**Gas. 2**	**Informatik**	**Wutz**	**1**
4	**124538**	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**	**1**
5	**856214**	**Schmidt**	**120178**	**Str 2**	**Informatik**	**Wutz**	**3**
5	**369852**	**Pitt**	**140677**	**Gas. 1**	**BWL**	**Butz**	**1**

- Beide Relationen sind nun in 1. NF

Zweite Normalform

- Ziel: aufgrund von funktionalen Abhängigkeiten Redundanzen entdecken
- Erlaubt keine partiellen Abhängigkeiten zwischen Schlüsseln des Relationenschemas und weiteren Attributen (jedes Nicht-Primärattribut muss also voll funktional abhängig sein von jedem Schlüssleattribut der Relation)

Prüfling

PNR	**MATNR**	Name	Geb	Adr	Fachbereich	Dekan	Note
3	**123456**	**Meier**	**010203**	**Weg 1**	**Informatik**	**Wutz**	**1**
3	**124538**	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**	**2**
4	**245633**	**Kunz**	**021279**	**Gas. 2**	**Informatik**	**Wutz**	**1**
4	**124538**	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**	**1**
5	**856214**	**Schmidt**	**120178**	**Str 2**	**Informatik**	**Wutz**	**3**
5	**369852**	**Pitt**	**140677**	**Gas. 1**	**BWL**	**Butz**	**1**

- Erkennbar: Daten des Studenten (Name, Geb, Adr, Fachbereich, Dekan) hängen nur von MATNR ab und nicht von PNR, ist somit nicht voll funktional abhängig
- Erzeugung der zweiten Normalform durch Elimination der rechten Seite der partiellen Abhängigkeit und Kopie der linken Seite

Student

MATNR	Name	Geb	Adr	Fachbereich	Dekan
123456	**Meier**	**010203**	**Weg 1**	**Informatik**	**Wutz**
124538	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**
245633	**Kunz**	**021279**	**Gas. 2**	**Informatik**	**Wutz**
124538	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**
856214	**Schmidt**	**120178**	**Str 2**	**Informatik**	**Wutz**
369852	**Pitt**	**140677**	**Gas. 1**	**BWL**	**Butz**

Prüfungsergebnis

PNR	**MATNR**	Note
3	**123456**	**1**
3	**124538**	**2**
4	**245633**	**1**
4	**124538**	**1**
5	**856214**	**3**
5	**369852**	**1**

- Eine Relation R ist in 2. NF, wenn sie in 1.NF ist und jedes Nicht-Primärattribut von R voll von jedem Schlüssel in R abhängt (also keine Attribut des Schlüssels unwesentlich ist)

- Problem der Anomalien noch nicht beseitigt
  Einfüge-A.: Fachbereichsdaten nicht ohne eingeschriebenen Studenten speicherbar
  Lösch-A: Fachbereichsdaten verschwinden mit Löschen des letzten Studenten
  Änderungs-A: Wechsel des Dekans muss an mehreren Stelle vollzogen werden

Dritte Normalform

- 3. NF: keine transitiven Abhängigkeiten

Student

MATNR	Name	Geb	Adr	Fachbereich	Dekan
123456	**Meier**	**010203**	**Weg 1**	**Informatik**	**Wutz**
124538	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**
245633	**Kunz**	**021279**	**Gas. 2**	**Informatik**	**Wutz**
124538	**Schulz**	**050678**	**Str 1**	**Informatik**	**Wutz**
856214	**Schmidt**	**120178**	**Str 2**	**Informatik**	**Wutz**
369852	**Pitt**	**140677**	**Gas. 1**	**BWL**	**Butz**

- transitive Abhängigkeit: Dekan ist von Fachbereich abhängig, da es zu jedem Fachbereich genau einen Dekan gibt (demnach ist Dekan transitiv abhängig von MATNR)
- Eliminieren von transitiven Abhängigkeiten: Auslagerung der abhängigen Attribute in eine neue Relation

Fachbereich

Fachbereich	Dekan
Informatik	**Wutz**
BWL	**Butz**

Student

MATNR	Name	Geb	Adr	Fachbereich
123456	**Meier**	**010203**	**Weg 1**	**Informatik**
124538	**Schulz**	**050678**	**Str 1**	**Informatik**
245633	**Kunz**	**021279**	**Gas. 2**	**Informatik**
124538	**Schulz**	**050678**	**Str 1**	**Informatik**
856214	**Schmidt**	**120178**	**Str 2**	**Informatik**
369852	**Pitt**	**140677**	**Gas. 1**	**BWL**

Exercise 3.2:

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 $/Unit=Funct (Item#)

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

Solution to 3.2:

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

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

.....

Exercises (+solutions) to BA Lesson DWH&DM – Chapter 4

Exercise 4.1: given the two tables (20 points + 10 points)

Airport:

FID	*Name*
MUC	Muenchen
FRA	Frankfurt
HAN	Hannover
STU	Stuttgart

MAN	Mannheim
BER	Berlin

Flight:

Fno	***From***	***To***	***Time***
161	MUC	HAN	9:15
164	HAN	MUC	11:15
181	STU	MUC	10:30
185	MUC	FRA	6:10
193	MAH	BER	14:30

Define the right SQL such that :

1. you get a list of airports which have no incoming flights (no arrivals) (6 points)
2. create a report (view) Flights_To_Munich of all flights to Munich(arrival) with Flight-Number, Departure-Airport (full name) and Departure-Time as columns (6 points)
3. insert a new flight from BER to HAN at 17:30 with FNo 471 (4 points)
4. Change FlightTime of Fno=181 to 10:35 (4 points)

Optional (difficult) –10 points:

5. calculates the numbers of flights from (departures) for each airport

Solution:

Ad 1.:
select fid, name from airport
where fid **not in**
(**select distinct** to from flight)

Ad 2.:
create view Flights_to_Munich
as select f.Fno as Flight-Number, a.name as Departure-Airport, f.time as Departure-Time from flight f, airport a
where f.to='MUC' **and** a.fid=f.from

Ad3.:
insert into flight
values (471,'BER','HAN','17:30')

Ad4.:
update flight
set time = '10:35'
where Fno=181

Ad5 (optional):

```
select name as Departure_Airport, count (*) as Departure_Count
from airport, flight
where fid=from
group by name
union
select name as Departure_Airport, 0 as Departure_Count
from airport
where not exists (select * from flight where from=fid)
order by departure_count
```

Delivers the following result:

```
**********************************************************************
db2 => select name as Departure_ Airport, count (*) as Departure_Count from airpo
rt, flight where fid=from group by name union select name as Departure_Airpo
rt, 0 as Departure_Count from airport where not exists (select * from flight whe
re from=fid) order by departure_count

DEPARTURE_AIRPORT    DEPARTURE_COUNT
---------------------------   ----------------------------
Berlin                                0
Frankfurt                             0
Hannover                              1
Mannheim                              1
Stuttgart                             1
Muenchen                              2

  6 record(s) selected.
**********************************************************************
```

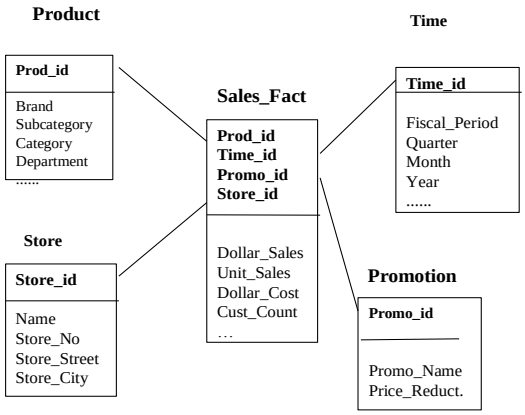
Here is also a **second solution** (which is shorter) and gives the same results as above by **Stefan Seufert**:

```
SELECT Name as Departure_ Airport, count (Flight.From) as  Departure_Count
FROM Airport LEFT OUTER JOIN Flight ON Airport.FID = Flight.From
GROUP BY Name
ORDER BY Departure_Count
```

The idea is, that count(Field) in contradiction to count(*) only count the fields which are not NULL. Since the attribute in the count function is from the flight table, only the flights which have departures are counted, all other get the 0 value.

Exercise 4.2:

Consider the following Star Schema:



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;
```

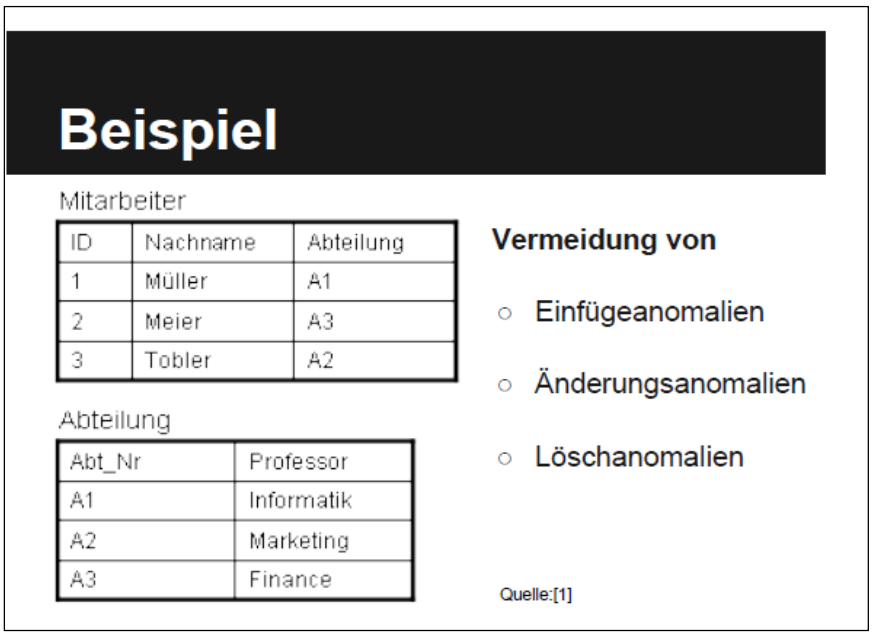
Exercise 4.3- Advanced Study about concepts in DWH:

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.

First SOLUTION:



Integrität beim Einfügen

Mitarbeiter

ID	Nachname	Abteilung
1	Müller	A1
2	Meier	A3
3	Tobler	A2

Abteilung

Abt_Nr	Professor
A1	Informatik
A2	Marketing
A3	Finance

Einfügen von
- ID: **4**
- Nachname: **Weber**
- Abteilung **A5**

in Tabelle Mitarbeiter

Quelle:[1]

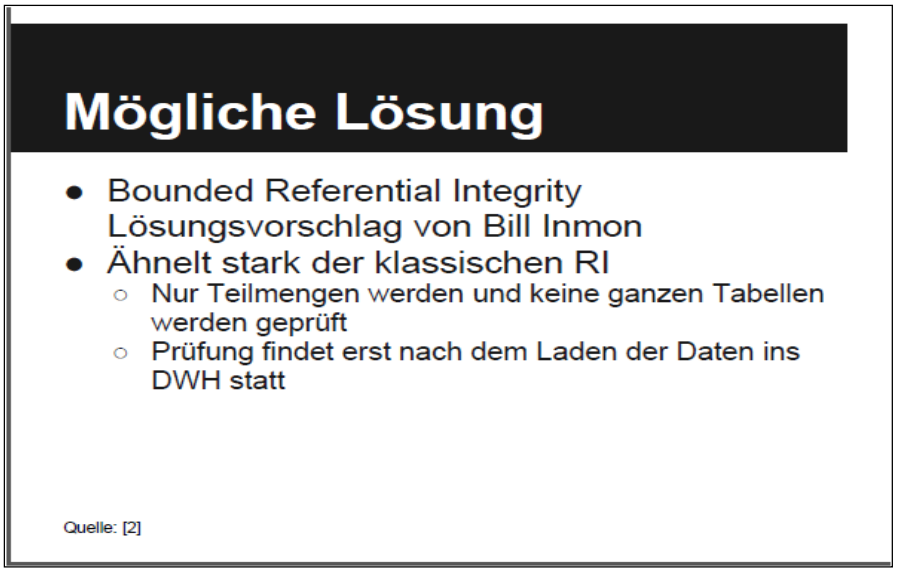
Gründe für RI im DWH

- Datenkonsistenz
- Änderungen müssen nur an einer Stelle durchgeführt werden.
- Integritätsprobleme werden verhindert

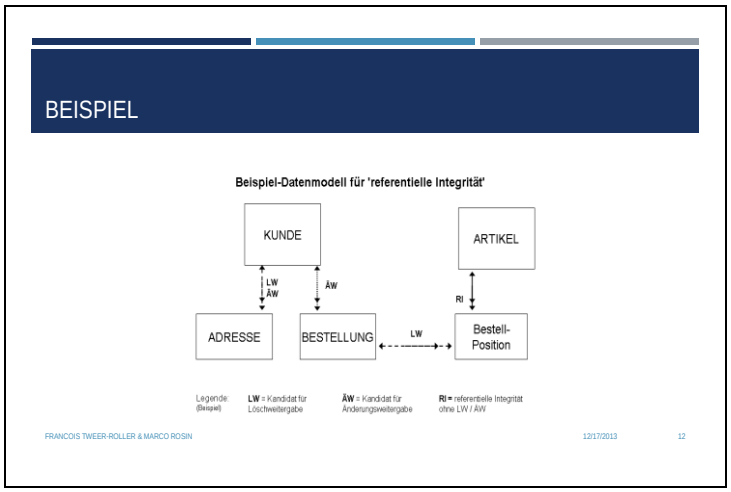
Gründe gegen RI im DWH

- Längere Zugriffszeiten / Viel Overhead
- Zu große Datenmengen für konventionelle RI-Methoden aus DBMS
- RI Prüfung wird meist vor dem Laden der Daten ins DWH durchgeführt
- Im DWH werden keine Updates durchgeführt

Quelle: [2]



Zweite SOLUTION:



Dritte Solution:

Definition

"Über referentielle Integrität werden in einem DBMS die Beziehungen zwischen Datenobjekten kontrolliert".

Vorteile

- Steigerung der Datenqualität: Referenzielle Integrität hilft Fehler zu vermeiden.
- Schnellere Entwicklung: Referenzielle Integrität muss nicht in jeder Applikation neu implementiert werden.
- Weniger Fehler: Einmal definierte referenzielle Integritätsbedingungen gelten für alle Applikationen der selben Datenbank
- Konsistentere Applikationen: Referenzielle Integrität ist für alle Applikationen, die auf die selbe Datenbank zugreifen gleich.

Nachteile

- Löschproblematik aufgrund von Integrität
- Temporäres außer Kraft setzen der RI für großen Datenimport.

Referenzielle Integrität in einem DWH

- Daten müssen im DWH nicht 100%ig konsistent sein.
- Durch Import von großen Datenmengen ist die Kontrolle der Integrität zu aufwendig

- Inkonsistente Daten können in keinen konsistenten Zustand gebracht werden.

Meiner Meinung nach ist die Realisierung von der referentiellen Integrität möglich, aber mit viel Aufwand und Kosten verbunden.

Exercises (+Solutions) to BA Lesson DWH&DM – Chapter 5

Exercise 5.1:

Compare ER Modelling (**ER**) with multidimensional data models (**MDDM**), like **STAR** or **SNOWFLAKE** schemas (see appendix page):
Compare in IBM Reedbook 'Data Modeling Techniques for DWH' (see DWH lesson homepage) Chapter6.3 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 --

SOLUTION:
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.
MDDM Model: Compare as examples the Star - and Snowflake schemas, which are explained in the next solution (5.2)

Exercise 5.2:

Compare MDDM Model schemas **STAR** and **SNOWFLAKE**
Compare in IBM Reedbook '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 (compare exercise 5.1):

Solution:

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

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, de-normalized 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.

Expert Meaning (from DM Review):

First, let's describe them.

A star schema is a dimensional structure in which a single fact is surrounded by a single circle of dimensions; any dimension that is multileveled is flattened out into a single dimension. The star schema is designed for direct support of queries that have an inherent dimension-fact structure.

A snowflake is also a structure in which a single fact is surrounded by a single circle of dimensions; however, in any dimension that is multileveled, at least one dimension structure is kept separate. The snowflake schema is designed for flexible querying across more complex dimension relationships. The snowflake schema is suitable for many-to-many and one-to-many relationships among related dimension levels. However, and this is significant, the snowflake schema is *required* for many-to-many fact-dimension relationships. A good example is customer and policy in insurance. A customer can have many policies and a policy can cover many customers.

The primary justification for using the star is performance and understandability. The simplicity of the star has been one of its attractions. While the star is generally considered to be the better performing structure, that is not always the case. In general, one should select a star as first choice where feasible. However, there are some conspicuous exceptions. The remainder of this response will address these situations.

First, some technologies such a MicroStrategy require a snowflake and others like Cognos require the star. This is significant.

Second, some queries naturally lend themselves to a breakdown into fact and dimension. Not all do. Where they do, a star is generally a better choice.

Third, there are some business requirements that just cannot be represented in a star. The relationship between customer and account in banking, and customer and policy in Insurance, cannot be represented in a pure star because the relationship across these is many-to-many. You really do not have any reasonable choice but to use a snowflake solution. There are many other examples of this. The world is not a star and cannot be force fit into it.

Fourth, a snowflake should be used wherever you need greater flexibility in the interrelationship across dimension levels and components. The main advantage of a snowflake is greater flexibility in the data.

Fifth, let us take the typical example of Order data in the DW. Dimensional designer would not bat an eyelash in collapsing the Order Header into the Order Item. However, consider this. Say there are 25 attributes common to the Order and that belong to the Order Header. You sell consumer products. A typical delivery can average 50 products. So you have 25 attributes with a ratio of 1:50. In this case, it would be grossly cumbersome to collapse the header data into the Line Item data as in a star. In a huge fact table you would be introducing a lot of redundancy more than say 2 billion rows in a fact table. By the way, the Walmart model, which is one of the most famous of all time, does not collapse Order Header into Order Item. However, if you are a video store, with few attribute describing the transaction, and an average ratio of 1:2, it would be best to collapse the two.

Sixth, take the example of changing dimensions. Say your dimension, Employee, consists of some data that does not change (or if it does you do not care, i.e., Type 1) and some data that does change (Type 2). Say also that there are some important relationships to the employee data that does not change (always getting its current value only), and not to the changeable data. The dimensional modeler would always collapse the two creating a Slowly Changing Dimension, Type 2. This means that the Type 1 is absorbed into the Type 2. In some cases I have worked on, it has caused more trouble than it was worth to collapse in this way. It was far better to split the dimension into Employee (type 1) and Employee History (type 2). Thereby, in such more complex history situations, a snowflake can be better.

Seventh, whether the star schema is more understandable than the snowflake is entire subjective. I have personally worked on several data warehouse where the user community complained that in the star, because everything was flattened out, they could not understand the hierarchy of the dimensions. This was particularly the case when the dimension had many columns.

Finally, it would be nice to quit the theorizing and run some tests. So I did. I took a data model with a wide customer dimension and ran it as a star and as a snowflake. The customer dimension had many attributes. We used about 150MM rows. I split the customer dimension into three tables, related 1:1:1. The result was that the snowflake performed faster. Why? Because with the wide dimension, the DBMS could fit fewer rows into a page. DBMSs read by pre-fetching data and with the wide rows it could pre-fetch less each time than with the skinnier rows. If you do this make sure you split the table based on data usage. Put data into each piece of the 1:1:1 that is used together.

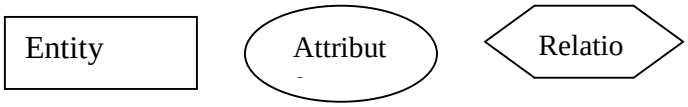
What is the point of all this? I think it is unwise to pre-determine what is the best solution. A number of important factors come into play and these need to be considered. I have worked to provide some of that thought-process in this response.

Exercise5.3:

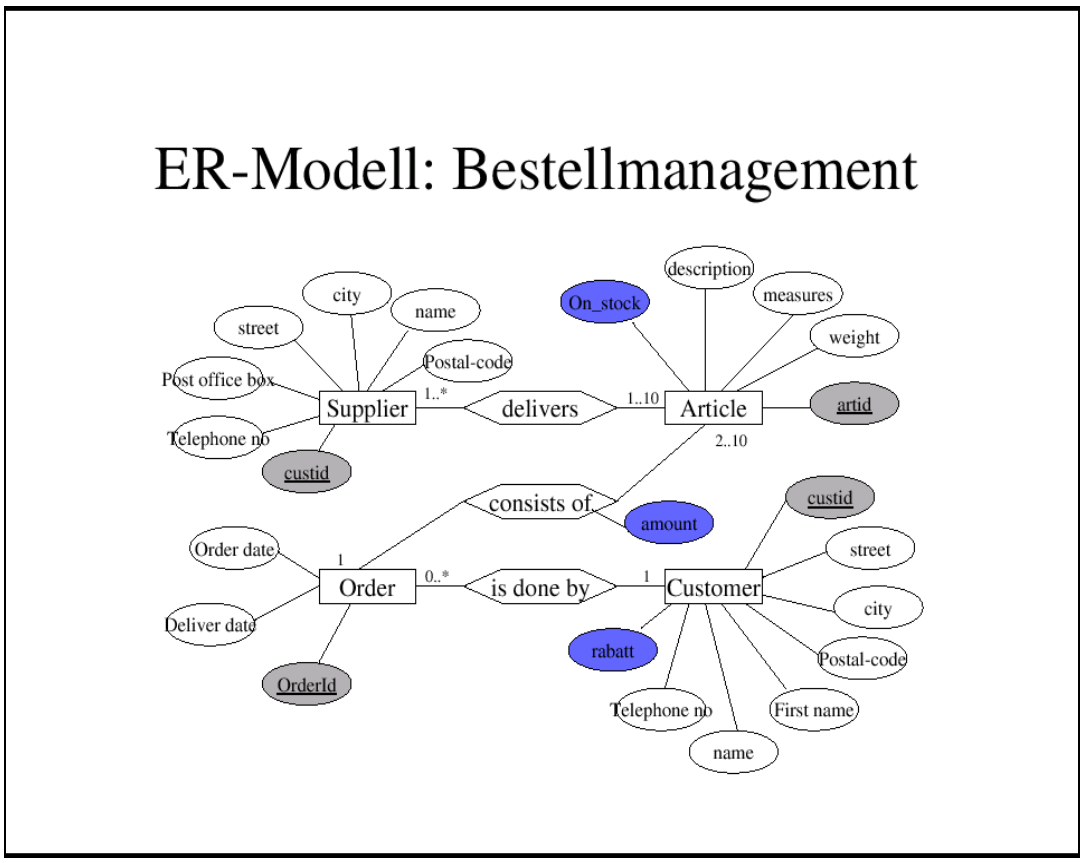
An enterprise wants to build up an ordering system.
The following objects should be administered by the new ordering system.

- **Supplier** with attributes: name, postal-code, city, street, post office box, telephone-no.
- **Article** with: description, measures, weight
- **Order** with: order date, delivery date
- **Customer** with**:** name**,** first name, postal-code, city, street, telephone-no

Conditions: Each article can be delivered by one or more suppliers. Each supplier delivers 1 to 10 articles. An order consists of 2 to 10 articles. Each article can only be one time on an order form. But you can order more than on piece of an article. Each order is done by a customer. Customer can have more than one order (no limit).
Good customers will get a 'rabatt'. The number of articles in the store should also be saved. It not important who is the supplier of the article. For each object we need a technical key for identification .
Task: Create an ER model. Model the necessary objects and the relations between them. Define the attributes and the keys. Use the following notation:



Solution:



Exercises to BA Lesson DWH&DM – Chapter 7

Exercise 7.1:

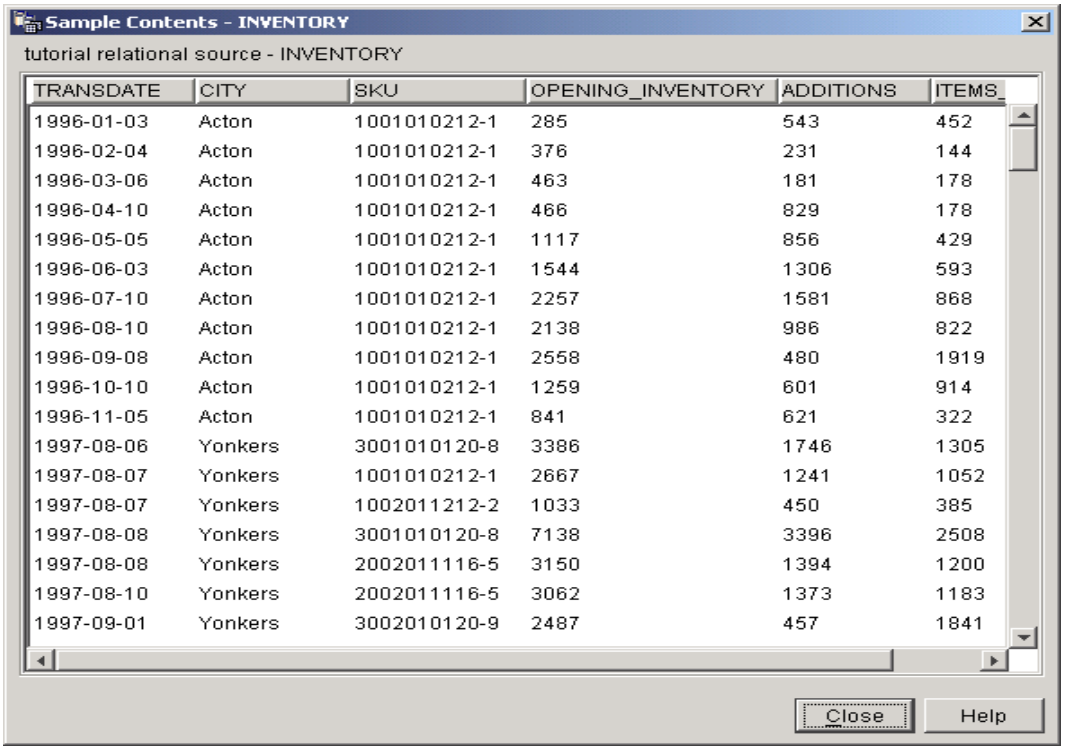
Compare the following three tables:

Sample Contents - PRODUCTION_COSTS

tutorial relational source - PRODUCTION_COSTS

TRANSDATE	CITY	SKU	COGS	MARKETING	MISC	PAYROLL
1996-01-03	Acton	1001010212-1	135	115	8	80
1996-02-04	Acton	1001010212-1	42	36	1	24
1996-03-06	Acton	1001010212-1	52	46	2	31
1996-04-10	Acton	1001010212-1	53	46	2	32
1996-05-05	Acton	1001010212-1	127	110	8	75
1996-06-03	Acton	1001010212-1	177	153	11	106
1996-07-10	Acton	1001010212-1	259	223	16	155
1996-08-10	Acton	1001010212-1	245	212	15	146
1996-09-08	Acton	1001010212-1	118	102	6	70
1996-10-10	Acton	1001010212-1	64	55	2	37
1996-11-05	Acton	1001010212-1	95	82	4	57
1997-08-06	Yonkers	3001010120-8	459	166	25	164
1997-08-07	Yonkers	1001010212-1	313	270	19	182
1997-08-07	Yonkers	1002011212-2	113	100	6	69
1997-08-08	Yonkers	3001010120-8	1000	343	56	352
1997-08-08	Yonkers	2002011116-5	454	221	21	163
1997-08-10	Yonkers	2002011116-5	437	216	21	162
1997-09-01	Yonkers	3002010120-9	130	44	6	43
1997-09-02	Yonkers	1002011212-2	56	48	1	32

Close Help

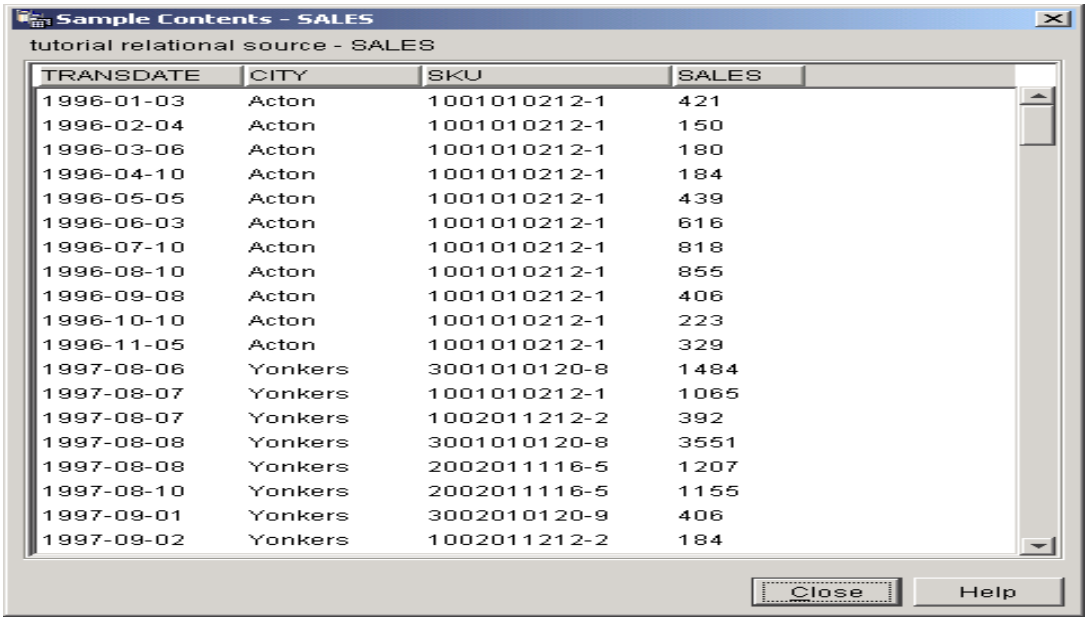


Sample Contents - INVENTORY

tutorial relational source - INVENTORY

TRANSDATE	CITY	SKU	OPENING_INVENTORY	ADDITIONS	ITEMS_
1996-01-03	Acton	1001010212-1	285	543	452
1996-02-04	Acton	1001010212-1	376	231	144
1996-03-06	Acton	1001010212-1	463	181	178
1996-04-10	Acton	1001010212-1	466	829	178
1996-05-05	Acton	1001010212-1	1117	856	429
1996-06-03	Acton	1001010212-1	1544	1306	593
1996-07-10	Acton	1001010212-1	2257	1581	868
1996-08-10	Acton	1001010212-1	2138	986	822
1996-09-08	Acton	1001010212-1	2558	480	1919
1996-10-10	Acton	1001010212-1	1259	601	914
1996-11-05	Acton	1001010212-1	841	621	322
1997-08-06	Yonkers	3001010120-8	3386	1746	1305
1997-08-07	Yonkers	1001010212-1	2667	1241	1052
1997-08-07	Yonkers	1002011212-2	1033	450	385
1997-08-08	Yonkers	3001010120-8	7138	3396	2508
1997-08-08	Yonkers	2002011116-5	3150	1394	1200
1997-08-10	Yonkers	2002011116-5	3062	1373	1183
1997-09-01	Yonkers	3002010120-9	2487	457	1841

Close Help



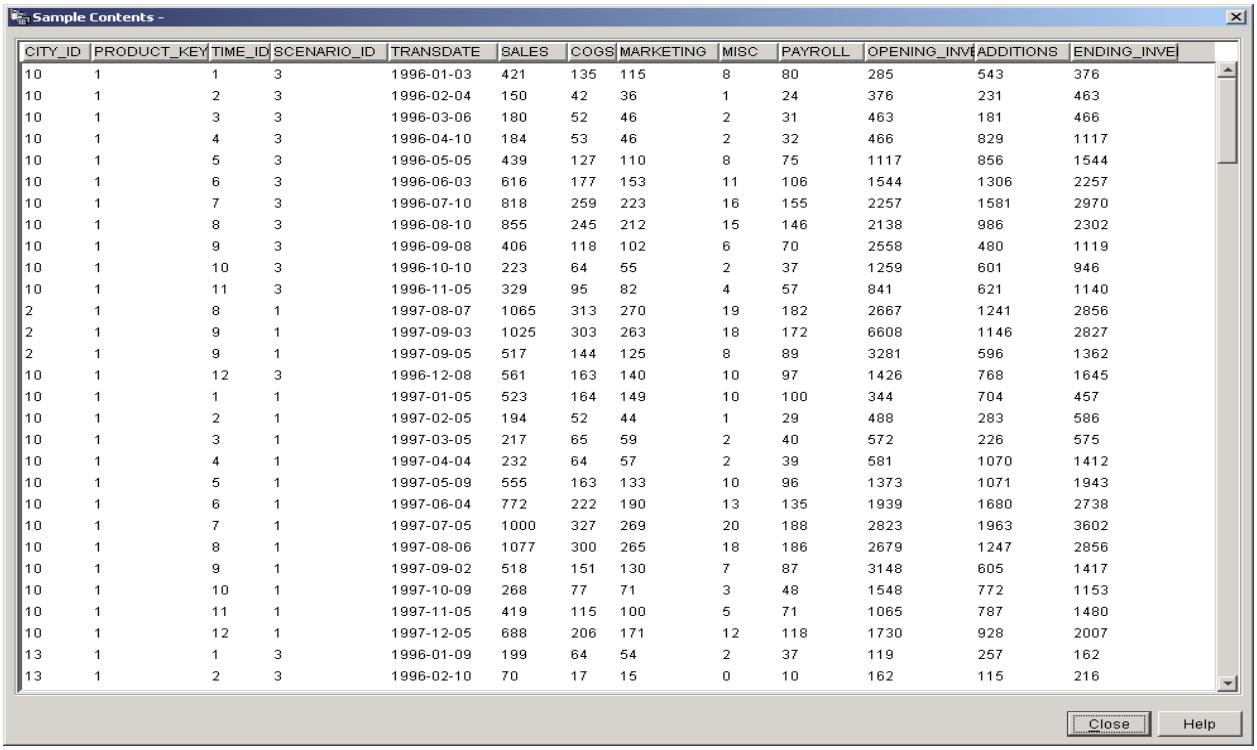
Sample Contents - SALES

tutorial relational source - SALES

TRANSDATE	CITY	SKU	SALES
1996-01-03	Acton	1001010212-1	421
1996-02-04	Acton	1001010212-1	150
1996-03-06	Acton	1001010212-1	180
1996-04-10	Acton	1001010212-1	184
1996-05-05	Acton	1001010212-1	439
1996-06-03	Acton	1001010212-1	616
1996-07-10	Acton	1001010212-1	818
1996-08-10	Acton	1001010212-1	855
1996-09-08	Acton	1001010212-1	406
1996-10-10	Acton	1001010212-1	223
1996-11-05	Acton	1001010212-1	329
1997-08-06	Yonkers	3001010120-8	1484
1997-08-07	Yonkers	1001010212-1	1065
1997-08-07	Yonkers	1002011212-2	392
1997-08-08	Yonkers	3001010120-8	3551
1997-08-08	Yonkers	2002011116-5	1207
1997-08-10	Yonkers	2002011116-5	1155
1997-09-01	Yonkers	3002010120-9	406
1997-09-02	Yonkers	1002011212-2	184

Close Help

write a SQL script, s.t. you get the following target table:



Sample Contents -

CITY_ID	PRODUCT_KEY	TIME_ID	SCENARIO_ID	TRANSDATE	SALES	COGS	MARKETING	MISC	PAYROLL	OPENING_INVE	ADDITIONS	ENDING_INVE
10	1	1	3	1996-01-03	421	135	115	8	80	285	543	376
10	1	2	3	1996-02-04	150	42	36	1	24	376	231	463
10	1	3	3	1996-03-06	180	52	46	2	31	463	181	466
10	1	4	3	1996-04-10	184	53	46	2	32	466	829	1117
10	1	5	3	1996-05-05	439	127	110	8	75	1117	856	1544
10	1	6	3	1996-06-03	616	177	153	11	106	1544	1306	2257
10	1	7	3	1996-07-10	818	259	223	16	155	2257	1581	2970
10	1	8	3	1996-08-10	855	245	212	15	146	2138	986	2302
10	1	9	3	1996-09-08	406	118	102	6	70	2558	480	1119
10	1	10	3	1996-10-10	223	64	55	2	37	1259	601	946
10	1	11	3	1996-11-05	329	95	82	4	57	841	621	1140
2	1	8	1	1997-08-07	1065	313	270	19	182	2667	1241	2856
2	1	9	1	1997-09-03	1025	303	263	18	172	6608	1146	2827
2	1	9	1	1997-09-05	517	144	125	8	89	3281	596	1362
10	1	12	3	1996-12-08	561	163	140	10	97	1426	768	1645
10	1	1	1	1997-01-05	523	164	149	10	100	344	704	457
10	1	2	1	1997-02-05	194	52	44	1	29	488	283	586
10	1	3	1	1997-03-05	217	65	59	2	40	572	226	575
10	1	4	1	1997-04-04	232	64	57	2	39	581	1070	1412
10	1	5	1	1997-05-09	555	163	133	10	96	1373	1071	1943
10	1	6	1	1997-06-04	772	222	190	13	135	1939	1680	2738
10	1	7	1	1997-07-05	1000	327	269	20	188	2823	1963	3602
10	1	8	1	1997-08-06	1077	300	265	18	186	2679	1247	2856
10	1	9	1	1997-09-02	518	151	130	7	87	3148	605	1417
10	1	10	1	1997-10-09	268	77	71	3	48	1548	772	1153
10	1	11	1	1997-11-05	419	115	100	5	71	1065	787	1480
10	1	12	1	1997-12-05	688	206	171	12	118	1730	928	2007
13	1	1	3	1996-01-09	199	64	54	2	37	119	257	162
13	1	2	3	1996-02-10	70	17	15	0	10	162	115	216

Close Help

We see the following conditions:

1. Map the name of the cities in the sources to a number 1 – 100, define this as **City_Id**
2. Define last digit of SKU in SALES as **Product_Key**
3. Define them Month of Transdate as **Time _Id (**range:01 –12)
4. Def. **Scenario_Id** with cases (Year of Transdate)=1997 as 1,(…Transdate)=1996 as 3 , else 2
5. Fill all columns of target table with the same columns of sources
6. Define new column: **Ending_Inventory** = (Opening_Inv. + Additions) -Items_Sold

Solution:

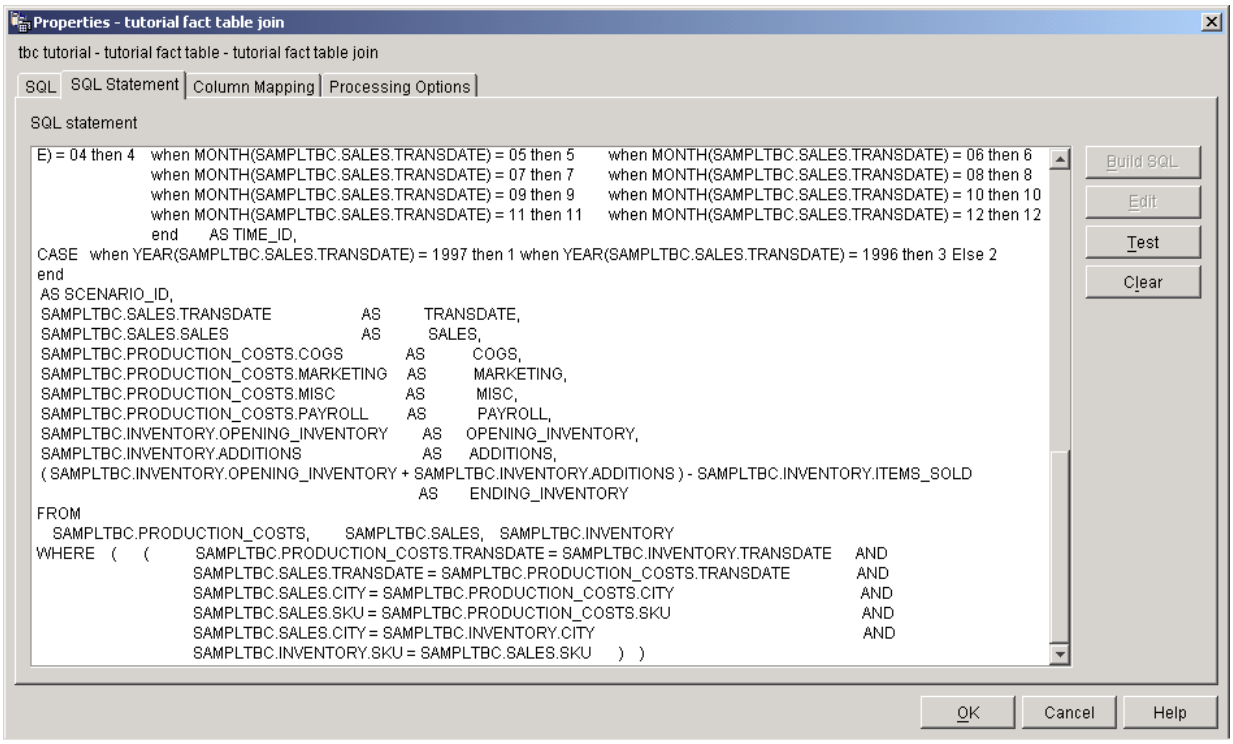
Select
Case SAMPLTBC.sales.city
When ‘Manhattan’ **then** 1
…..
When ‘Maui’ **then** 100
End
As City_Id

Substr (SAMPLTBC.sales.SKU,12,1) **as** Product_Key

Case
When Month(SAMPLTBC.sales.transdate) = 01 **then** 1
….
When Month(SAMPLTBC.sales.transdate) = 12 t**hen** 12
End
As Time_Id

Case
When Year(SAMPLTBC.sales.transdate) = 1997 **then** 1

When Year(SAMPLTBC.sales.transdate) = 1996 **then** 3
Else 2
End
As Scenario_Id
….
See screenshot:



Exercise 7.2:

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

Solution - Informatica:

Powercenter

INFORMATICA
The Data Integration Company™

- ETL-Tool mit verschiedenen Lösungen
  - Standard
  - Advanced
  - Virtualization
  - Real Time (z.B. mit Echtzeitkonnektivität)
  - Big Data (z.B. mit natürlicher Sprachverarbeitung)

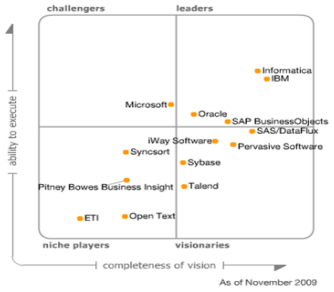
Funktionen

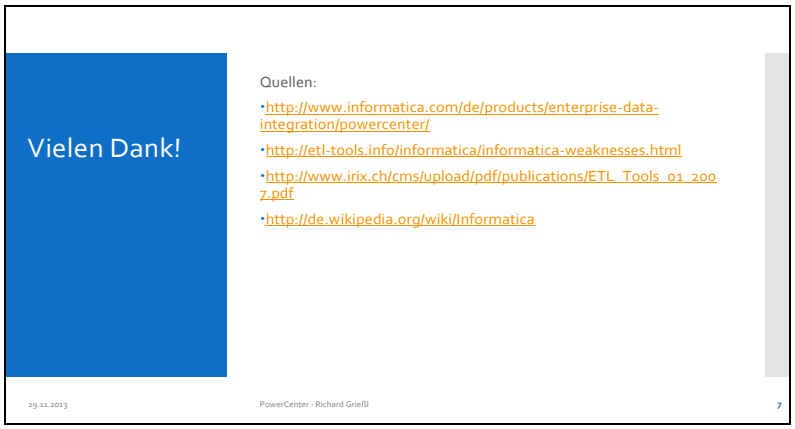
- Datenextraktion, Datenintegration, Datenmigration, Datenkonsolidierung, Datentransformation, Datenverdichtung und Datenbereinigung
- Flexibel Einsetzbar
- gemeinsame Entwicklungsumgebungen
- Automatisierte Test Tools
- Überwachung der Integrationsprozesse
- Schulungen

Stärken und Schwächen

- Stärken:
  - Einheitliche Benutzeroberfläche
  - Sehr umfassend
  - Optimierte Lösungen
- Schwächen:
  - Fehlerhaftes sortieren
  - Filter werden ignoriert (beim Kopieren)
  - Probleme beim Importieren von XML

Vergleich





Exercises (+solutions) to BA Lesson DWH&DM – Chapter 8

Exercise 8.1:

Find and define the Benefits & Drawbacks of

- MOLAP
- ROLAP

Systems
Use the information of the lesson or use your own experience

Solution:

Criteria	**ROLAP**	**MOLAP**
Data volume	+ > 50 GB possible: low expansion factor (low aggregation rate)	- Not > 50 GB: expansion factor too big (high aggregation rate)
Dimensions	+ > 10 possible (depends only on DBMS)	- Bad performance for > 10 (due to high aggregation rate)
Query Performance	(+ When querying single tables) - When joining many tables	+ When using high aggregated data (- when using low aggregated data)
Update flexibility	+ Update during operation possible + Fast and flexible	- Cube has to be rebuilt completely each time (partly correct, in depends on calculation rules) - Operation has to be stopped
Query complexity	+ Complex, dynamic queries possible (impact on query performance)	- Only standard queries, that the cube is built for, possible (but combinations are possible)
Usability	- Not intuitive: SQL knowledge necessary	+ Intuitive, easy to handle, no special knowledge required
Price	+ Cheaper; simpler SQL-based front-ends sufficient (but more performance needed)	- Expensive; costly front-end tool necessary

➔ ROLAP is for many criteria superior to MOLAP. As most data marts today are bigger than 50 GB, ROLAP is many cases better choice due to performance and storage reasons.

Second Solution:

MOLAP		ROLAP	
-	Erstellen der Cubes aufwendig	+	Komplexere Anfragen möglich (Verwendung von SQL)
+	Schnelle Queries	-	Anwender muss SQL-Kenntnisse haben
-	Zugriff nur auf Daten des Cubes	-	Queries dauern länger, da komplexer
+	Auf Problemstellung angepasste Anfragen möglich	+	Zugriff auf alle Daten in DB
-	Bei Update müssen Cubes neu erstellt werden	+	Update auch während Operationen möglich
-	Nur bei Cube-Erstellung definierte Anfragen möglich		

Exercise 8.2:

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:
Vorstellung von Cognos:

Überblick

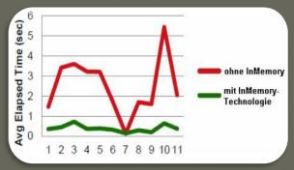
- Übernahme von IBM 2008
- 3500 Mitarbeiter (Sitz in Frankfurt am Main)
- Softwarelösungen
  - Business Intelligence
  - Geschäftsanalyse
  - finanzielles Performance Management

Funktionen

- Abfragen & Berichte
  - Jahreseinkommen, Quartalszahlenbericht
- Dashboards
  - Interaktiver Zugriff auf Inhalt, mit personalisiertem Erscheinungsbild und Kriterien für Daten
- Analyse
  - Informationszugriff aus verschiedenen Blickwinkeln und Perspektiven
- Zusammenarbeit
  - Kommunikationstools und Social Networking
- Echtzeitüberwachung

Besonderheiten

- InMemory-Technologie (Nutzung des Arbeitsspeicher)



- Mobile Client

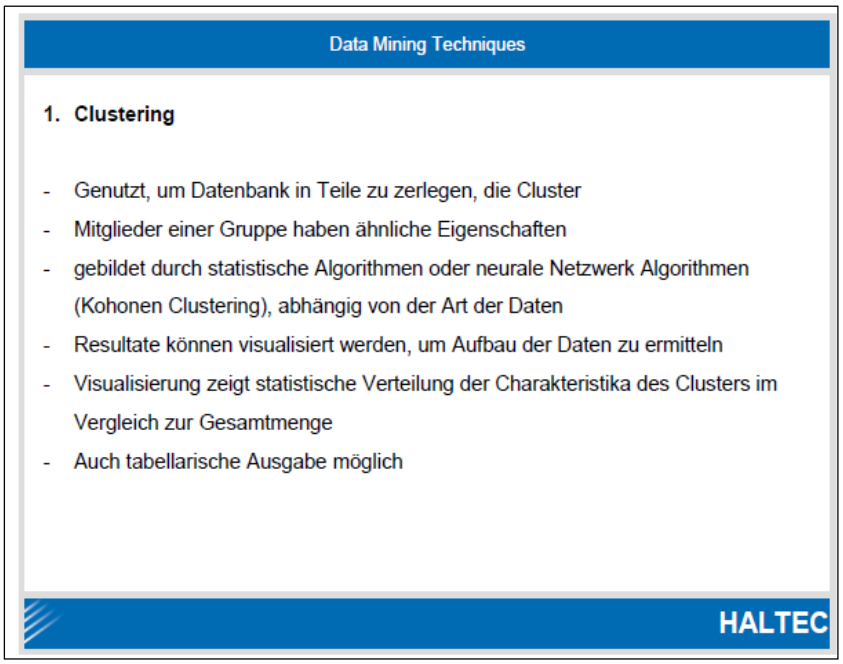
See the Internet

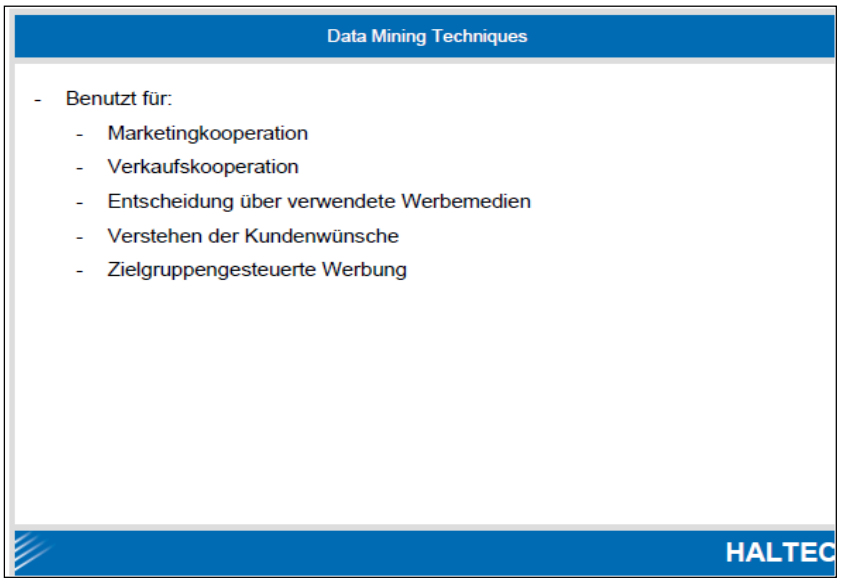
Exercises (+Solutions) to BA Lesson DWH&DM – Chapter 9

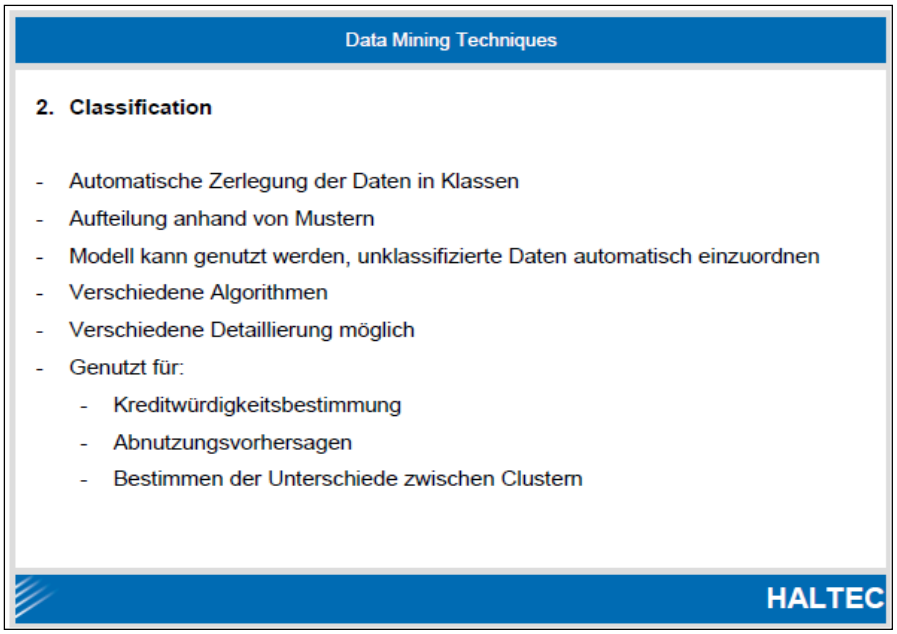
Exercise 9.1:
Describe the following Data Mining techniques. Search this information in the internet, i.e. Wikipedia or other knowledge portals:

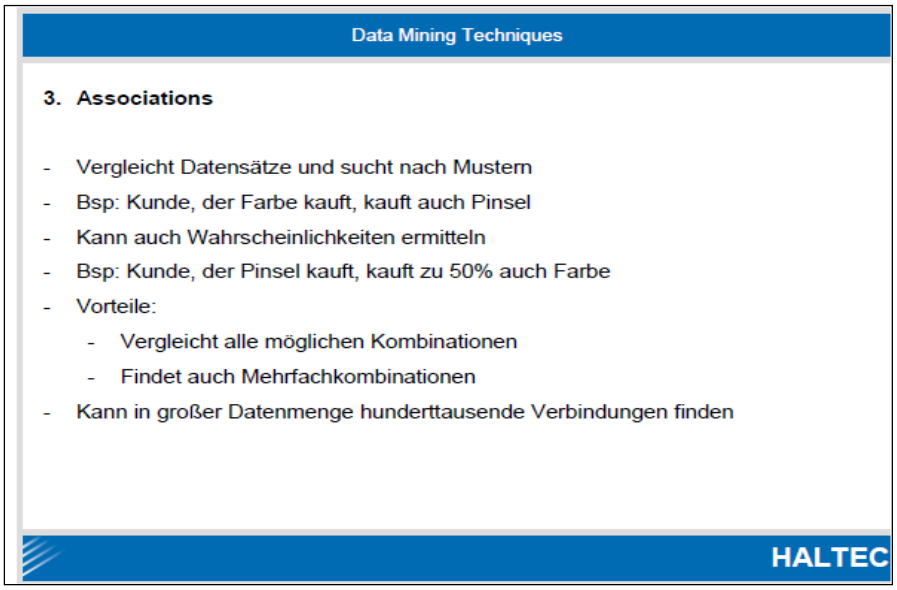
- **Clustering**
- **Classification**
- **Associations**

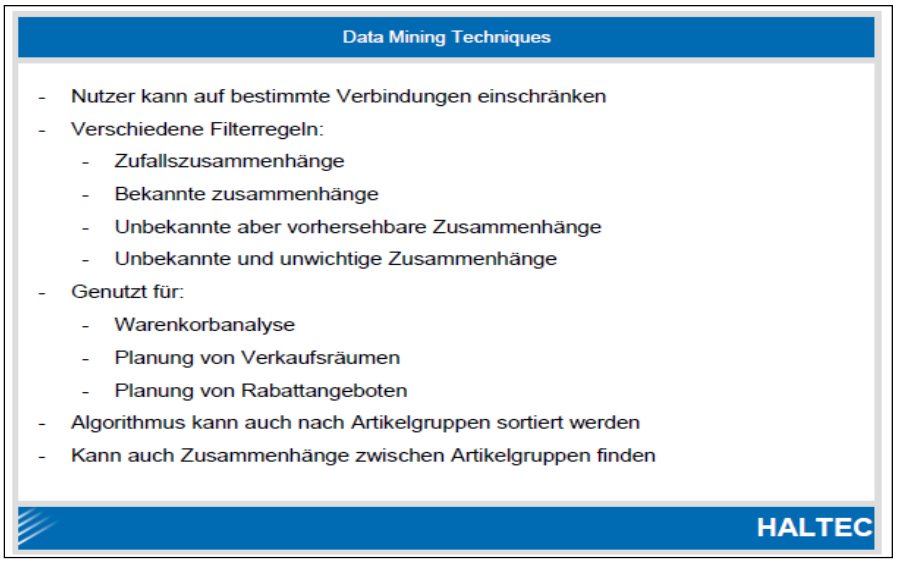
Solution:






Data Mining Techniques
2. Classification
- Automatische Zerlegung der Daten in Klassen
- Aufteilung anhand von Mustern
- Modell kann genutzt werden, unklassifizierte Daten automatisch einzuordnen
- Verschiedene Algorithmen
- Verschiedene Detaillierung möglich
- Genutzt für:
  - Kreditwürdigkeitsbestimmung
  - Abnutzungsvorhersagen
  - Bestimmen der Unterschiede zwischen Clustern
HALTEC

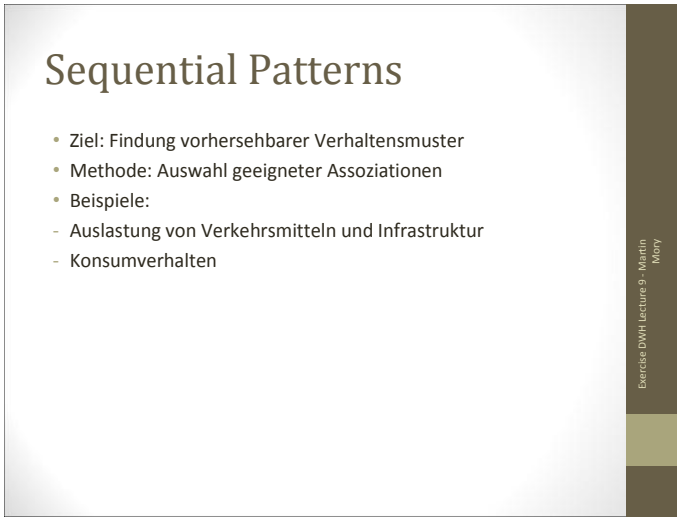

Data Mining Techniques
3. Associations
- Vergleicht Datensätze und sucht nach Mustern
- Bsp: Kunde, der Farbe kauft, kauft auch Pinsel
- Kann auch Wahrscheinlichkeiten ermitteln
- Bsp: Kunde, der Pinsel kauft, kauft zu 50% auch Farbe
- Vorteile:
  - Vergleicht alle möglichen Kombinationen
  - Findet auch Mehrfachkombinationen
- Kann in großer Datenmenge hunderttausende Verbindungen finden
HALTEC

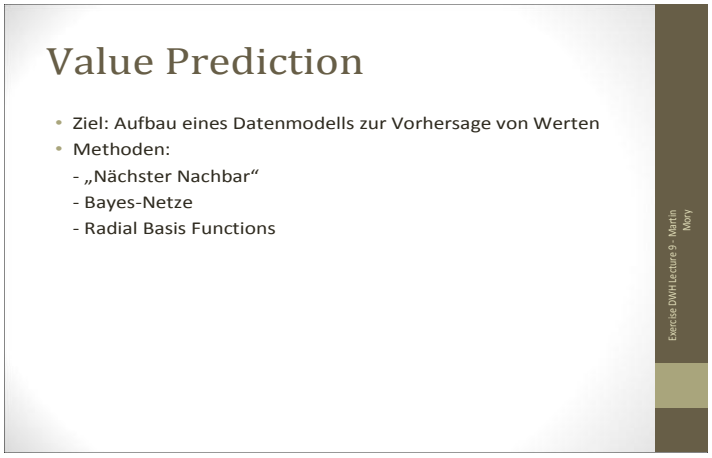

Data Mining Techniques
- Nutzer kann auf bestimmte Verbindungen einschränken
- Verschiedene Filterregeln:
  - Zufallszusammenhänge
  - Bekannte zusammenhänge
  - Unbekannte aber vorhersehbare Zusammenhänge
  - Unbekannte und unwichtige Zusammenhänge
- Genutzt für:
  - Warenkorbanalyse
  - Planung von Verkaufsräumen
  - Planung von Rabattangeboten
- Algorithmus kann auch nach Artikelgruppen sortiert werden
- Kann auch Zusammenhänge zwischen Artikelgruppen finden
HALTEC

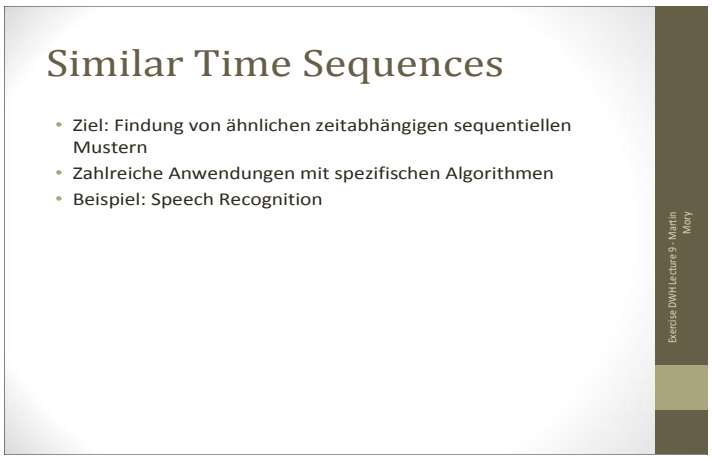
Exercise 9.2:
Describe the following Data Mining techniques. Search this information in the internet, i.e. Wikipedia or other knowledge portals:

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

Solution:







Exercises (+Solutions) to BA Lesson DWH&DM – Chapter 10

Exercise 10.1:
Search for the actual "Gartner Quadrant" of DM tools. Give detail description of two of the leading DM tools in the quadrant: