Data Warehousing

Architektur

Komponenten

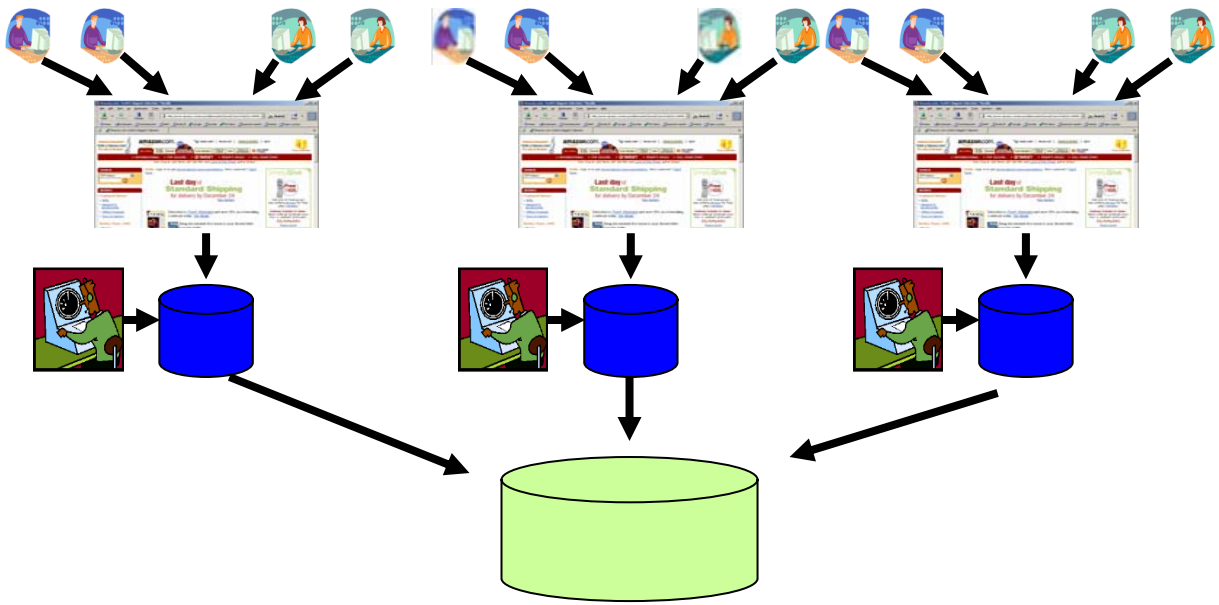
Prozesse

Ulf Leser

Wissensmanagement in der Bioinformatik

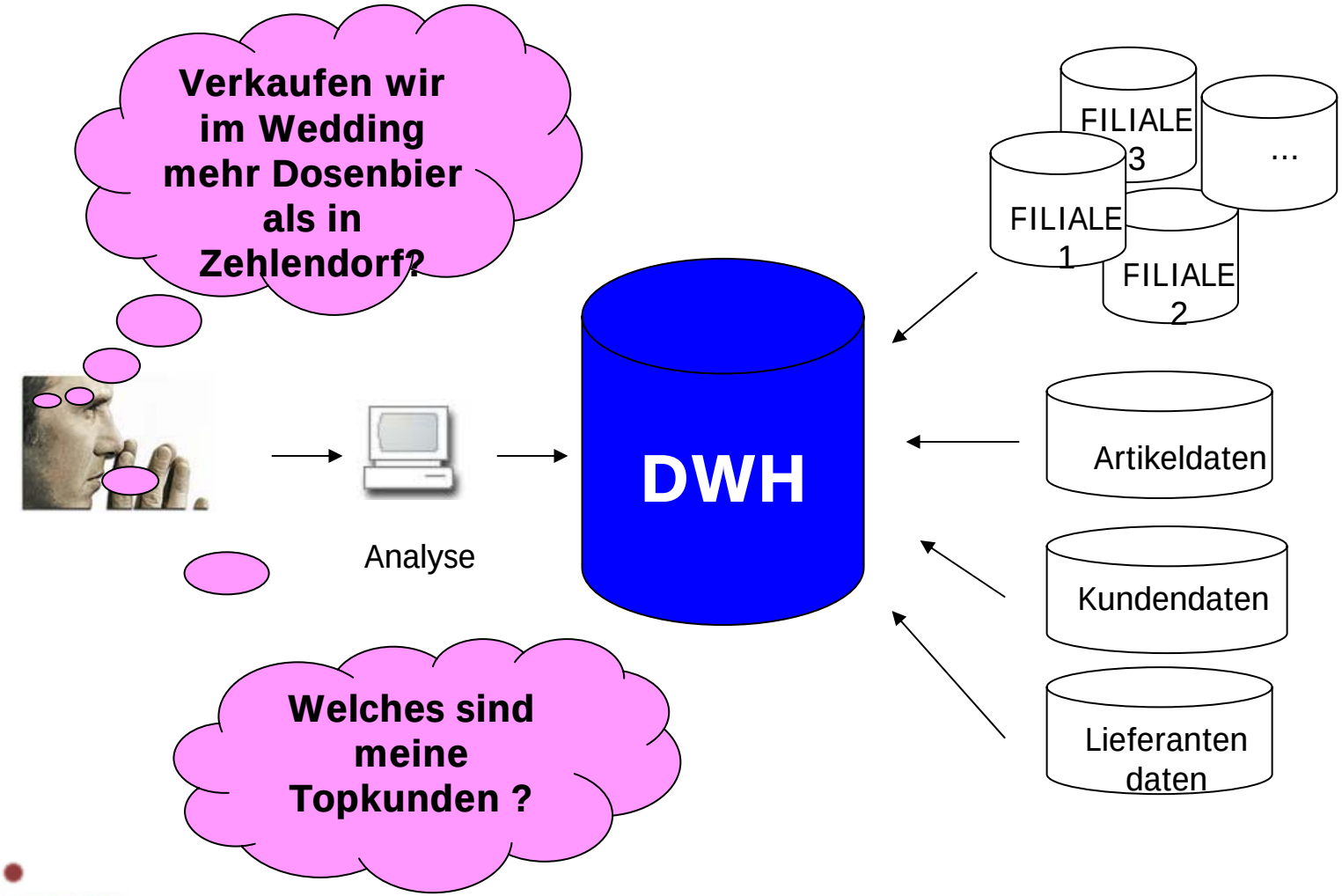
Zusammenfassung letzte Vorlesung

Aufbau eines Data Warehouse



- Redundante, transformierte Datenhaltung
- Asynchrone Aktualisierung

Zweck: Analyse und Integration



Vergleich OLTP - OLAP

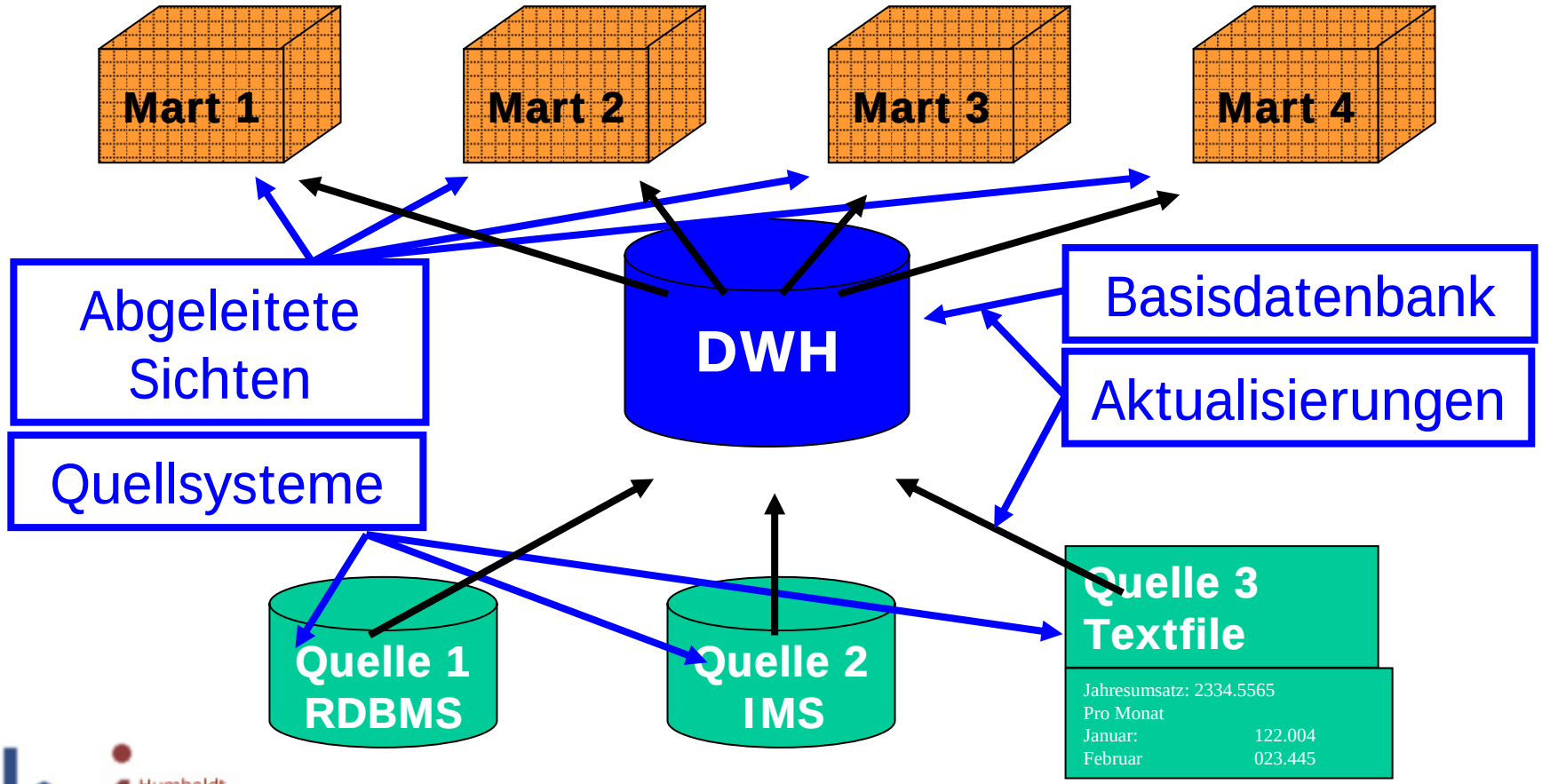
	OLTP	OLAP
Typische Operationen	Insert, Update, Delete, Select	Select Bulk-Inserts
Transaktionen	Viele und kurz	Lesetransaktionen
Typische Anfragen	Einfache Queries, Primärschlüsselzugriff, Schnelle Abfolgen von Selects/inserts/updates/deletes	Komplexe Queries: Aggregate, Groupierung, Subselects, etc. Range Queries über mehrere Attribute
Daten pro Operation	Wenige Tupel	Megabyte
Datenmenge in DB	Gigabyte	Terabyte
Eigenschaften der Daten	Rohdaten, häufige Änderungen	Abgeleitete Daten, historisch & stabil
Modellierung	Anwendungsorientiert	Themenorientiert
Typische Benutzer	Sachbearbeiter	Management

Inhalt dieser Vorlesung

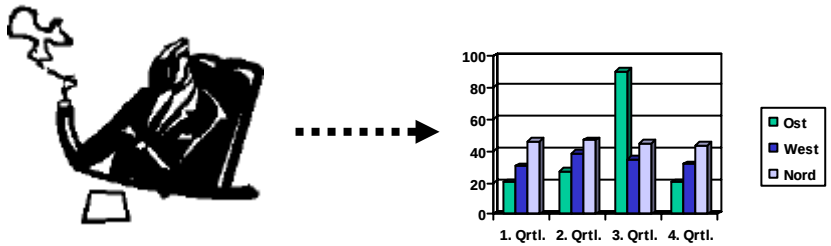
- Architektur
- Komponenten
- Prozesse

DWH Grobarchitektur

Hubs and Spokes



DWH Architektur & Komponenten



Monitoring

Analysewerkzeuge

Metadaten

Quelle 1
RDBMS

Quelle 2
IMS

Staging Area

Staging Area

Cube

Mart 2

Mart 1

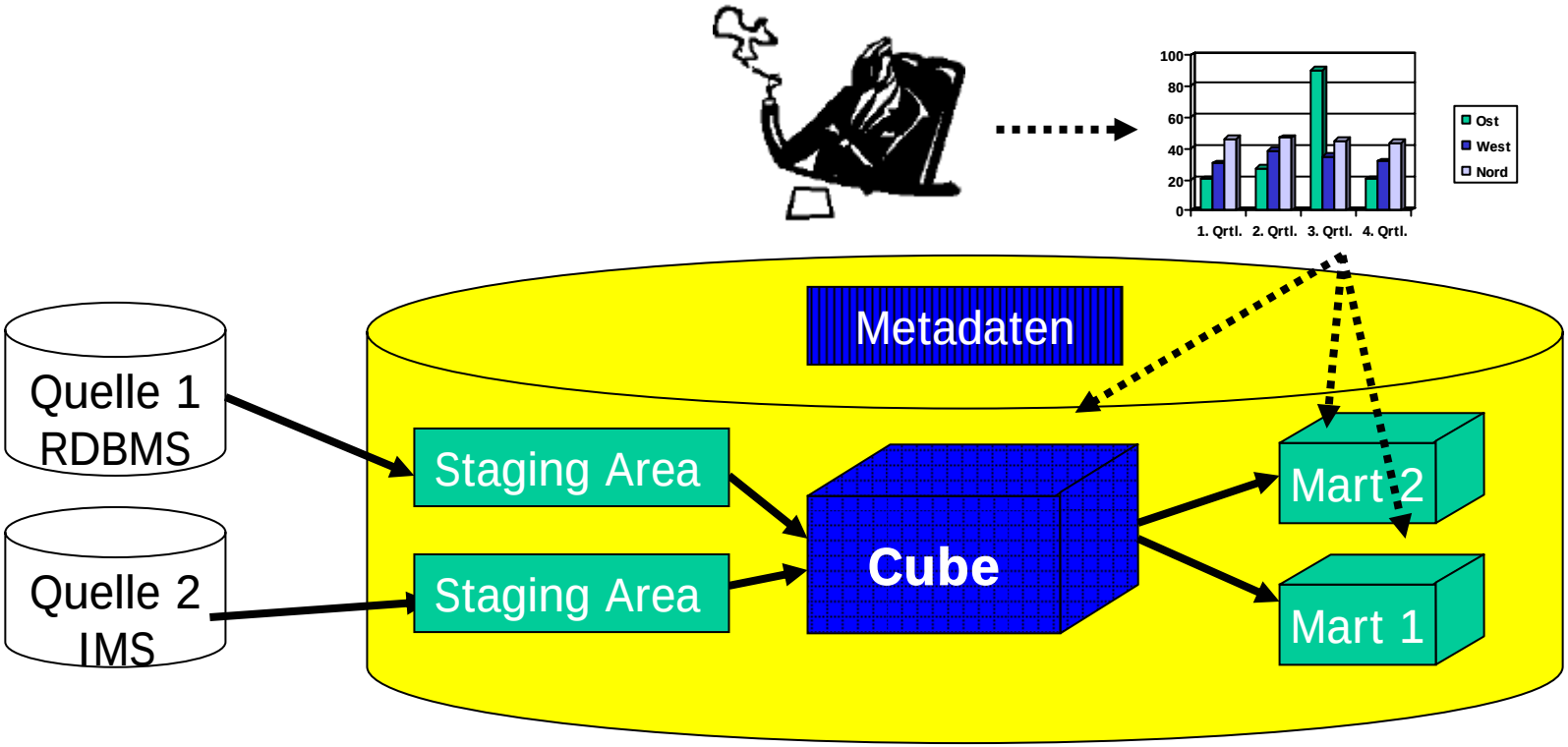
Datenquellen

Basisdaten

Abgeleitete Sichten

Arbeitsbereich

Langlebigkeit

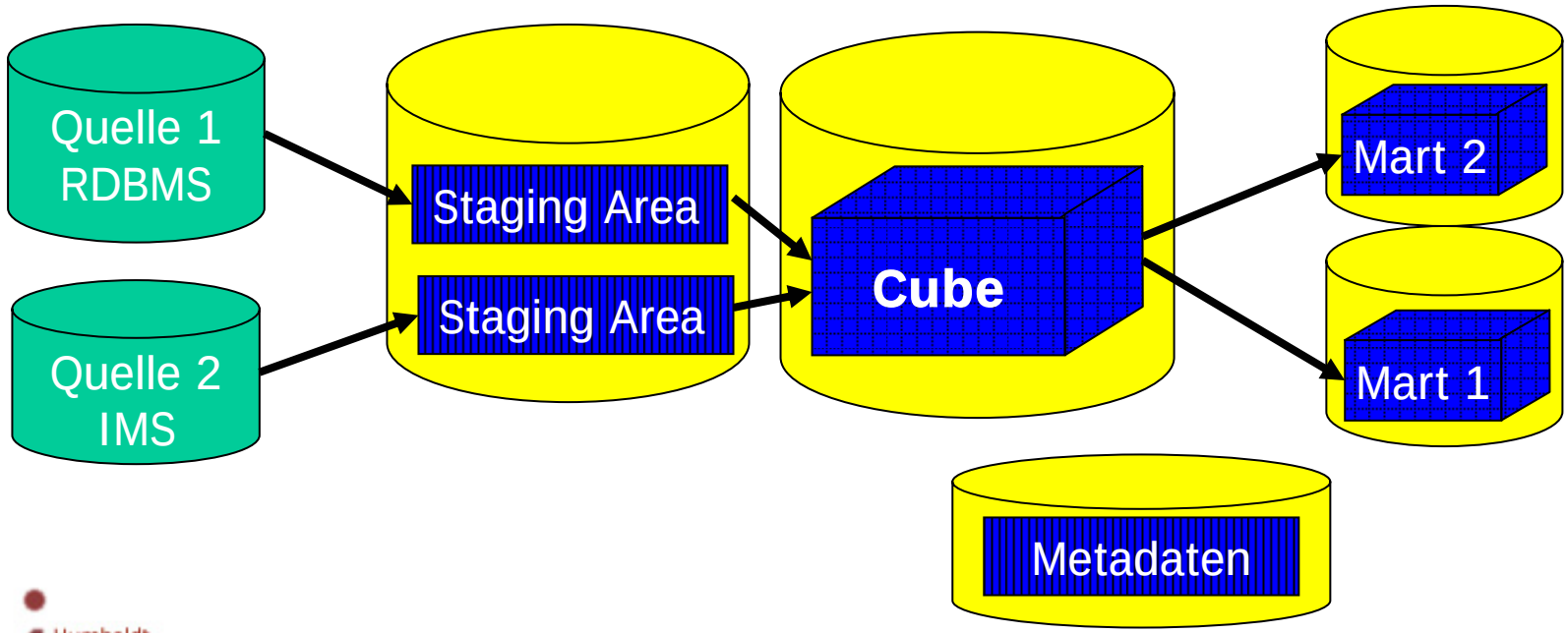


Flüchtig

Persistent

Alternativen

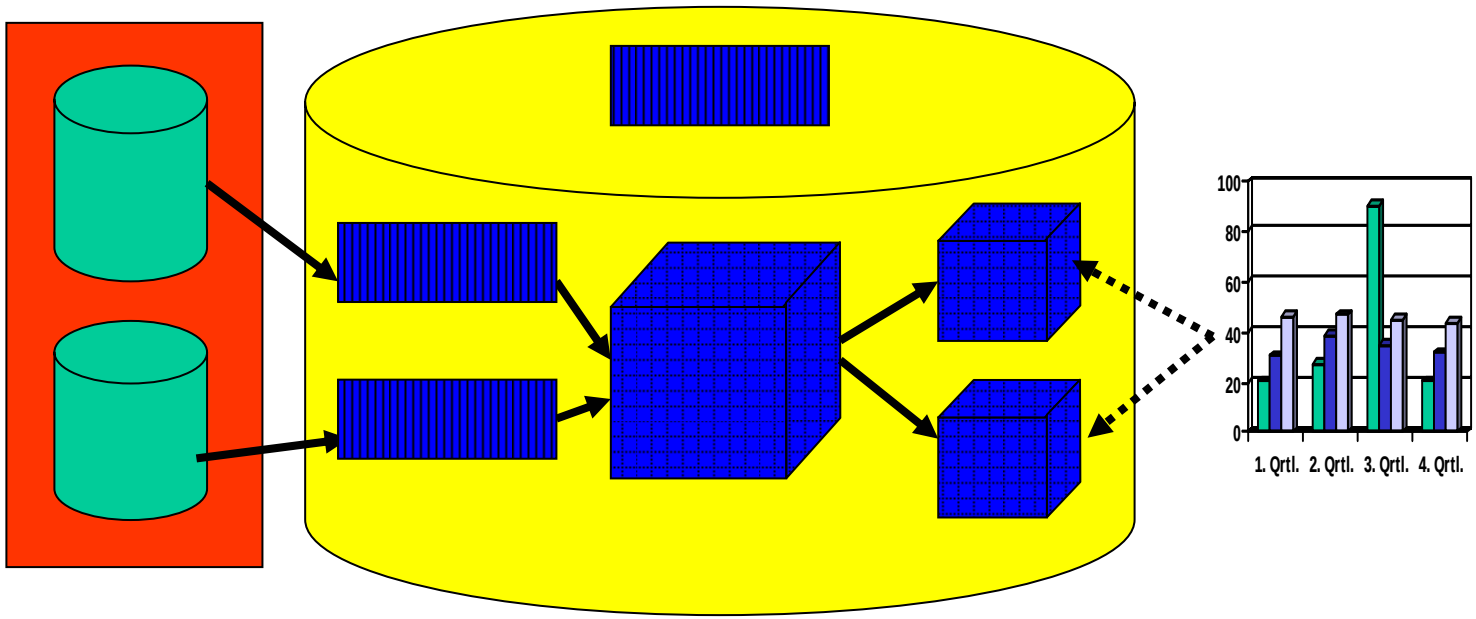
- Physikalische Aufteilung variabel
  - Data Marts auf eigenen Rechnern (Laptop)
  - Staging Area auf eigenen Servern
  - Metadaten auf eigenem Server (Repository)



Komponenten im Einzelnen

1. Datenquellen
2. Staging Area
3. Basisdatenbank
4. Abgeleitete Sichten
5. Analysewerkzeuge
6. Metadatenrepository
7. Data Warehouse Manager

1. Datenquellen

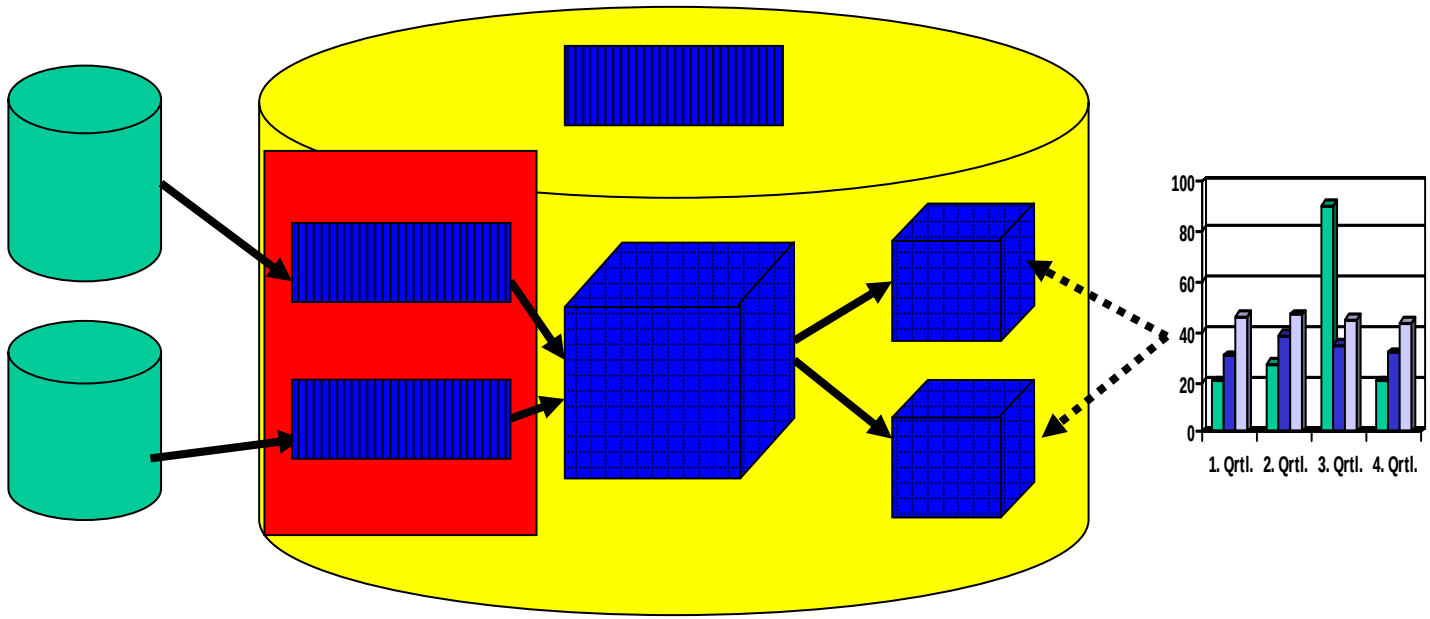


Datenquellen

- Meist sehr heterogen
  - Technisch: RDBMS, IMS, Mainframe, Textfiles, ...
  - Logisch: Schema, Format, Repräsentation,...
  - Syntaktisch: Datum, Währung, Zahlenkodierung, ...
  - Verfügbarkeit: Kontinuierlich, Periodisch, ...
  - Qualität: Fehlende / falsche Werte, Duplikate, ...
  - Rechtlich: Datenschutz (Kunden & Mitarbeiter!)
- Zugriff
  - Push: Quelle erzeugt regelmäßig Extrakte
  - Pull: DWH stößt Zugriff an / Online-Zugriff

➢ Individuelle Behandlung notwendig

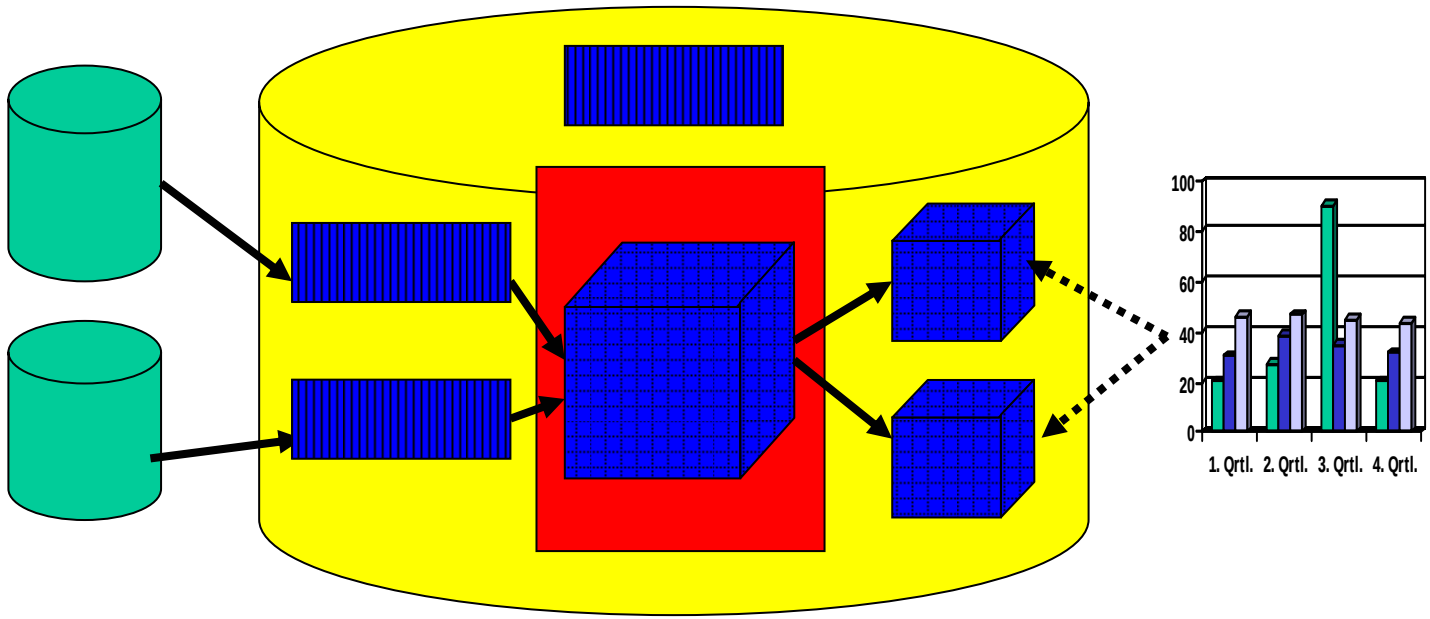
2. Arbeitsbereich (Staging Area)



Arbeitsbereich

- Temporärer Speicher
- Quellnahes Schema
- Sinn
  - ETL Arbeitsschritte effizienter implementierbar
    - Mengenoperationen, SQL
  - Zugriff auf Basisdatenbank möglich (Upsert)
  - Vergleich zwischen Datenquellen möglich
  - Filterfunktion: Nur einwandfreie Daten in Basisdatenbank übernehmen

3. Basisdatenbank



Basisdatenbank

- Zentrale Komponente des DWH
  - Begriff „DWH“ meint oft nur die Basisdatenbank
- Speichert Daten in feinster Auflösung
  - Einzelne Verkäufe
  - Einzelne Bons
- Historische Daten
- Große Datenmengen
  - Spezielle Modellierung
  - Spezielle Optimierungsstrategien

DWH als ...

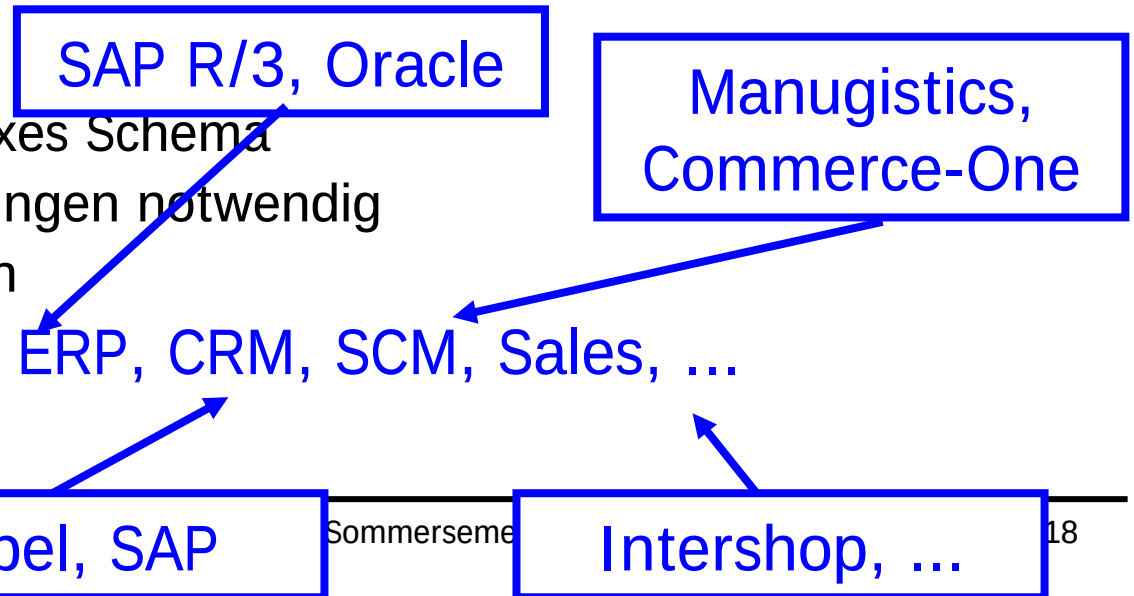
Unterschiedliche Philosophien

- Enterprise DWH
  - Schemaintegration
- Analyseorientiertes DWH
  - Multidimensionale Modellierung

DWH als Enterprise Model

- Idee: DWH enthält alle Unternehmensdaten
  - Schema muss Unternehmen komplett abdecken
  - Konzeptionelles Enterprise Model als Grundlage der Unternehmens-DV
- Nutzen
  - Angleichung von Unternehmensabläufen
  - Computergestützter Zugriff als alle Unternehmensdaten und -prozesse
- Probleme
  - Extrem komplexes Schema
  - Häufige Änderungen notwendig
  - Unklarer Nutzen

➢ Scheitert meist: ERP, CRM, SCM, Sales, ...



Schemaintegration

- Gegeben: Menge Quellen Q_i mit Schema S_i
- Gesucht: Schema $S = \cup S_i$
  - Das muss eine „semantische UNION“ sein
- Aber: Heterogenitäten
  - Datenmodelle: OO, Relational, IMS, ...
  - Schematisch: Klassen, Relationen, Attribute, Werte, ...
  - Semantik: Homonyme, Synonyme, ...
  - Syntax: Formate, Einheiten, Sprache, ...
- Viele Vorschläge, wenig erfolgreiche Verfahren
- Schemaintegration praktisch nicht automatisierbar

➢ Halbautomatische, vorschlagorientierte Systeme

Schemaintegration 2

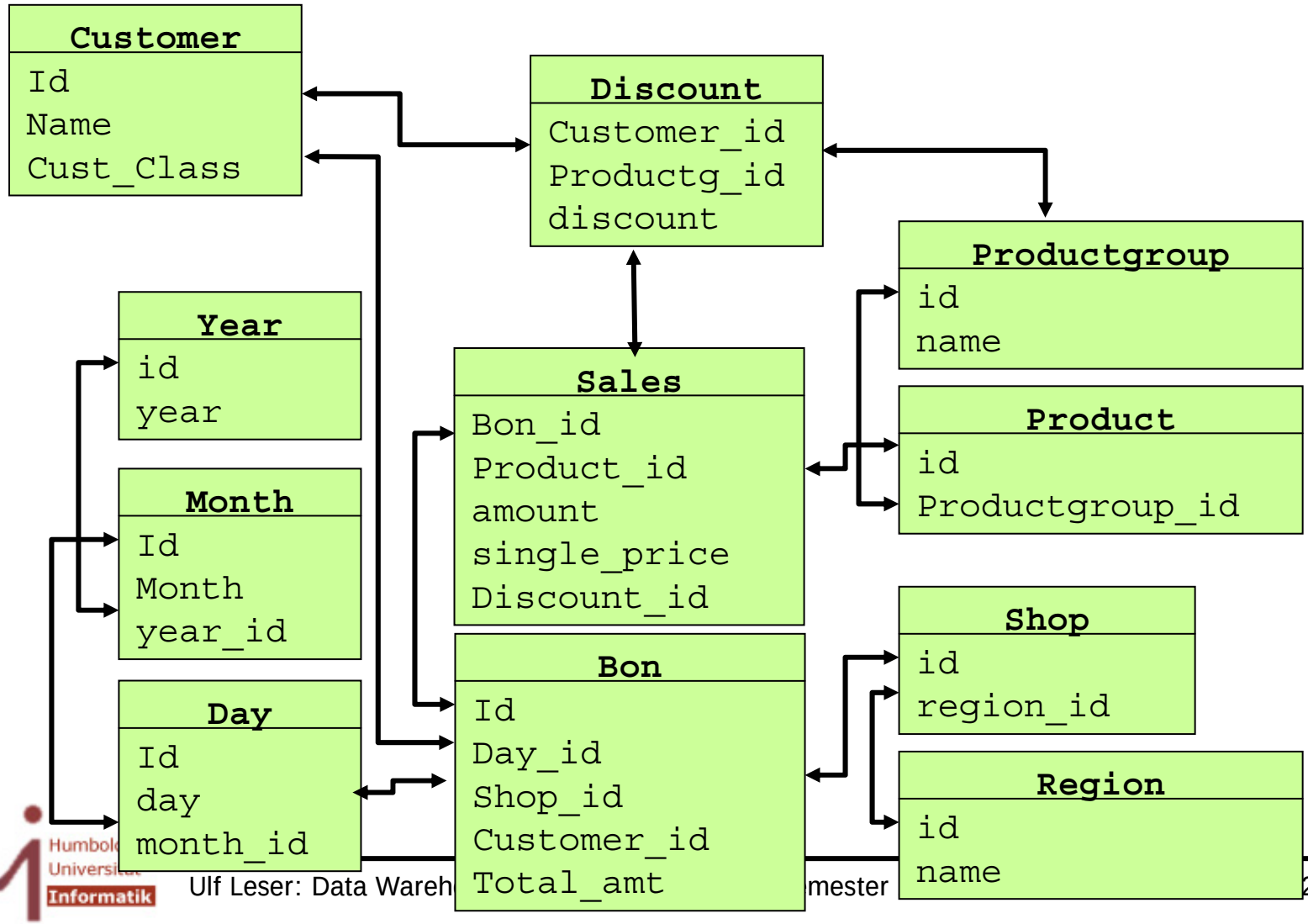
- Hauptproblem: Semantik von Schemaelementen
  - Relationale Schema extrem semantikarm
    - Nur Relationen und Attribute
  - Was speichert die Relation A20RR?
  - Was speichert das Feld Kunde.Name?
    - Vorname? Nachname? Titel? ...
  - Was ist Umsatz?
    - Brutto? Netto? Nach Abzug Rabatte? Nach Abzug Steuern? ...
- Aktives Forschungsfeld: Schema Matching
  - „Raten“ von Korrespondenzen
  - Data Mining: Vergleich von Werteverteilungen und –bereichen
  - Ausnutzung der Struktur der Schemata

Analyseorientiertes DWH

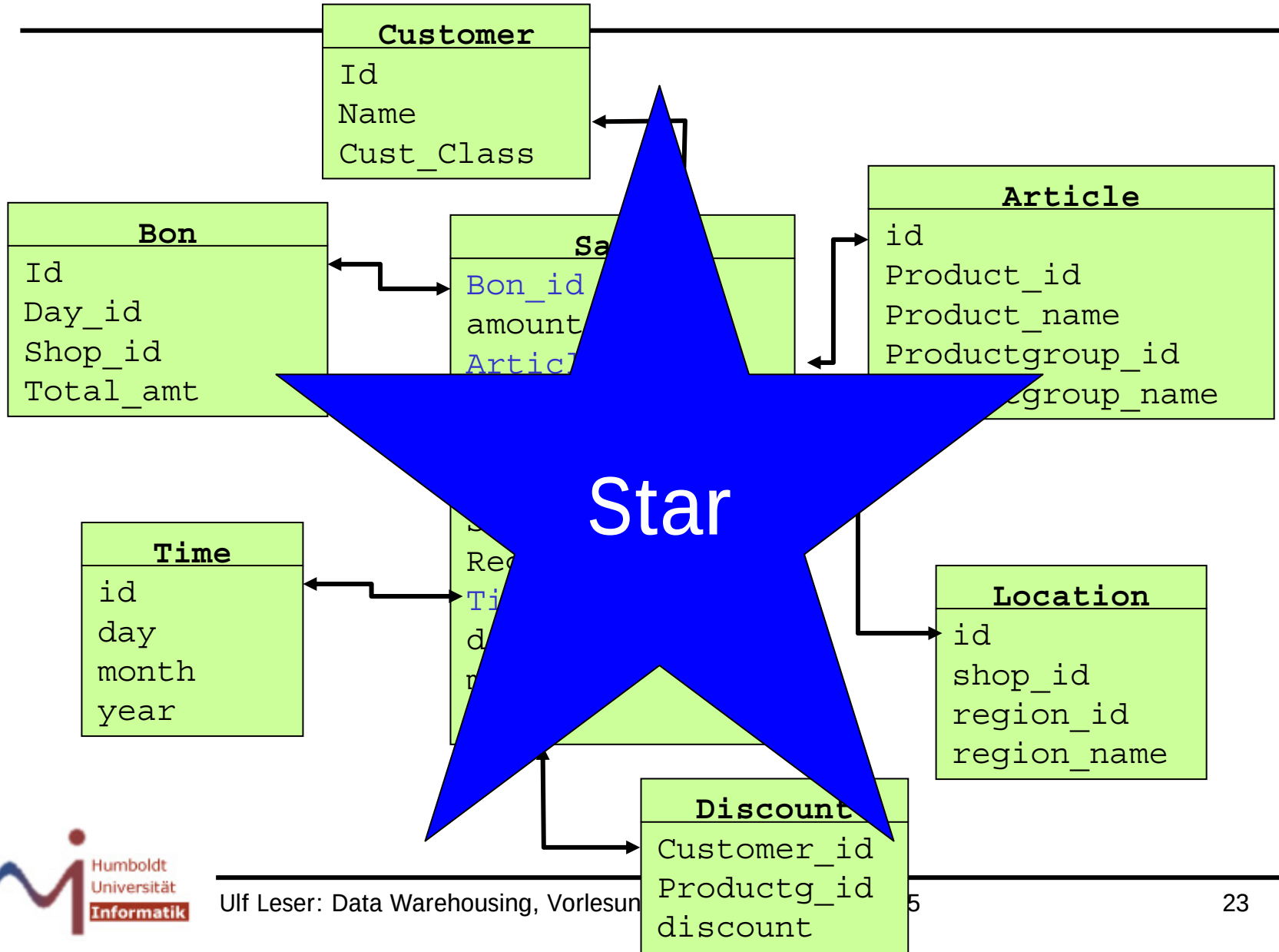
- Klassische Datenmodellierung
  - Ziele: Redundanzvermeidung / Integritätswahrung / hoher Durchsatz bei nebenläufigem Zugriff
  - Normalformen, Fremdschlüssel, Satzsperren
  - Für Lesen und Schreiben geeignet
- Ergebnis
  - Viele Relationen, unübersichtliches Schema
  - Viele Joins in (fast) allen Queries notwendig
  - Optimieren schwierig: Viele Pläne, genaue und aktuelle Statistiken notwendig
  - Joins lenken vom Analyseziel ab – man möchte lieber mit Begriffen des Geschäftsprozesses umgehen

➢ Multidimensionale Modellierung

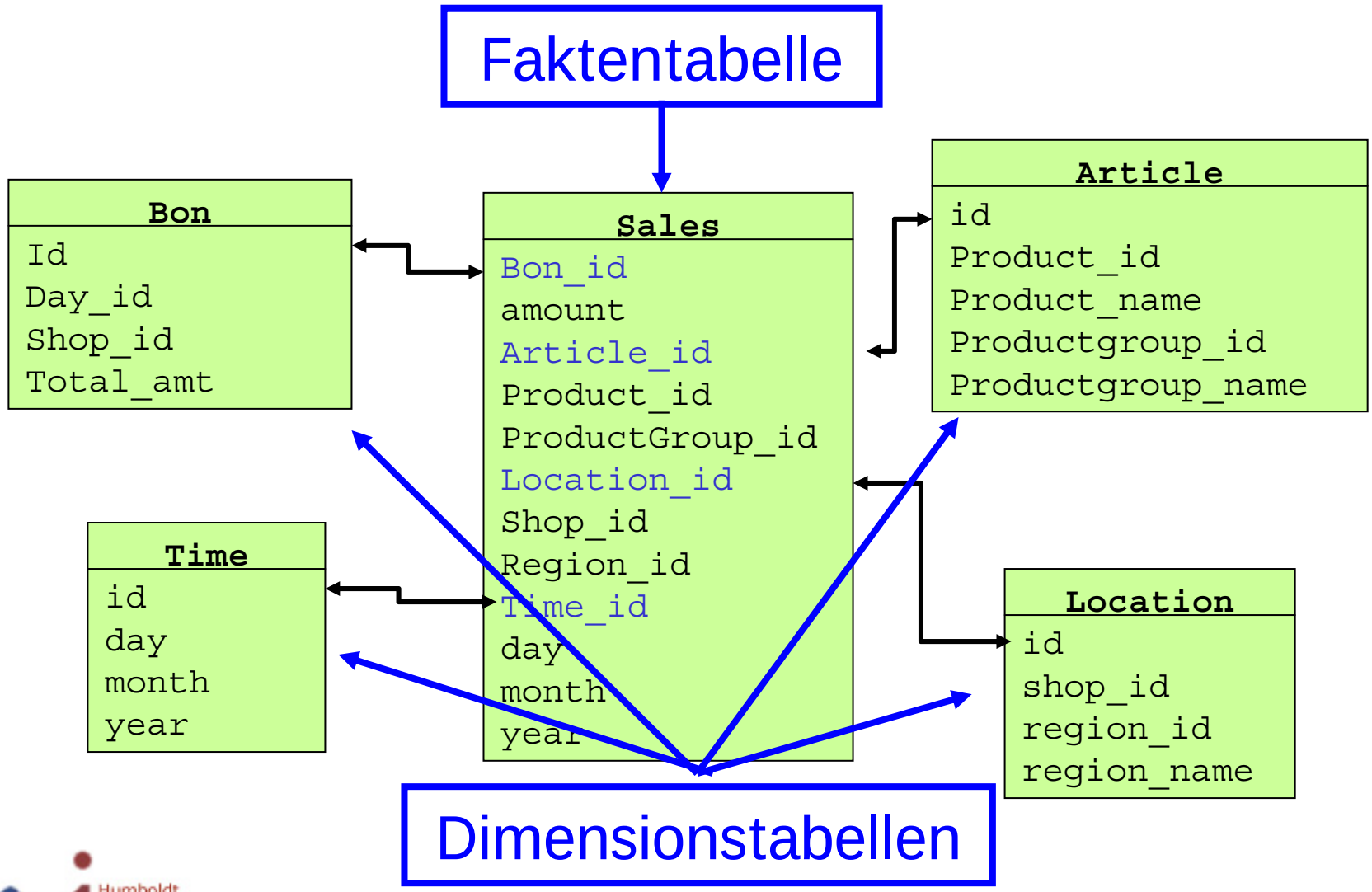
Beispiel: Normalisiertes Schema



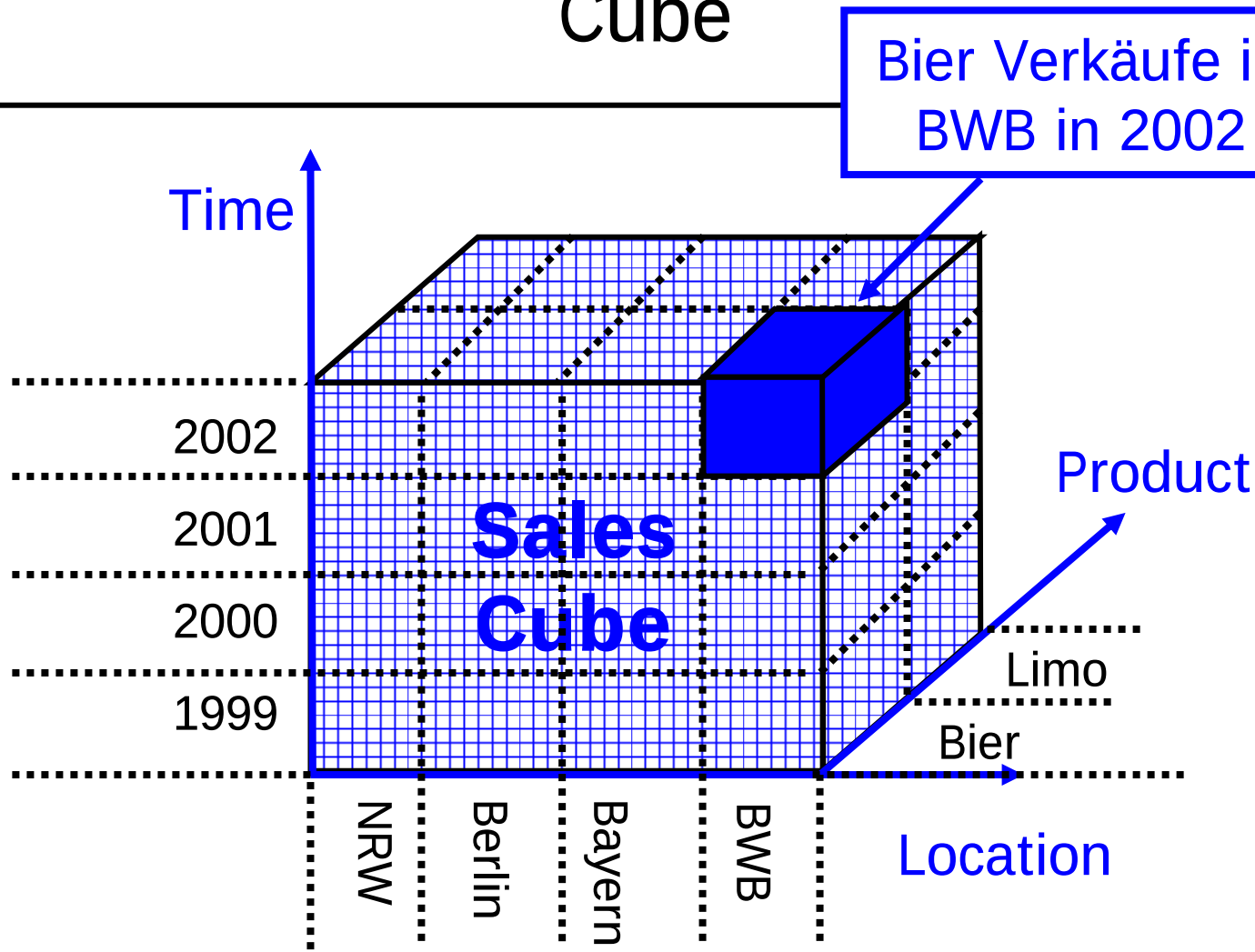
Multidimensionales Schema



Beispiel –2-

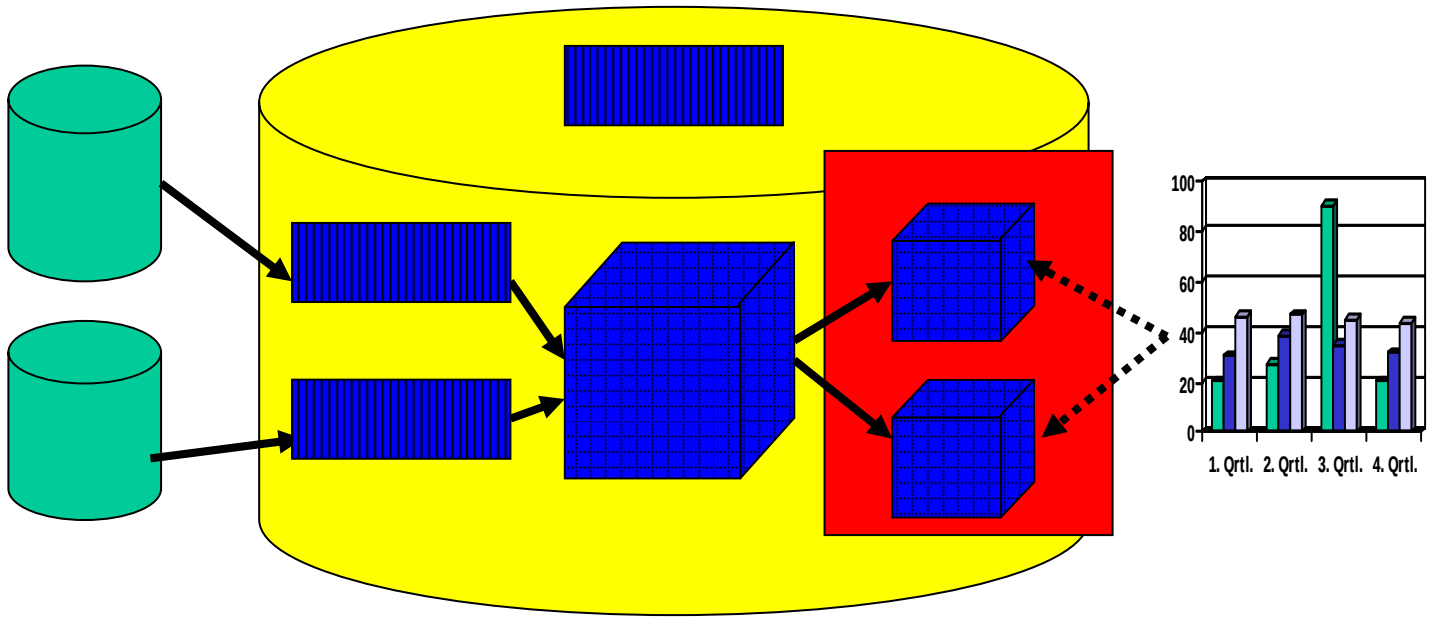

Faktentabelle
Bon
Id
Day_id
Shop_id
Total_amt
Sales
Bon_id
amount
Article_id
Product_id
ProductGroup_id
Location_id
Shop_id
Region_id
Time_id
day
month
year
Article
id
Product_id
Product_name
Productgroup_id
Productgroup_name
Time
id
day
month
year
Location
id
shop_id
region_id
region_name
Dimensionstabellen

Cube



Cube -> Hypercube: Bon / Lieferant / Kunde / ...

4. Abgeleitete Sichten



Abgeleitete Sichten

- Analysten benötigt spezielle Daten
  - Aggregiert
    - Alle Verkäufe in Norddeutschland nach Lieferanten
    - Alle Verkäufe nach Niederlassung und Produkten
  - Ausgewählt
    - Alle Verkäufe in Niederlassung X
    - Alle Verkäufe von Lieferant Y
- Probleme bei Auswertung auf Cube
  - Wiederholte Durchforstung sehr großer Datenbestände notwendig
  - Hohe Detailstufe des Cubes für viele Anfragen nicht notwendig

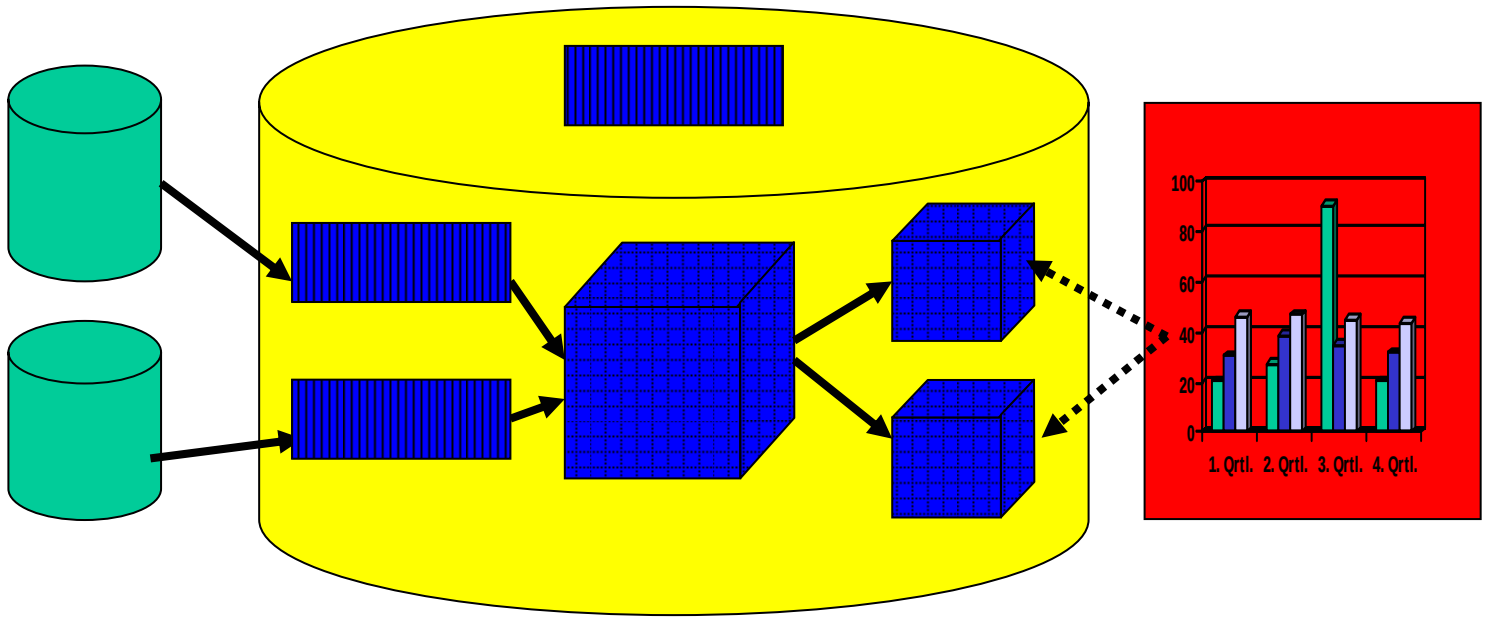
➢ Vorab-Erstellung von abgeleiteten Daten

➢ Prä-aggregierte, angereicherte und gefilterte Sichten

Abgeleitete Sichten – Weitere Themen

- Aktualität der Sichten
  - Asynchrone / synchrone Aktualisierung
  - Manuelle / automatische Aktualisierung
  - ➢ „Materialized Views“
- Verwendung der Sichten
  - Materialisierte Aggregation nach Produkten verwendbar für Aggregation nach Produktgruppen?
  - Materialisierte Aggregation nach Wochen verwendbar für Aggregation nach Monaten?
  - ➢ „Answering Queries using Views“

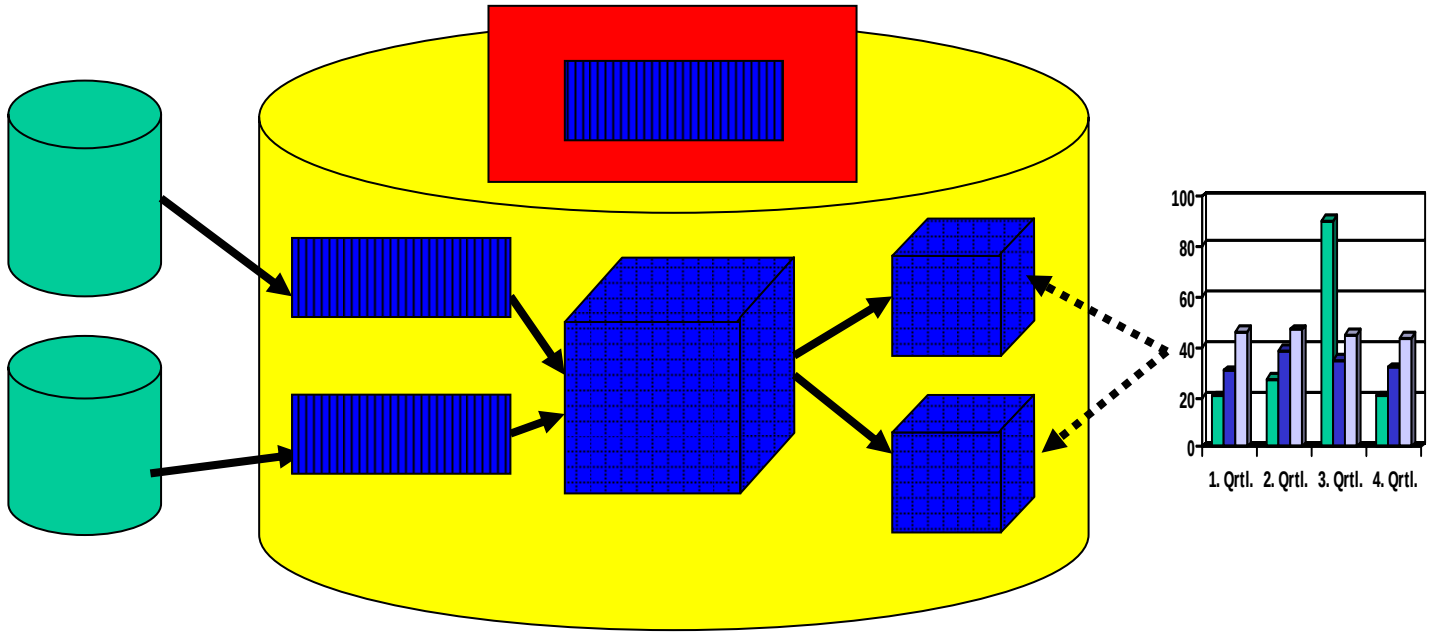
Datenanalyse



5. Analysewerkzeuge

- Hier nicht Thema
  - Siehe Datenanalyse, Data Mining, Statistik, ...
- OLAP Werkzeuge
  - Häufig proprietäre Systeme, eigene (geheime) Indexstrukturen
  - Abgesetzte Datenhaltung: OLAP auf dem Laptop
    - Thema mobile Datenbanken – Aktualisierung in flexiblen Abständen
  - Excel
- Funktionalität
  - Grafische Werkzeuge
  - Interaktive Datenauswahl, Filtering, Chaining, ...
  - Navigation, spez. im Cube
  - Präsentation: Grafiken, Tabellen, Reports, ...
- 70-80% aller Analysen sind Standardreports

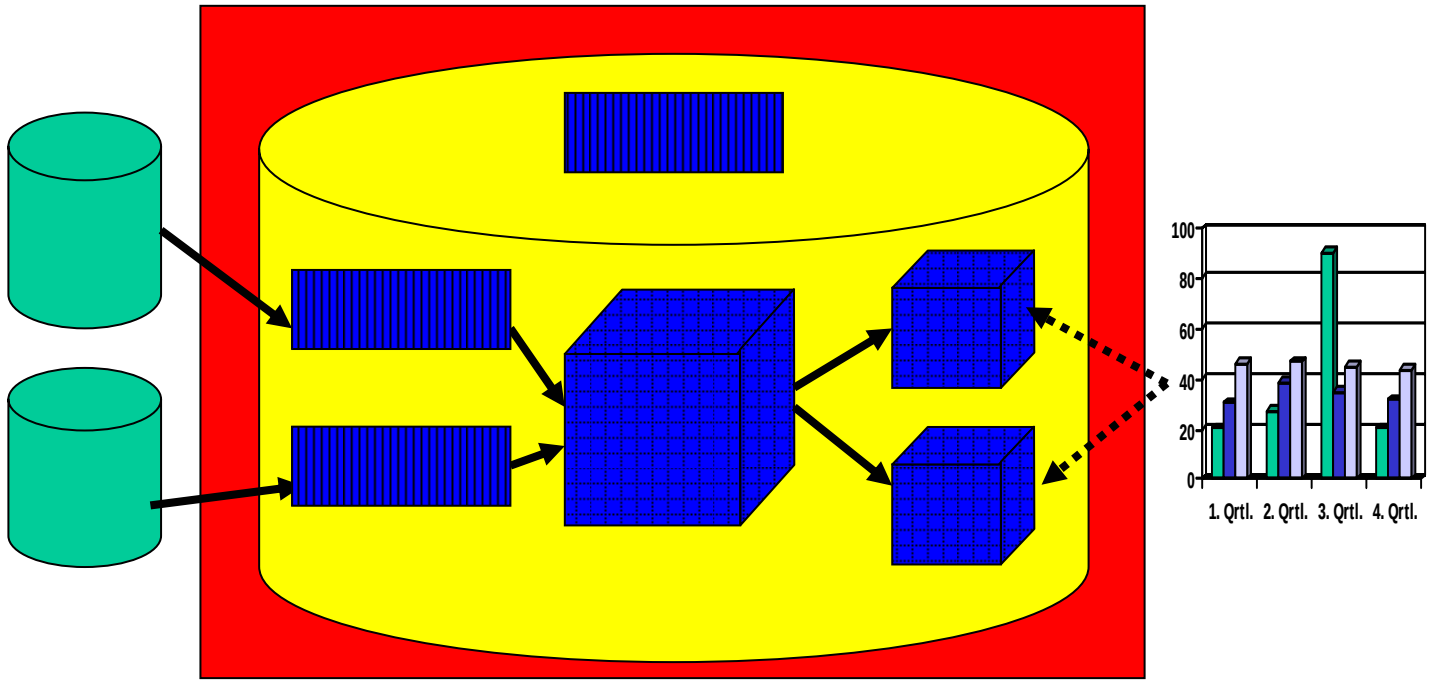
6. Metadatenrepository



Metadatenrepository

- „... identified as key success factor in DWH ...“
- Erweiterung der Datenbank Systemkatalogs
- Speicherung aller DWH relevanten Metadaten
  - Quellbeschreibungen, Datentypen, Prozessbeschreibungen, Schema, Zugriffsgruppen, Sichtdefinitionen, Skripte, Autoren, Versionskontrolle, Konfigurationsmanagement, ...
- Ziele
  - Nachvollziehbarkeit der Prozesse
    Wer, wann, was? Wie aktuell sind meine Daten?
  - Vermeidung von Fehlinterpretationen
    Welcher Zeitraum ist hier gemeint? % von was?
  - Technische Beschreibung des DWH
    Wer hat das programmiert? Was passiert, wenn ...?
- Produkte: Platinum CA, Microsoft, Oracle, ...
- Standards: IRDS, OIM, CWM, ...

7. DWH Manager



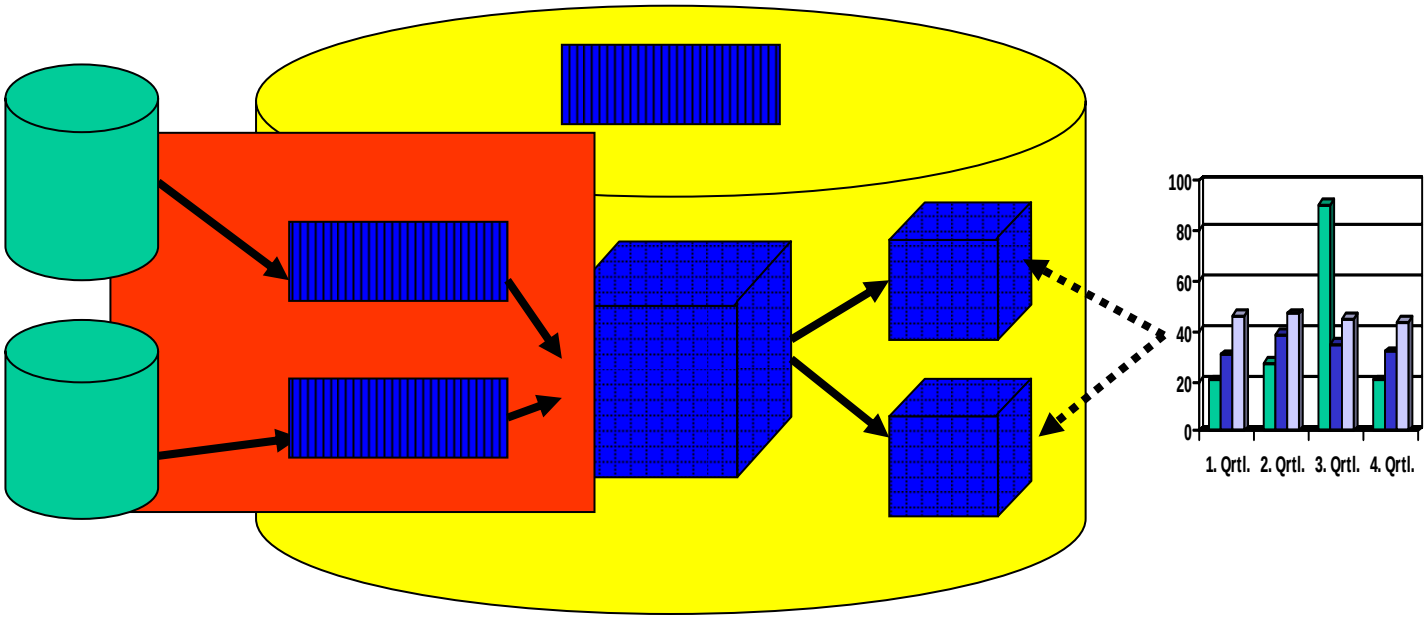
DWH Manager

- Häufig „virtuelle“ Komponenten
  - Steuerung aller Prozesse: ETL, Sichtaktualisierung,…
  - Verwaltung der Metadaten
  - Performancemonitoring und Betriebsunterstützung
  - Zugriffsschutz und Auditing
- Oftmals in Ausschnitten abgedeckt durch Standardwerkzeuge
  - DB-Administrationswerkzeuge
  - ETL Tools
  - Batchsysteme

Inhalt dieser Vorlesung

- Architektur
- Komponenten
- Prozesse

8. ETL



- Extraction
- Transformation
- Load

ETL - Extraktion

- Aufgabe
  - Filtern der „richtigen“ Daten aus den Quellen
  - Bereitstellung der Datenfiles im gewünschten Format zum gewünschten Zeitpunkt am gewünschten Ort
  - Kontinuierliche Datenversorgung des DWH
- Prinzip: Producer - Consumer
  - Quelle informiert über Änderungen
  - DWH konsumiert Änderungen

Parameter von Extraktionskomponenten

- Extraktor
  - Komponente zum Extrahieren der Daten aus den Quellen
- Wann liefert der Extraktor die Daten?
  - Periodisch
  - Synchron
  - Ereignisgesteuert
- Welche Daten liefert der Extraktor?
  - Kompletten Datenbestand (Snapshot)
  - Alle Änderungen (Logfile)
  - Nettoänderungen zu festen Zeitpunkten (Snapshot-Diff)
- In welcher Art liefert der Extraktor die Daten
  - SQL Befehle (synchron/Logfile: Replication)
  - Flatfiles

ETL - Transformation

- Aufgabe
  - Umwandlung der Daten in eine „DWH-gerechte“ Form
- Form follows Function
  - Quellen: hoher Transaktionsdurchsatz
  - DWH: spezifische statistische Analysen
- Arten von Transformationen
  - Schematransformationen
  - Datentransformationen
- Transformationen möglich an zwei Stellen
  - Transformation der Quell-Extrakte in Load-Files
  - Transformation von Staging-Area nach Basis-DB

Schematransformationen

- 1 Welt – 100 Anwendungen – 1000 Schema
  - Unterschiedliche Auffassungen
  - Unterschiedliche Anforderungen
  - Unterschiedliche Historie
- Unterschiedliche Datenmodelle
  - Relationales Modell
  - Objektorientierte Modelle (UML)
  - Satzorientierte Formate (Cobol)
  - Hierarchische Formate (IMS, XML)
- Unterschiedliche Modellierung
  - Was ist Relation, was Attribut, was Wert ?
  - Schlüssel

Datentransformationen

- Syntax von Werten
  - Datum: 20. Januar 2003, 20.01.2003, 1/20/03
  - Codierungen: „1: Adr. unbekannt, 2: alte Adresse, 3: gültige Adresse, 4: Adr. bei Ehepartner, ...“
  - Sprache
  - Abkürzungen/Schreibweisen: Str., strasse, Straße, ...
- Datentypen, Semantik
  - Datentypen: Real, Integer, String
  - Genauigkeit, Feldlänge, Nachkommastellen, ...
  - Skalen: Noten, Temperatur, Längen, Währungen,...

Transformationen beim Laden der Daten

- Extraktion der Daten aus Quelle mit einfachen Filtern
  - Spaltenauswahl, Keine Reklamationen, ...
- Erstellen eines LOAD Files mit einfachen Konvertierungen
  - Zahl – String, Integer – Real, ...
- Transformation meist zeilenorientiert
- Später Benutzung des DB-spezifischen LOADERs

```
read_line
parse_line
if (f[10]=2 & f[12]>0)
  write(file, f[1], string(f[4]), f[6]+f[7],...
...
bulk_upload( file)
```

Transformationen in Staging Area

- Benutzt SQL
- Effiziente mengenorientierte Berechnungen möglich
- Vergleiche über Zeilen hinaus möglich
  - Schlüsseleigenschaft, Namensduplikaterkennung, ...
- Vergleiche mit Daten in Basisdatenbank möglich
  - Duplikate, Plausibilität (Ausreißer), Konsistenz (Artikel-Ids)
- Typisch: Tagging von Datensätzen durch Prüf-Regeln

```
UPDATE sales SET price=price/MWST;
UPDATE sales SET cust_name=
  (SELECT cust_name FROM customer WHERE id=cust_id);
...
UPDATE sales SET flag1=FALSE WHERE cust_name IS NULL;
...
INSERT INTO DWH
  SELECT * FROM sales WHERE f1=TRUE & f2=TRUE & ...
```

ETL - Laden

- Aufgabe
  - Effizientes Einbringen der neuen Daten in das DWH
- Techniken
  - SQL – Satzbasiert
    - Standardschnittstellen: Embedded SQL, JDBC, ...
    - Einzelne Operationen oder proprietäre Erweiterungen
      - Array Insert
    - Beachtung und Aktivierung aller Datenbankverfahren
      - Trigger, Indexaktualisierung, Concurrency, ...
  - BULK Loader Funktionen
    - DB-spezifische Erweiterungen zum Laden großer Datenmengen
  - Benutzung von Anwendungsschnittstellen
    - Bei manchen Produkten notwendig (SAP)

BULK Uploads

- Für große Datenmengen einzige ausreichend performante Schnittstelle
- Kritischer Prozess
  - LOAD füllt i.d.R. immer nur eine Tabelle
  - LOAD setzt eine Sperre auf die gesamte Tabelle
  - Während LOAD werden Integritätsconstraints, Trigger, Indexaktualisierung deaktiviert
    - Nach LOAD werden IC überprüft und Indexe aktualisiert
    - Trigger werden nicht ausgeführt
  - Update oder Insert ? (Upsert!)
- Performance von LOAD oft limitierender Faktor

Beispiel

Handelshaus, Daten einer Woche, 1 Filiale	
Laden mit voller Qualitätskontrolle	10 min
Laden mit partieller Datenverbesserung	2 min
Nur Laden	45 sec

Handelshaus, Daten einer Woche, 2000 Filiale	
Laden mit voller Qualitätskontrolle	330h = 14d
Laden mit partieller Datenverbesserung	67 h = 2,8d
Nur Laden	25h = 1d

Zusammenfassung

- Komponenten
  - Datenquellen
  - Staging Area
  - Basisdatenbank
  - Abgeleitete Sichten / Data Marts
  - Analysewerkzeuge
  - Metadatenrepository
  - Data Warehouse Manager
- Prozesse
  - ETL: Extraktion, Transformation, Load