AIFB

Kapitel 6

Data Warehouse

6.1 Grundlagen

6.2 Was ist ein Data Warehouse?

6.3 Architektur

6.4 Entwicklungszyklus

aus dem Heise-News-Ticker vom 16.12.1999

IBM baut weltgrößtes Data Warehouse für Telekom

Die Deutsche Telekom hat bei IBM das mit **100 Terabyte** bislang größte Data Warehouse geordert. Dort will die Telekom künftig **alle Kundendaten zentral** erfassen und Applikationen zur Kundenbetreuung installieren. Laut IBM soll das Data Warehouse im dritten Quartal 2000 mit zunächst 25 Terabyte in Betrieb gehen und danach schrittweise innerhalb von 18 Monaten bis zur endgültigen Größe ausgebaut werden. Über die Kosten des Projekts ist nichts bekannt.

Den Aufbau des Data Warehouse übernimmt die Deutsche Telekom Innovationsgesellschaft (T-Nova) auf Basis von IBMs PowerPC-Multiprozessorservern RS/6000 SP und dem IBM-Datenbanksystem DB2. IBM ist nach eigenen Angaben mit 173 installierten Systemen, die mehr als ein Terabyte Daten verwalten, Marktführer in diesem Bereich.

AIFB

6.1 Grundlagen

- Anspruch an Datenbanken in Unternehmen ist vielschichtig.
- Unterteilung nach Einsatzzweck
  - **Operative Systeme**
    - eingesetzt von Sachbearbeitern, am Bankschalter, etc.
    - dienen der täglichen Arbeit
  - **Informationelle Systeme**
    - helfen dem Management, (strategische) Entscheidungen zu finden.
      - DSS = decision support systems
      - Systeme für Business Intelligence Anwendungen
    - bieten Grundlage für weitere Analysen mit OLAP / Data Mining

AIFB

Informationelle Systeme

- zugeschnitten auf **Gegenstandsbereiche** (sog. ***Subjects***), z.B.
  - Kunde,
  - Produkt,
  - Vertriebsregion
- unterstützen **Informations- und Analyseaufgaben**, d.h. das Management in der Entscheidungsfindung (vgl. nächste Folien)
- wenige Zugriffe aber mit relativ **hohem Datenvolumen**
- Datenbankeinträge werden nicht geändert (**keine Updates**)
- Antwortzeitverhalten spielt untergeordnete Rolle

Unterstützte Informations- und Analyseaufgaben

vgl. [Dueck 99] und Vorlesung "Knowledge Discovery"

- Typische Branchen für sog. **Business Intelligence** Anwendungen
  - Banken und Versicherungen
  - (Einzel-)Handel
  - Telekommunikation
  - Transportwesen
- Typische **BI Anwendungen**
  - Risikoabschätzung, Cross-Selling, Portfolio-Analyse
  - Warenkorbanalyse, Kundenverhalten
  - Customer Relationship Management
  - Erkennen von Missbrauch (*fraud detection*)
  - Kampagnen-Management

AIFB

Analyse-Methoden und -Werkzeuge

- Interaktive Techniken / OLAP
  - Erzeugen von **Zusammenfassungen**, z.B.
    - *Wie sieht die Summe S der Umsätze aller Filialen aus?*
  - **Drill-Down-Analysis**: zusammengefasste Daten werden analysiert, z.B.
    - *Welche Filiale macht bei dieser Summe den größten Umsatz?*
    - *Welche Produkte tragen zum Umsatz der Filiale bei?*
  - **Ranking**, z.B.
    - *Mit welchen Produkten haben wir den höchsten/ niedrigsten Umsatz gemacht?*

AIFB

Analyse-Methoden und -Werkzeuge (2)

- Data Mining / Statistik
  - **Segmentierung:**
    - *Gibt es Gruppen von Kunden mit ähnlichem Verhalten?*
  - **Klassifikation**:
    - *In welche dieser Gruppe(n) gehört ein neuer Kunden?*
  - **Vorhersage**:
    - *Welche Kunden könnte ich demnächst (warum) verlieren?*
  - **Abhängigkeitsanalyse**:
    - *Wie reagierten meine Kunden auf die letzte Marketingaktion? (Kampagnenmanagement)*
  - **Warenkorbanalyse**:
    - *Welche Waren(-gruppen) werden häufig gemeinsam gekauft?*
  - **Fraud Detection**:
    - *Wie erkennt man Benutzung gestohlener Kreditkarten?*

AIFB

Informationelle Systeme

- enthalten sehr **große Datenmengen**
- enthalten zum großen Teil **historische, zusammengefasste Daten**
  - Historie aus Daten der operativen Systeme ist nachvollziehbar
- relativ hohe **Redundanz**
- Überblick über alle relevanten Unternehmensdaten
- **komplexe**, oft heuristische **Ad-hoc-Anfragen**
  - z.B. auf der Basis von OLAP-Funktionalitäten
- Daten sind **wohl strukturiert, integriert** und **konsolidiert**
- Anzahl Benutzer ist eher klein („Power-User“)

AIFB

Operativ vs. Informationell

Aus diesen Charakteristiken ergeben sich

fundamentale Wiedersprüche:

Operative Systeme	Informationelle Systeme
• Schnelle Antwortzeit • Anwendungsorientiert • Aktuelle Daten • Detaillierte, primäre Daten • Häufige Änderungen • Dient täglicher Arbeit	• Hohe Speicherkapazität • Gegenstandsorientiert • Historische Daten • Auch zusammengefasste, abgeleitete Daten • Keine Updates • Dient als Datenspeicher für Analyse und Entscheidungsfindung

⇒ Man muss beide Systeme trennen.

Data Warehouse für den informationellen Systemteil

6.2 Was ist ein Data Warehouse?

- „Mit dem Begriff Data Warehouse wird eine von den operationalen DV-Systemen isolierte Datenbank umschrieben, die als unternehmensweite Datenbasis für Management-Unterstützungssysteme dient.“

  [Muksch et al. 1996]

- „A Data Warehouse is a
  - subject-oriented,
  - integrated,
  - time-variant,
  - nonvolatile

  collection of data in support of management‘s decision-making process.“

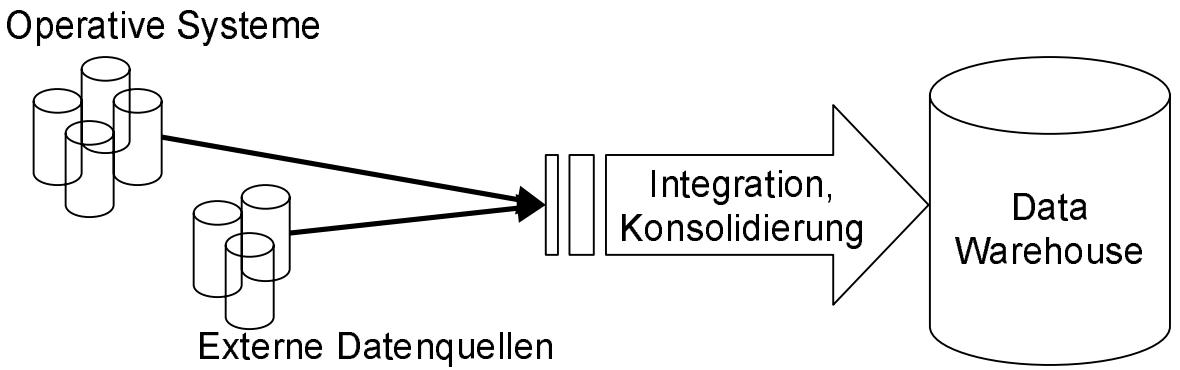
  [Inmon, Hackathron 1994]

AIFB

6.2.1 Gegenstandsorientierung (subject-oriented)

- DWh ist an **Gegenstandsbereichen** des Unternehmens orientiert,
  - z.B. Produkten, Kunden, Lieferanten
- Gegensatz zu **Funktions**- oder **Anwendungsorientierung** bei operativen (**legacy**) Systemen:
  - Funktionen sind z.B. Einkauf, Lagerhaltung, Verkauf
- Bei der Entwicklung eines DWh stehen die **Daten im Mittelpunkt**.
  - Bei operationalen Systemen muss auch der Prozess berücksichtigt werden.
- DWh enthält nur solche Daten, die für DSS-Analysten/ Manager relevant und interessant sind, bzw. werden könnten.

6.2.2 Integration



- In zwei verschiedenen operativen Systemen können
  - die **gleichen Daten** unter **verschiedenen Bezeichnern** abgelegt sein
  - die **gleichen Bezeichner** für **verschiedene Zwecke** benutzt werden
  - der **gleiche Sachverhalt** auf **verschiedene Weise** kodiert sein

AIFB

6.2.2 Integration (Forts.)

- Daten aus **verschiedenen Quellen** werden im DWh vereinheitlicht, u.a. durch
  - **konsistente** Vergabe und Definition von Bezeichnern
  - **einheitliche Kodierung**
    - z.B. wird jedes Datum in der Form <YYY-MM-DD> gespeichert
  - einheitliches Festlegen der Maßeinheiten von Attributen
    - z.B. werden Preise in Dollar angegeben
  - **Auflösung von strukturellen Konflikten**
    - z.B. Schema-Wert-Konflikt
- Integration führt dazu, dass alle Daten im DWh in einer einzigen, **allgemein akzeptierten Art und Weise** gespeichert sind.
- Erst die Integration erlaubt die einfache und effektive Nutzung der DWh-Daten für Anwendungen z.B. im Management
- Integration ist ein **schwieriger und zeitaufwendiger** Prozess

Strukturkonflikten in relationalen Schemata

Beispiel: Datenbank für Aktienkurse [Saltor et. al. 1993]

- Datenbank New York (ein Tupel pro Tag pro Aktie)

date	stock	clsprice
991008	IBM	347
991008	HP	418
991008	GM	250
991009	IBM	350
991009	HP	420
991009	GM	215

- Datenbank Barcelona (eine Relation pro Tag, ein Attribut pro Aktie)

date	HP	IBM	GM
991008	418	365	250
991009	420	350	200

- Datenbank Melbourne (eine Relation pro Aktie, ein Tupel pro Tag)

HP	date	clsprice
	991008	425
	991009	420

IBM	date	clsprice
	991008	347
	991009	350

GM	date	clsprice
	991008	385
	991009	320

AIFB

Beobachtungen bei Strukturkonflikten

- **Werte** des *stock*-Attributs in DB New York entsprechen
  - Namen der **Attribute** in DB Barcelona
  - Namen von **Relationen** in DB Melbourne
- **Daten** in DB New York entsprechen
  - **Schemadaten** in DB Barcelona bzw. DB Melbourne
- Beseitigung der Strukturkonflikte erfordert Transformation
  - von Daten in Schemadaten und
  - von Schemadaten in Daten

AIFB

Behandlung der Strukturkonflikten

- **Beobachtungen**:
  - Schema der DB Melbourne ist **„*Spezialisierung*“** des Schemas der DB New York,
    d.h. jede Relation in DB Melbourne ist „*Subklasse*“ (Partition) der Relation *S* in DB New York
  - Schema der DB New York ist **„*Generalisierung*“** des Schemas von DB Melbourne,
    d.h. Relation *S* in DB New York ist „*Superklasse*“ der Relationen in DB Melbourne

⇒ verwende die Operationen

- ***partition by attribute***
- ***discriminated union***

für Schematransformationen

AIFB

Behandlung der Strukturkonflikten: *partition by attribute*

- Transformation von Daten in Schemadaten
- **Idee**:
  - Partition einer Relation über die Werte eines ausgewählten, partitionierenden Attributes
  - resultierende Relationen haben gleiches Schema wie Ausgangsrelation, jedoch ohne partitionierendes Attribut
  - **Werte** des partitionierenden Attributes **werden zu Namen der resultierenden Relationen**
- **Beispiel**:
  - Relationen der DB Melbourne entstanden durch Partitionierung der Relation S aus DB New York, mit dem stock-Attribut als partitionierendem Attribut
- invers zu *discriminated union*

AIFB

Behandlung der Strukturkonflikten: *discriminated union*

- Transformation von Schemadaten in Daten
- **Idee**:
  - Vereinigung von n kompatiblen Relationen in eine neue Relation mit einem zusätzlichen Attribut (Diskriminante)
  - **Diskriminante hat** Namen der Operanden**relationen als Wert**
  - Extension der Resultat-Relation entsteht durch:
    - Produktbildung der Extension jeder Operanden-Relation mit dem Namen der Operanden-Relation
    - Vereinigung dieser Resultate
- **Beispiel**:
  - Relation S in DB New York ist discriminated union der 3 Relationen in DB Melbourne; stock-Attribut ist die Diskriminante
- invers zu *partition by attribute*

AIFB

Weitere Strukturkonflikte

- **Beobachtung**:
  - Schema der DB Barcelona ist **Aggregation** des Schemas der DB New York,
    d.h. jedes Tupel der Relation S in DB Barcelona ist die kartesische Aggregation der Werte mit gleichem Datum der Relation S in DB New York
  - Schema der DB New York ist **Dekomposition** des Schemas der DB Barcelona,
    d.h. mehrere Tupel der Relation S in DB New York sind Teile desselben Tupels der Relation S in DB Barcelona
- Dieser Strukturkonflikt kann durch Transformation von Aggregierungsbeziehungen beseitigt werden:
  - Die Operationen
    - ***decomposition***
    - ***composition***

    werden aber hier nicht vorgestelt.

AIFB

6.2.3 Zeitraumbezug (time variancy)

- In **operativen Systemen** ist der **aktuelle Datenbestand** gespeichert. Er kann jederzeit geändert werden (**Update**).
- DWh enthält eine ganze **Historie von Daten**
- DWh besteht aus **Snapshots** der operativen Systeme
- DWh-Daten sind zu einem bestimmten Zeitpunkt gültig (gewesen). Der **Gültigkeitszeitraum** ist an allen Daten im DWh vermerkt (als Teil des Schlüssels).
- **Zeithorizont** des DWh: ca. 5-10 Jahre
  - operative Systeme: max. 60-90 Tage

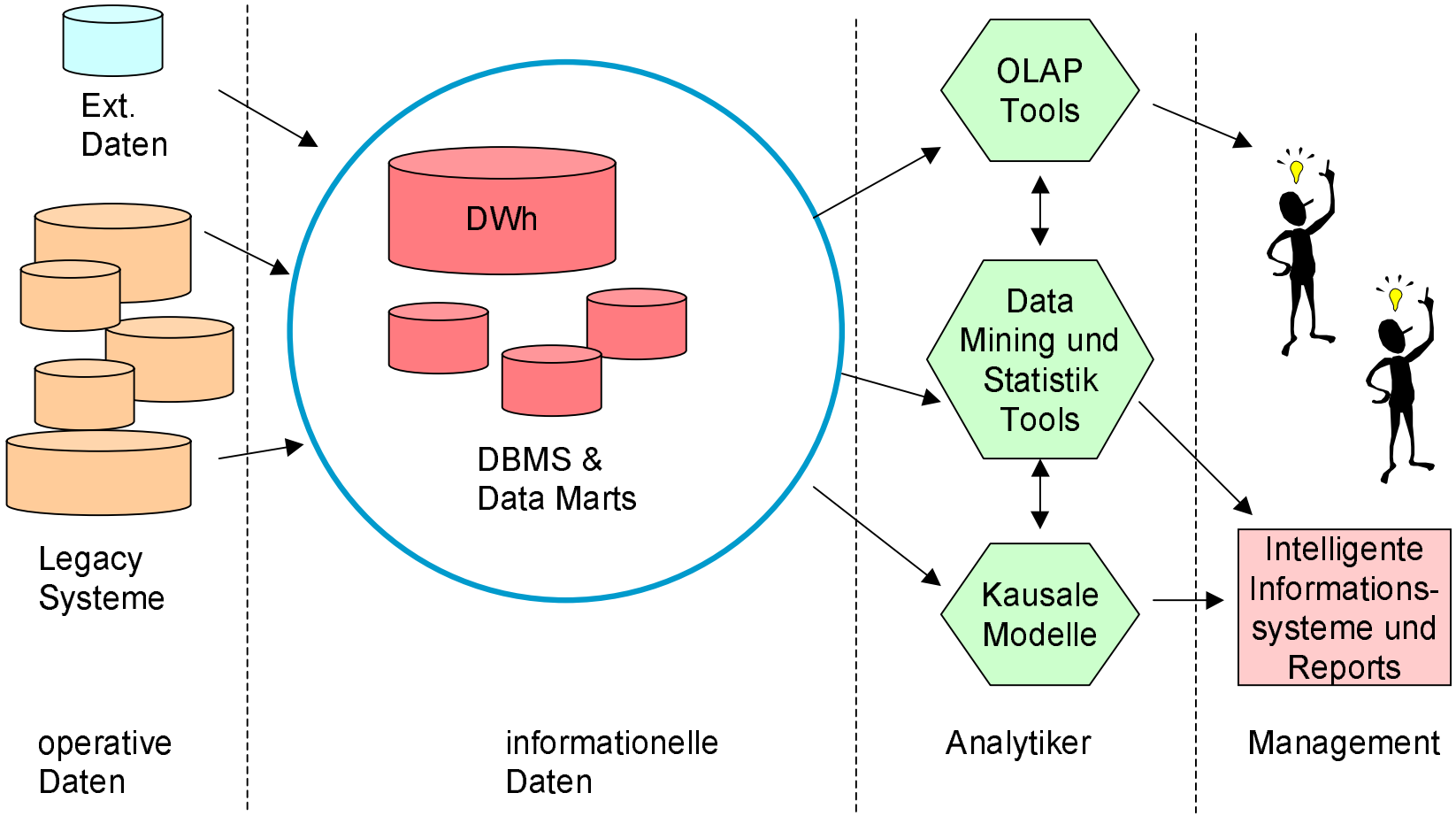
AIFB

6.2.4 Beständigkeit (nonvolatility)

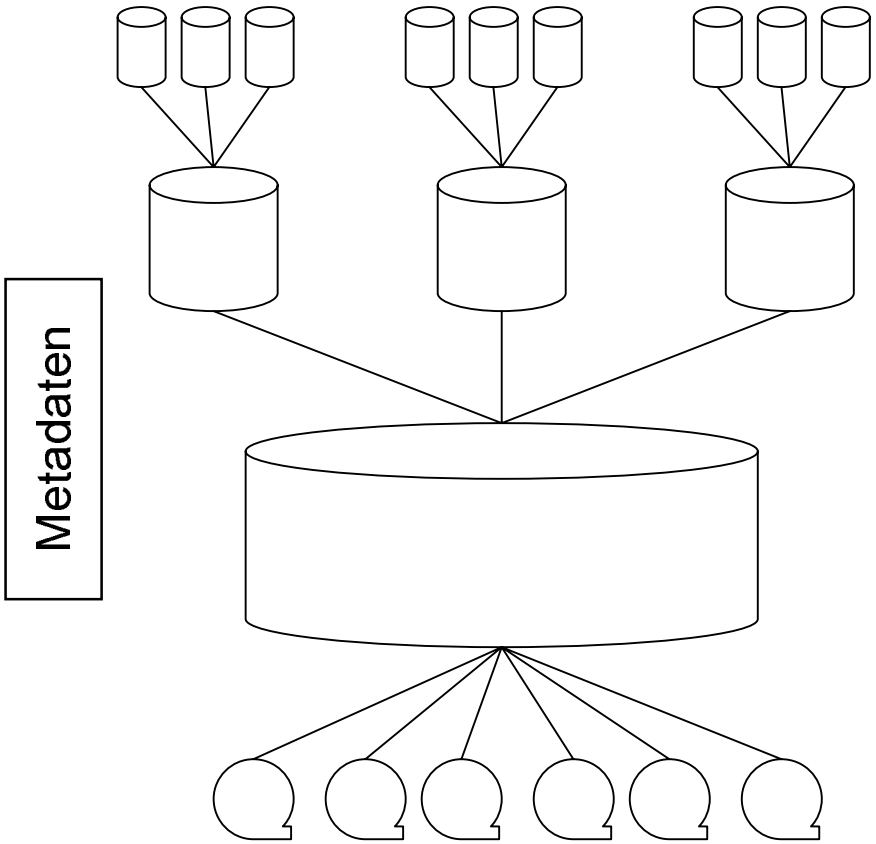
- **Operative Systeme**:
  - **Daten werden oft geändert, gelöscht, eingefügt.**
  - Aufwendige Mechanismen, um Deadlocks zu vermeiden
  - Locking-Mechanismen etc.

- DWh:
  - **primär nur Leseoperationen**
  - Daten werden aus den operativen Systemen (initial) **geladen.**
  - Analysesysteme greifen **lesend** auf DWh-Daten zu.
  - Es gibt **keine Updates**.

6.3 Architektur einer Data Warehouse Umgebung



Die innere Struktur eines Data Warehouse



Stark zusammengefasste Daten

Leicht zusammengefasste Daten

Aktuelle detaillierte Daten

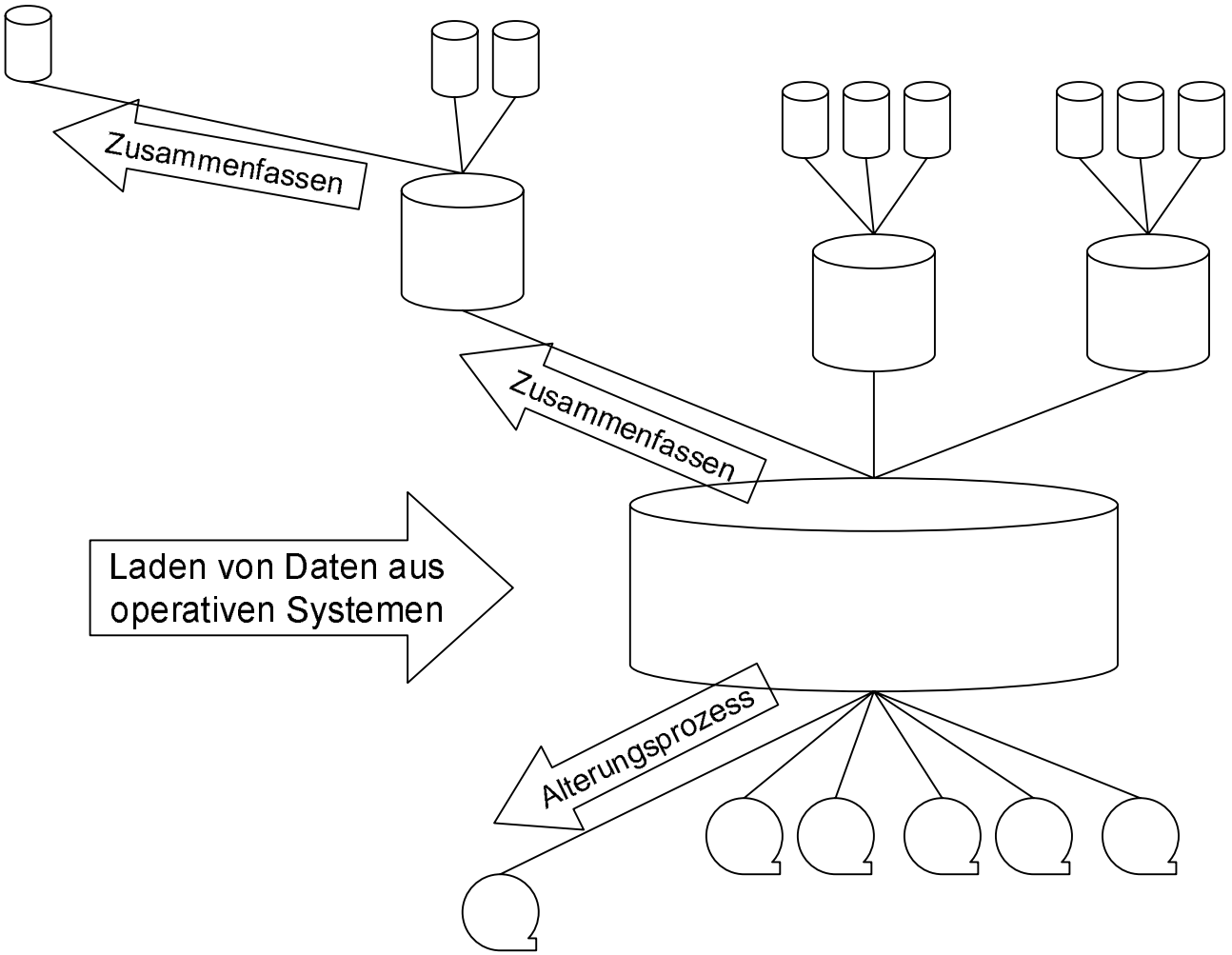
Alte (detaillierte) Daten

Beispiel: Telekommunikationsunternehmen

detailliert	leicht zusammengefasst

• Für jeden Kunden, jedes Gespräch inkl.
  • Zone
  • Teilnehmer
  • Zeitpunkt
  • Dauer
  • Gebühren
  • Art des Dienstes
∅ 45.000 Byte pro Kunde
	• Für jeden Kunden
	  • Anzahl der Gespräche insgesamt
	  • Anzahl der Ferngespräche
	  • ∅ Gesprächsdauer
	  • Umsatz je Zone
	  • Umsatz insgesamt
	
	Zusammenfassung monatlich
	ca. 200 Byte pro Kunde

Datenflüsse im Data Warehouse



AIFB

Metadaten

- Metadaten sind Daten über Daten
- Metadaten lassen sich in drei Kategorien einteilen:
  - semantische
  - verwaltungstechnische und
  - schematische

- **semantische Metadaten**
  - Festlegung der DWh-**Terminologie**
  - **Transformations**- und **Integrationsregeln** für die Abbildung der operativen Daten in die DWh-Daten
  - **Aggregationsregeln** für das Zusammenfassen der Daten auf verschiedenen Aggregationsstufen

AIFB

Metadaten

- **verwaltungstechnische Metadaten**
  - Festlegung von **Benutzer (-gruppen)** und zugehörige **Zugriffsrechte**
  - **statistische** Daten über das DWh
    - **Größe** von Tabellen
    - **Zugriffshäufigkeiten** auf Tabellen, Aggregationsstufen

- **schematische Metadaten**
  - **logisches Schema** des DWh
  - Abbildung zwischen logischem und physischem Schema
  - Quellen der DWh-Daten

AIFB

6.4 DWh-Entwicklungszyklus

- DWh-Entwicklungszyklus unterscheidet sich von klassischer System-Entwicklung:
  - Am Anfang des Data-Warehouse-Entwicklungszyklus stehen die Daten (der Prozess ist ***data-driven***)
  - Das Data Warehouse wird **schrittweise** entwickelt.
  - **Gründe:**
    - genaue Ziele/ Anforderungen an das DWh sind meistens noch nicht bekannt, Größe auch schlecht abschätzbar
    - Kosten und Entwicklungszeit schlecht abschätzbar
    - benötigte Ressourcen (Mitarbeiter, Rechner, …) sind hoch

AIFB

Iterative Vorgehensweise

- **iteratives Vorgehen** und kurze ***Feedback Loops*** haben viele Vorteile:
  - Anwender können ihre Anforderungen erst dann detailliert artikulieren, wenn der erste DWh-Prototyp vorliegt (1. Stufe)
  - Management wird erst dann größeres Projektbudget genehmigen, wenn positive Resultate sicher greifbar sind.
  - Qualität des DWh wird durch Feedback Loops mit Anwendern deutlich verbessert.

  ⇒ Leitmotiv:

  Think big! Start small! Grow step by step!

AIFB

Monitoring der DWh-Benutzung

- Monitoring ist Voraussetzung für Anpassung des DWh an aktuelle Nutzung
  - Welche Daten des DWh werden regelmäßig genutzt?
  - In welchem Umfang wächst der Datenbestand?
  - Wer benutzt das DWh?
  - Welche Antwortzeiten treten bei welchen Anfragen auf?
  - Wie ist die Belastung des DWh?

AIFB

Datenschutz

- Sicherheit und Datenschutz im DWh:
  - Daten in einem DWh können **schutzwürdig** sein:
    - Finanzdaten
    - medizinische Daten
    - sonstige personenbezogene Daten (Einkommen, ...)
  - Beispiel: **Telekom**
    - detaillierte Daten über „Telefonierverhalten“ aller Telekomkunden