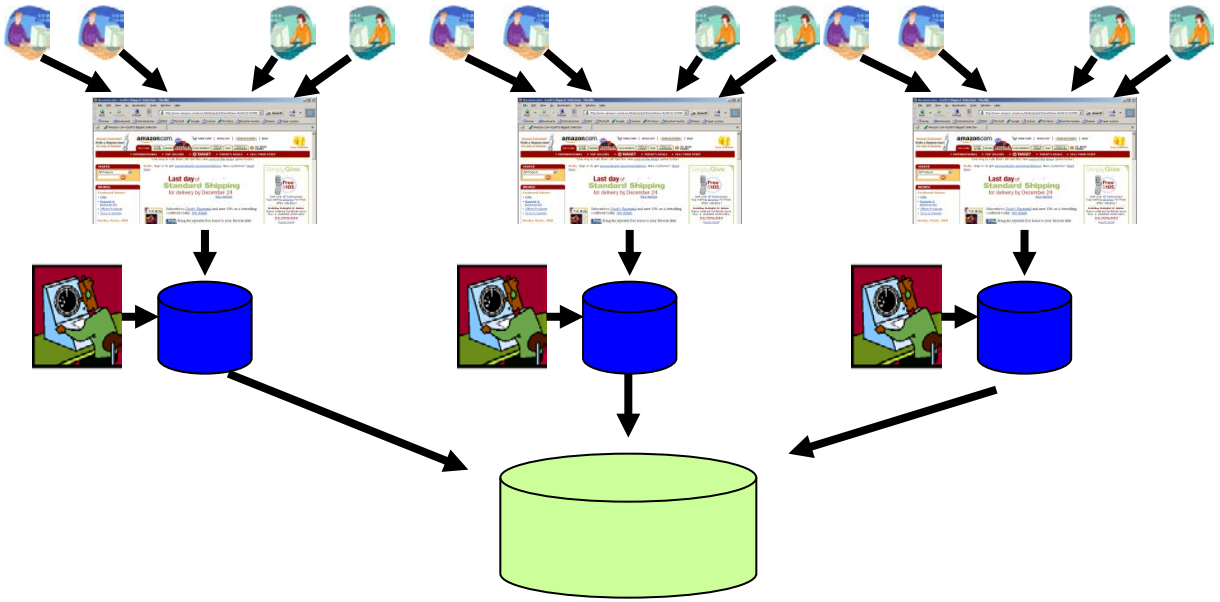
# Data Warehousing und Data Mining

## Architektur und Komponenten von Data Warehouses

Ulf Leser

Wissensmanagement in der Bioinformatik


# Data Warehouse



- Redundante Datenhaltung
- DWH kann unabhängig von Quellen entworfen werden
  - Optimiert für andere Arten von Anfragen
- Quellen werden nur bei periodischen Uploads belastet
- Heterogenität muss beim Upload abgefangen werden

# Inhalt dieser Vorlesung

- Definition & Einbettung
- Architektur & Komponenten
- ETL: Extraction, Transformation, Load

# Definition DWH

- *"A DWH is a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decisions"*
  [Inm96]

  - Subj-oriented: Verkäufe, Personen, Produkte, etc.
  - Integrated: Erstellt aus vielen Quellen
  - Non-Volatile: Hält Daten unverändert über die Zeit
  - Time-Variant: Vergleich von Daten über die Zeit
  - Decisions: Wichtige Daten rein, unwichtige raus

# Geschichte von DWH

- Managementinformationssysteme (MIS),
  Decision Support Systeme (DSS),
  Executive Information Systeme (EIS)
  - Seit den 60er Jahren
  - Feste, programmierte Reports
  - Doppelte Datenhaltung teuer – Downtimes für operative Sys.
  - Nur einfachste Analysemethoden (kein Data Mining ö.ä.)
  - Keine User-Interaktion, keine Ad-Hoc Queries

➢ Schattendasein

# Erfolg von DWH

- Top-Thema seit Mitte der 90er Jahre
- Voraussetzungen
  - Extreme Verbilligung von Plattenspeicherplatz
  - Relationale Modellierung: Anwendungsneutral
  - IT in allen Unternehmensbereichen (z.B. SAP R/3)
  - Vernetzung und Standardisierung (SQL)
  - Trend zur rationalen Entscheidungsfindung im Unternehmen auf Basis aller verfügbaren Daten
  - Schnellere Rechner ermöglichen interaktive Anwendungen
- Aber
  - Vision der vollständigen Integration scheitert bisher (immer wieder aufs neue)
  - Soziale versus technische Aspekte

# Betriebswirtschaftliche Sicht

- Ein DWH
  - Ermöglicht viele neue Fragen
  - Verbessert viele Antworten erheblich
- ... durch ...
  - Direkter Zugriff auf integrierte Daten
  - Übergreifende, vergleichende, historische Analysen
    - Produkte, Niederlassungen, Kunden, ...
  - Bessere Datenqualität
    - Fehlerminimierung, Ergänzung, Plausibilitätschecks
  - Anreicherung mit externen Daten
    - Externe Kundenprofile, geographische Daten, ...

# Informatische Sicht

- Operative Systeme
  - Kassensysteme, Bestellabwicklung, Lagerverwaltung
  - Viele Benutzer, kurze Transaktionen, einfache Queries
  - „Echtzeit“-Anforderungen
  - Kurzes Gedächtnis
  - OLTP (Online Transaction Processing)
- DWH
  - Sortimentplanung, Kapazitätsplanung, Marketing
  - Wenige(r) Benutzer, komplexe Queries, nur lesend
  - Zeitlich weniger kritisch
  - Historische Daten
  - OLAP (Online Analytical Processing)

# OLTP Beispiel

Login

```
SELECT pw FROM kunde WHERE login=„...“
UPDATE kunde SET last_acc=date, tries=0 WHERE
COMMIT
```

Willkommen

```
SELECT k_id, name FROM kunde WHERE login=„...“
SELECT last_pur FROM purchase WHERE k_id=...
COMMIT
```

Bestellung

```
SELECT av_qty FROM stock WHERE p_id=...
UPDATE stock SET av_qty=av_qty-1 where ...
INSERT INTO shop_cart VALUES( o_id, k_id, ...
COMMIT
```

Best. löschen

```
DELETE FROM shop_cart WHERE o_id=...
UPDATE stock SET av_qty=av_qty+1 where ...
COMMIT
```

# OLAP Beispiel

- Welche Produkte hatten im letzten Jahr im Bereich Bamberg einen Umsatzrückgang um mehr als 10%?
  - Welche Produktgruppen sind davon betroffen?
  - Welche Lieferanten haben diese Produkte?
- Welche Kunden haben über die letzten 5 Jahre eine Bestellung über 50 Euro innerhalb von 4 Wochen nach einem persönlichen Anschreiben aufgegeben?
  - Wie hoch waren die Bestellungen im Schnitt?
  - Wie hoch waren die Bestellungen im Vergleich zu den durchschnittlich. Bestellungen des selben Kunden in einem vergleichbaren Zeitraum?
  - Lohnen sich Mailing-Aktionen?
- Haben Zweigstellen einen höheren Umsatz, die gemeinsam gekaufte Produkte zusammen stellen ?
  - Welche Produkte werden überhaupt zusammen gekauft – und wo?

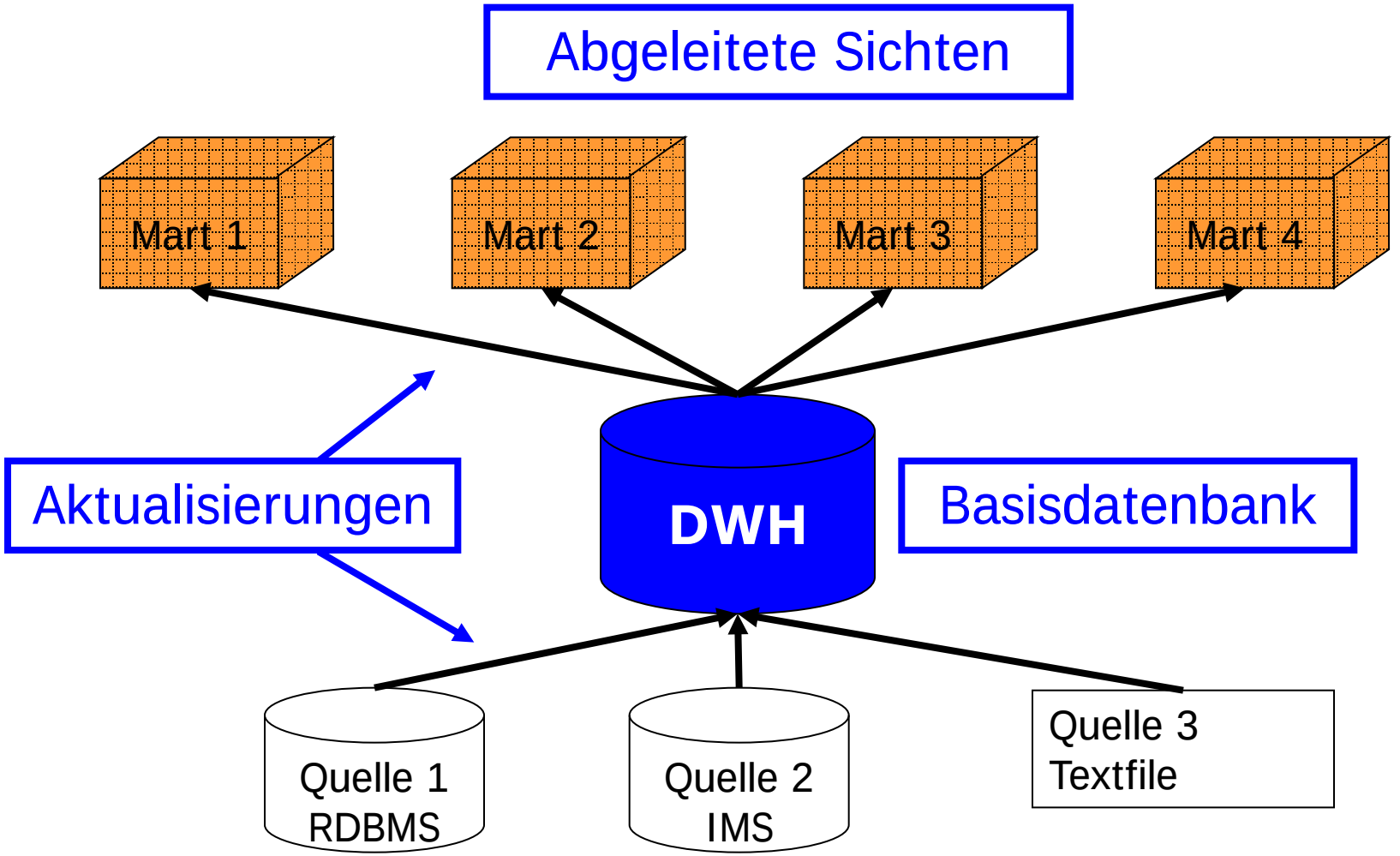
# OLAP versus OLTP

| | OLTP | OLAP |
|---|---|---|
| **Typische Operationen** | Insert, Update, Delete, Select | Select (Bulk-Inserts) |
| **Transaktionen** | Viele und kurz | Nur lesend |
| **Typische Anfragen** | Einfache Queries, Primärschlüsselzugriff, Schnelle Abfolgen von Selects/inserts/updates/ deletes | Komplexe Queries: Aggregate, Gruppierung, Subselects, etc.<br>Bereichsanfragen über mehrere Attribute<br>Aggregate mit komp. Rechnungen |
| **Daten pro Operation** | Wenige Tupel | Mega-/ Gigabyte |
| **Datenmenge in DB** | Gigabyte | Terabyte |
| **Datenart** | Rohdaten, häufige Änderungen, nur intern | Abgeleitete Daten, historisch & stabil, externe Daten |
| **Erwartete Antwortzeiten** | Echtzeit bis wenige Sekunden | I.A. nicht zeitkritisch (aber UI-Erlebnis) |
| **Modellierung** | Anwendungsorientiert | Themenorientiert |
| **Typische Benutzer** | Elektronische Systeme (Sachbearbeiter) | Management |

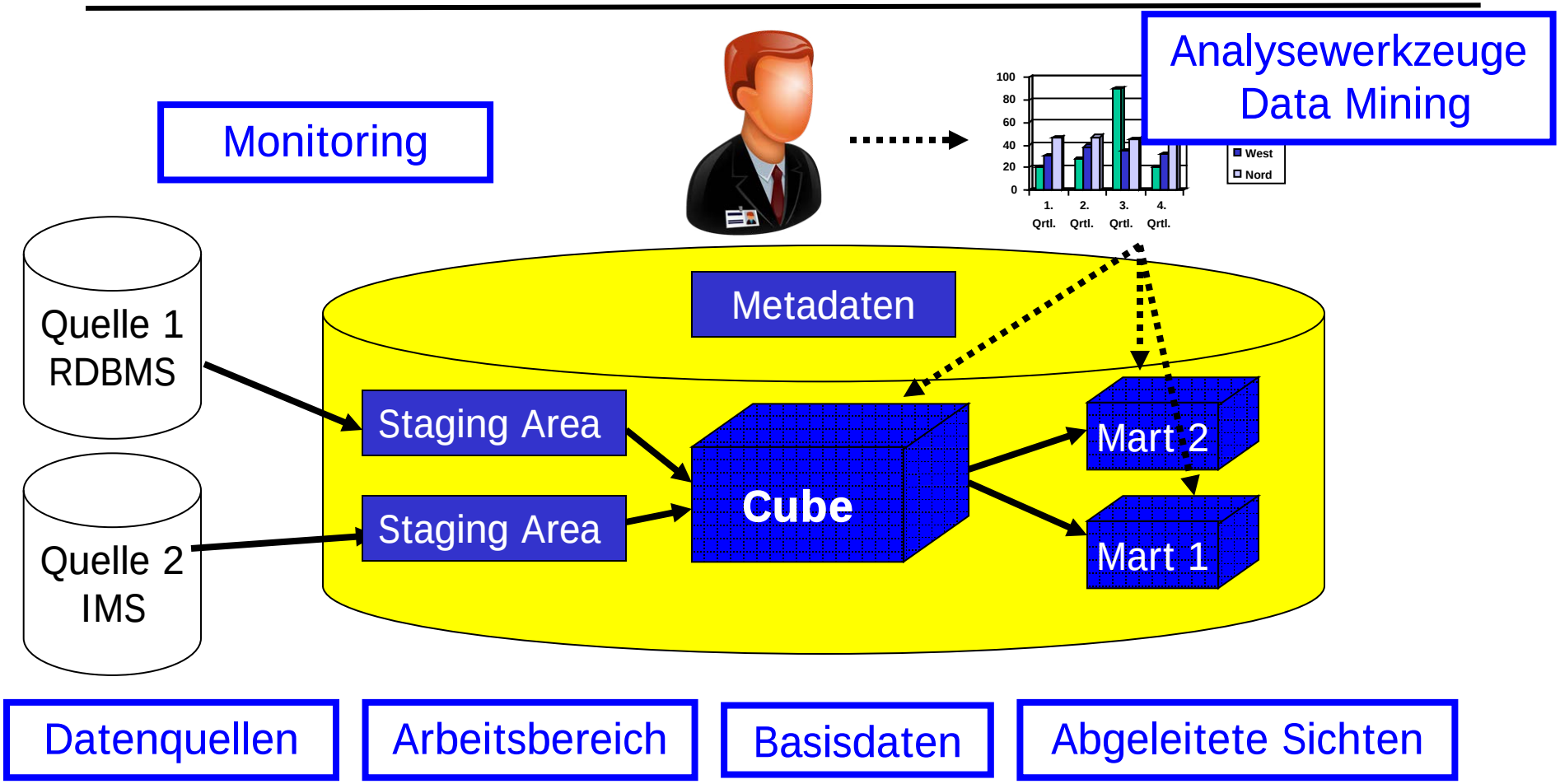
# Inhalt dieser Vorlesung

- Definition & Einbettung
- Architektur & Komponenten
- ETL: Extraction, Transformation, Load

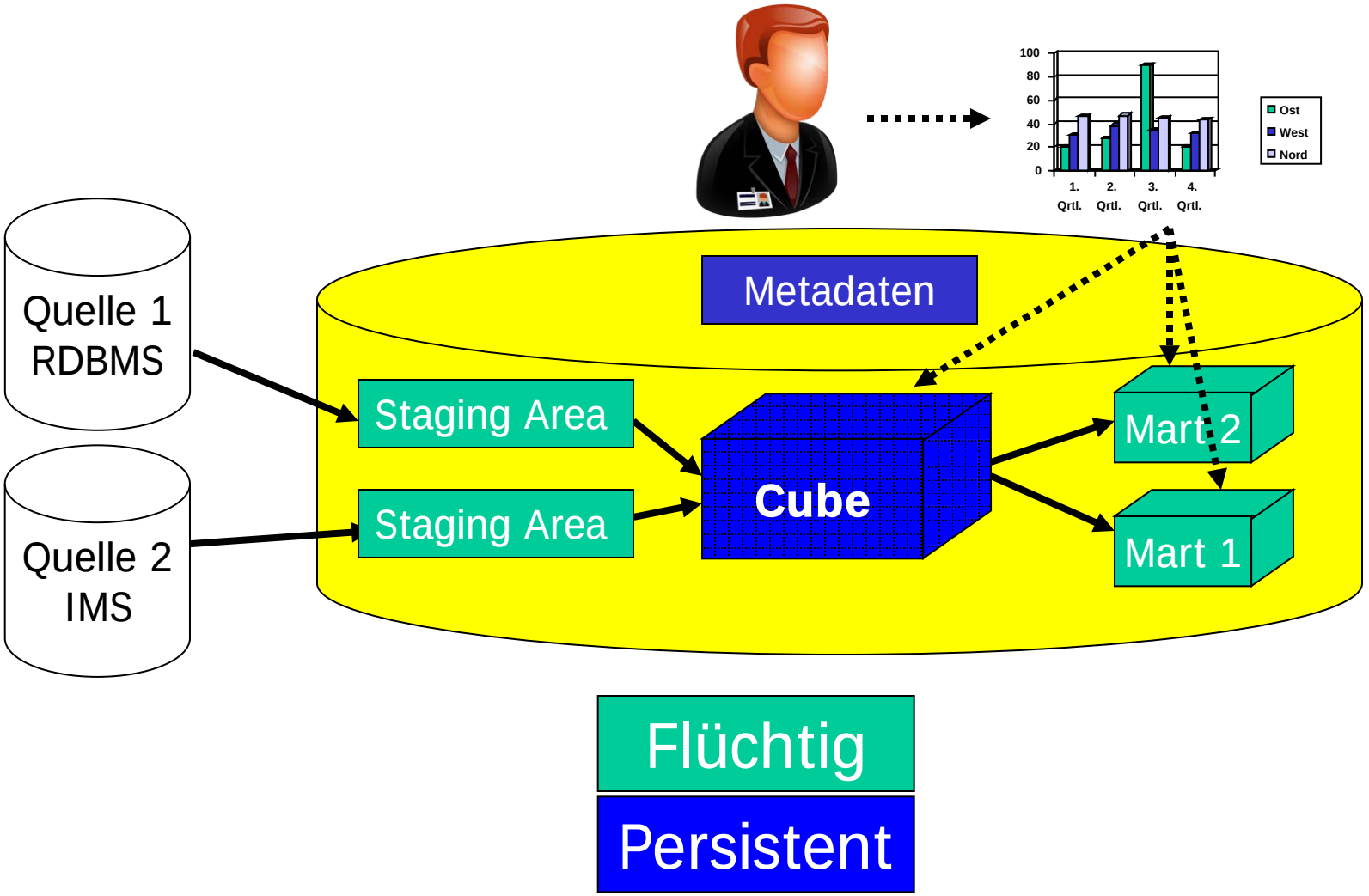
# Grobarchitektur: „Hubs and Spokes“



# Verfeinerung: DWH Architektur

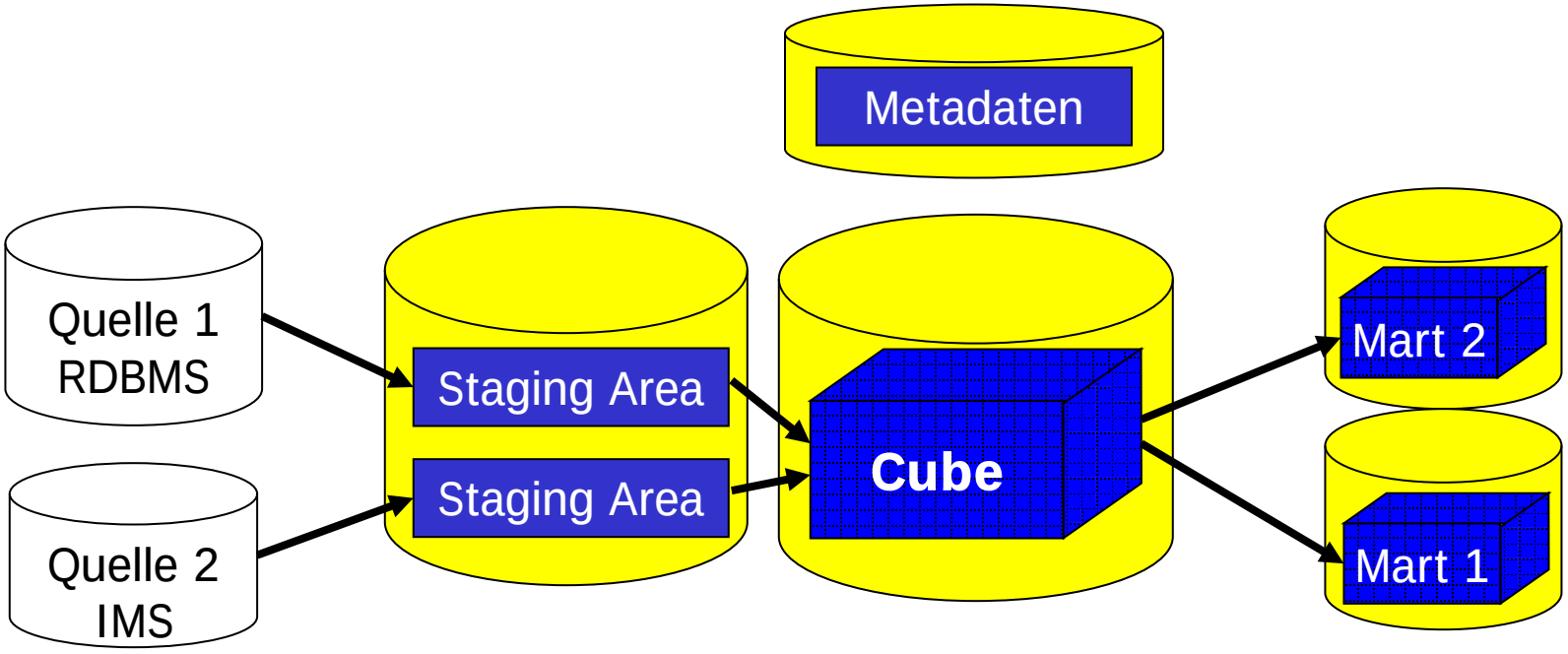


# Langlebigkeit

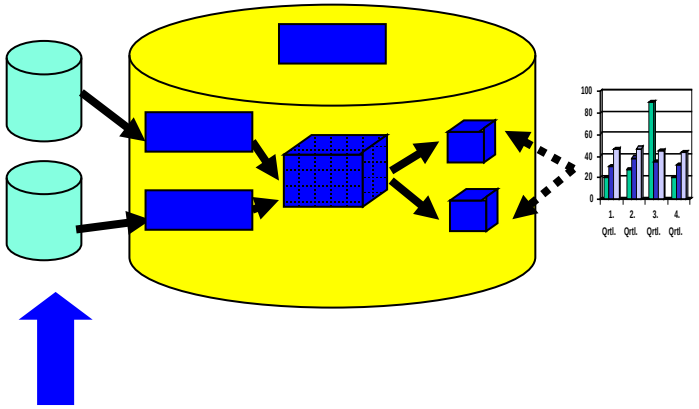


# Alternativen

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

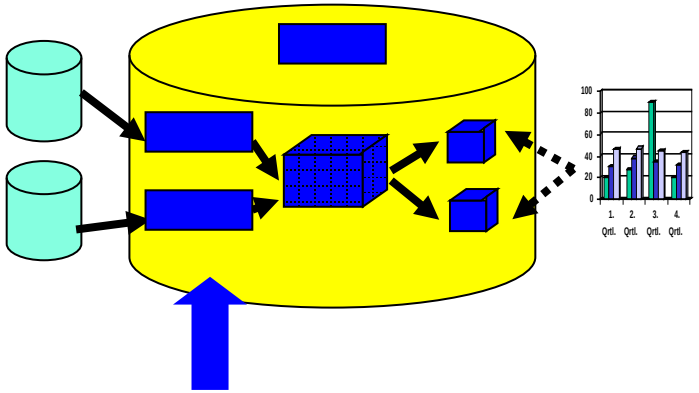


# Datenquellen



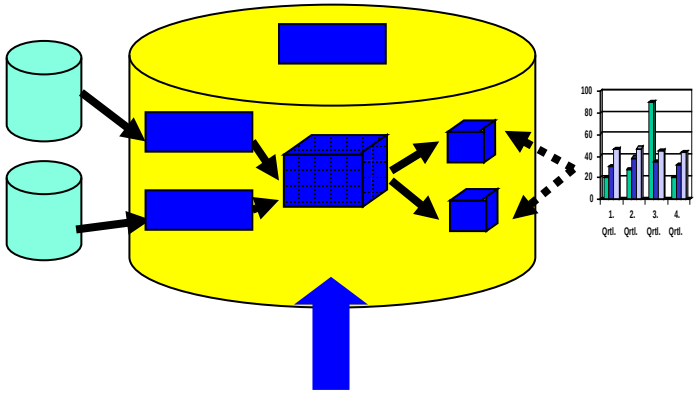
- Meist heterogen
  - Technisch: RDBMS, IMS, Access Mainframe, COBOL, Textfiles, ...
  - Logisch: Schema, Format, Repräsentation,...
  - Syntaktisch: Datum, Währung, ...
  - Rechtlich: Datenschutz (Kunden & Mitarbeiter)
- Zugriff
  - Push: Quelle erzeugt (regelmäßig) Extrakte
  - Pull: DWH fragt Änderungen ab
- Individuelle Behandlung notwendig

# Arbeitsbereich



- Temporärer Speicher
  - ETL Arbeitsschritte effizienter implementierbar
    - Mengenoperationen, SQL
  - Zugriff auf Basisdatenbank möglich
  - Vergleich zwischen Datenquellen möglich
  - Filtern: Nur einwandfreie Daten in Basisdatenbank übernehmen

# Basisdatenbank (Cube)

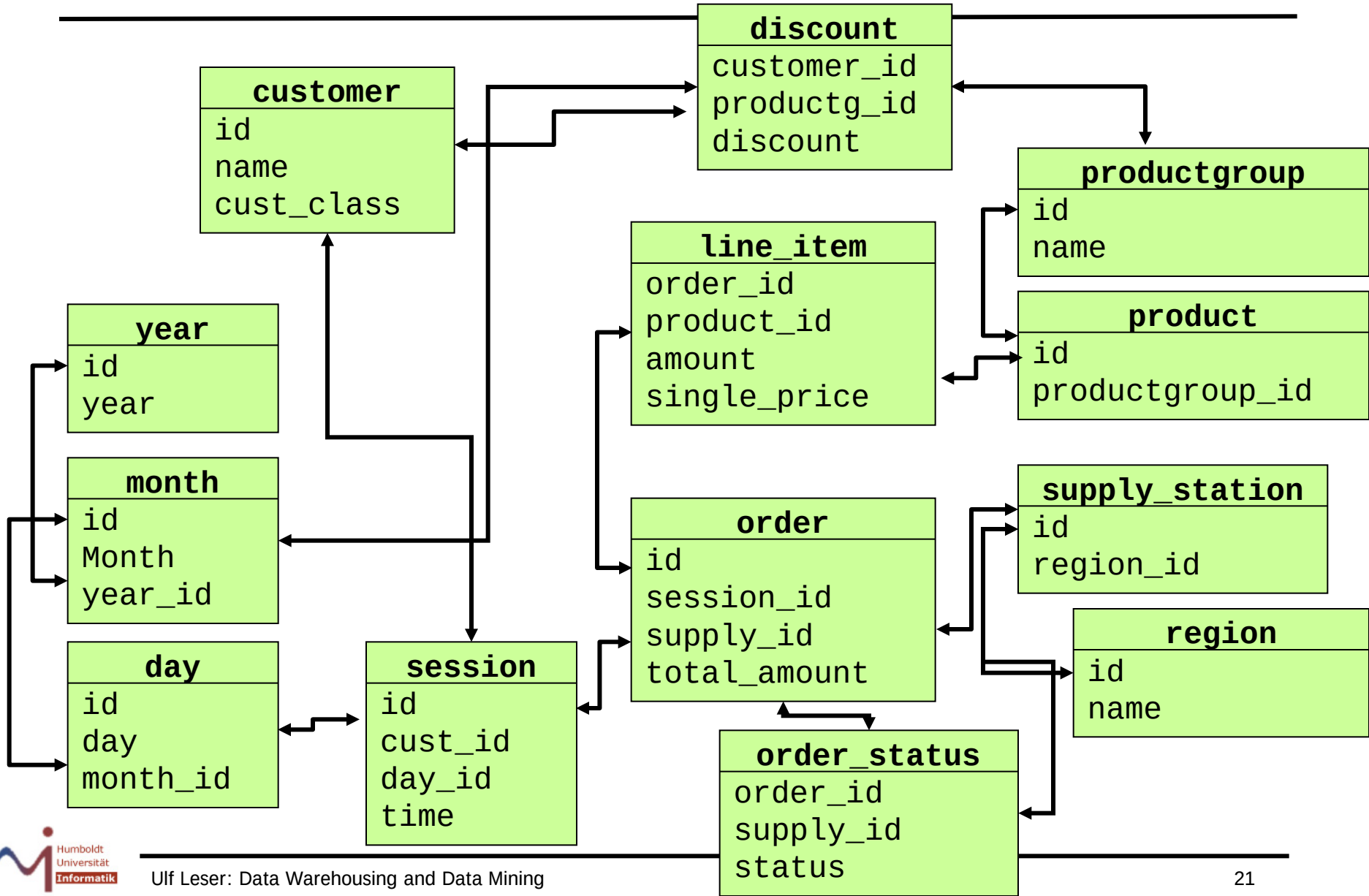



- Zentrale Komponente des DWH
- Speichert Daten in feinster Auflösung
  - Einzelne Verkäufe, einzelne Bons, ...
- Historische Daten
- Große Datenmengen
  - Spezielle Modellierung
  - Spezielle Optimierungsstrategien

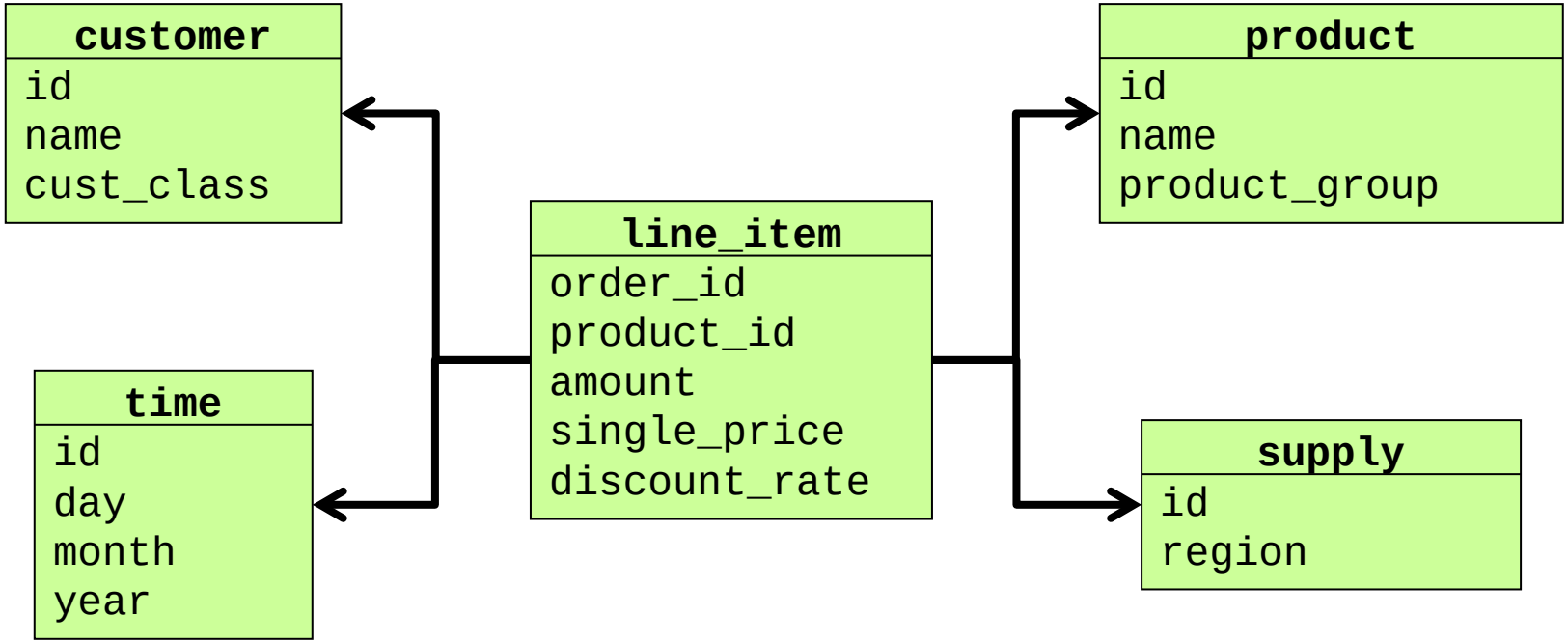
# Analyseorientiertes DWH

- Klassische Datenmodellierung
  - Orientiert sich an technischen Anforderungen
    - Ziele: Redundanzvermeidung / Integritätswahrung / hoher Durchsatz bei nebenläufigem Zugriff
    - Normalformen, Fremdschlüssel, Satzsperren
    - Für Lesen und Schreiben geeignet
  - Viele Relationen, unübersichtliches Schema
  - Für Informatiker
- Im DWH: Multidimensionale Modellierung
  - Orientiert sich am Unternehmensziel
    - Ziel: Verbesserung des Geschäfts (Umsatz, Gewinn, ...)
    - Modellierung von „Business Entities“ (Produkte, Kunde, ...)
    - Read-only
  - Übersichtliches, leicht verständliches Modell
  - Für Nicht-Informatiker

# Beispiel: Normalisiertes Schema

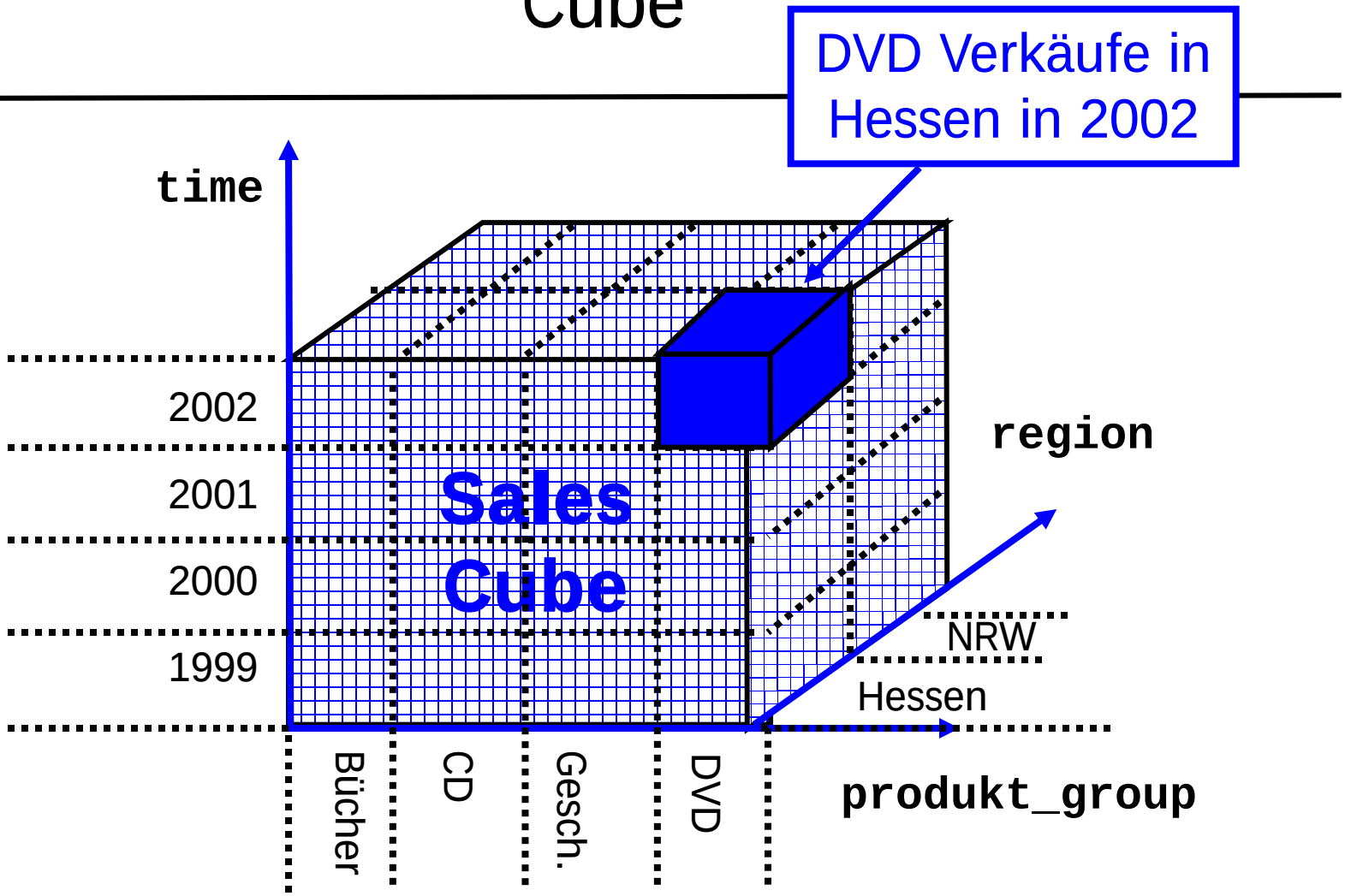


# Multidimensionales Schema

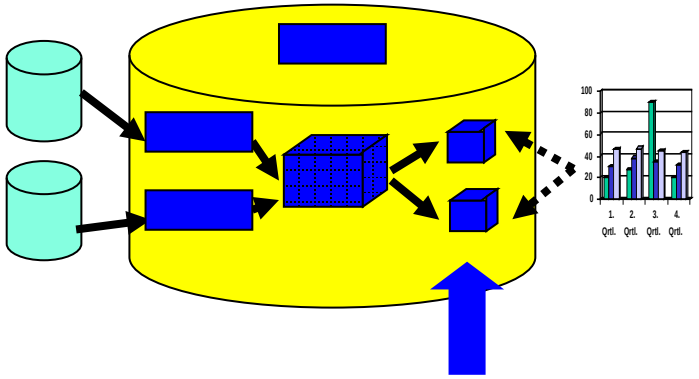



- Technische Informationen raus (Session)
- Vereinfachung / Denormalierung (discount_rate)
- Fokus auf Verkäufe (line_item)

# Cube

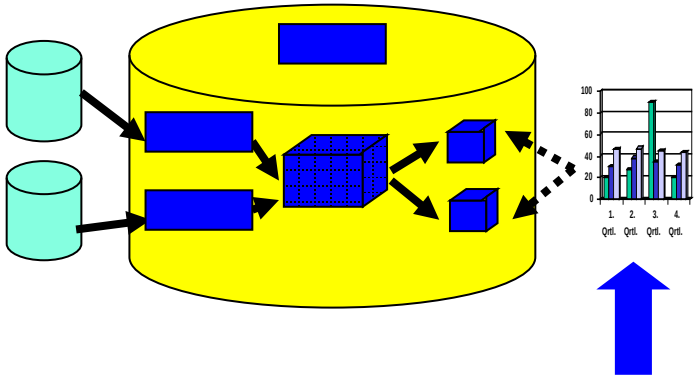

DVD Verkäufe in Hessen in 2002
time
2002
2001
2000
1999
Sales Cube
region
NRW
Hessen
Bücher
CD
Gesch.
DVD
produkt_group

# Abgeleitete Sichten



- Typische Anfragen auf Cube
  - Aggregiert und gruppiert
    - Verkäufe nach Monat und Lieferant
    - ... nach Niederlassung und Produkt
- Probleme bei Auswertung
  - Queries scannen sehr große Datenbestände
  - Hohe Detailstufe des Cubes für viele Anfragen nicht notwendig
- Vorhalten abgeleiteter Daten
  - Technisch: Materialisierte Sichten
    - Prä-aggregiert, angereichert, gefiltert, automatisch aktualisiert
  - „Data Marts“

# Datenanalyse

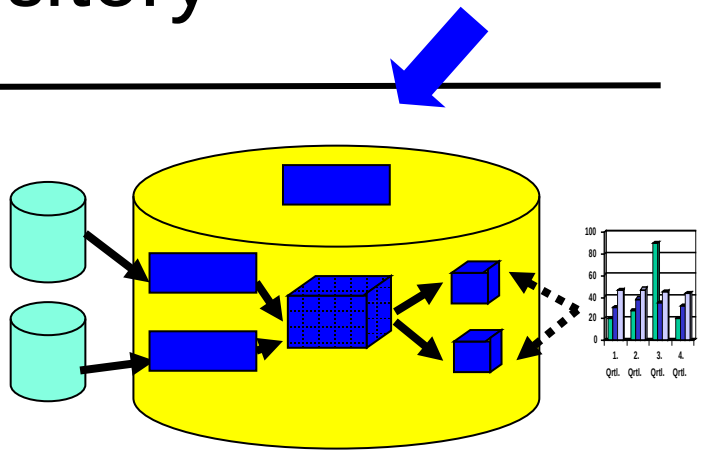


- Funktionalität
  - Grafische Oberflächen zur Navigation (Cube)
  - Interaktive Datenauswahl, Filtering, Chaining, ...
  - Präsentation: Grafiken, Tabellen, Reports, ...
  - Management: Zugriffsrechte, Scheduling, ...
- Standardreports versus Ad-hoc Anfragen
- Gutes UI verlangt sehr schnelle Antwortzeiten
- Viele kommerzielle Systeme
  - SAS, SPSS, BusinessObjects, Cognos, Excel, ...

# Data Mining

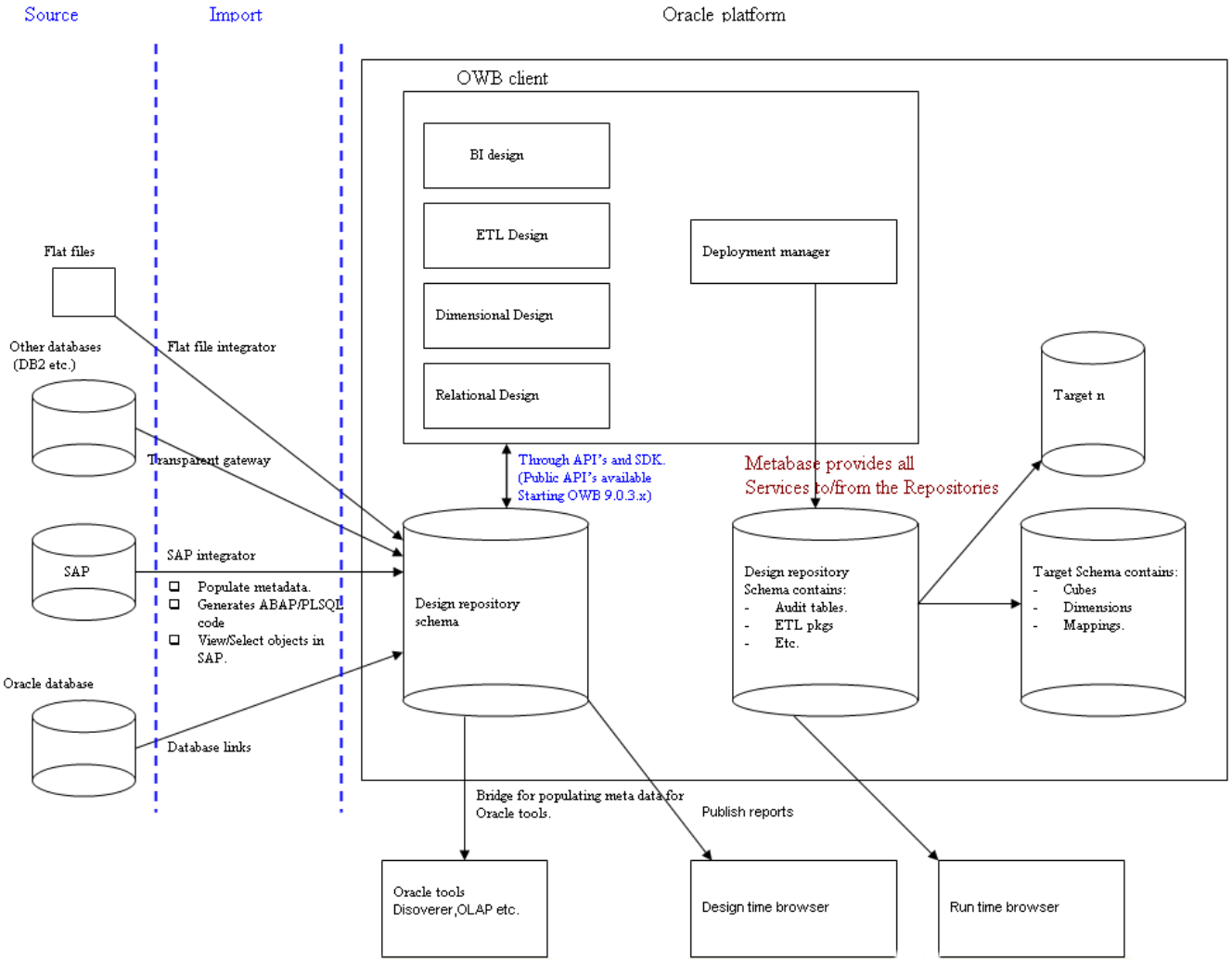
- „Finden verborgender, nicht-trivialer Informationen“
- Bereiche
  - Statistische Analyse
  - Maschinelle Lernverfahren
  - Knowledge Discovery in Databases (KDD)
- Suche nach Auffälligkeiten, Mustern, Regeln
  - Viele Kunden, die Windeln kaufen, kaufen auch Bier
- Suche nach Erklärungsmodellen
  - Modell: Abstraktion der Wirklichkeit
  - Korrelation versus Kausalität

# Metadaten-Repository

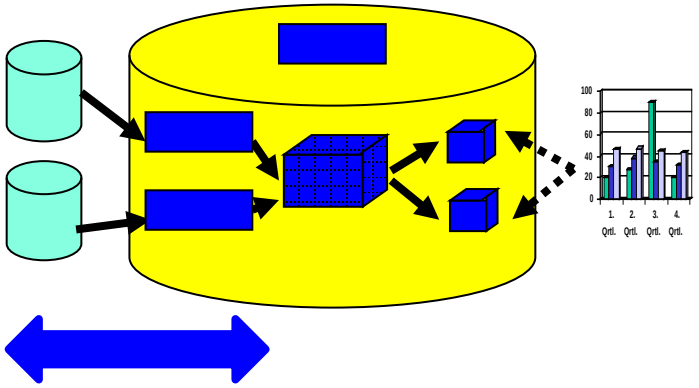


- „... key success factor in DWH ...“
  - Quellbeschreibungen, Datentypen, Skripte, Prozessbeschreibungen, Schema, Zugriffsgruppen, Sicht-definitionen, Versionskontrolle, Konfigurationsmanagement, ...
  - – Erweiterung des klassischen DB-Systemkatalogs
- Ziele
  - – Nachvollziehbarkeit der Prozesse
  - – Zentrale Steuerung der Prozesse
- Standards: IRDS, OIM, CWM, ...

# Oracle Warehouse Builder



# Inhalt dieser Vorlesung



- Definition & Einbettung
- Architektur & Komponenten
- ETL: Extraction, Transformation, Load

# ETL - Extraktion

- Filtern der „richtigen“ Daten aus den Quellen
  - Korrekt und relevant (für das Ziel des DWH)
- Bereitstellung der Datenfiles im gewünschten Format zum gewünschten Zeitpunkt am gewünschten Ort
- Kontinuierliche Datenversorgung des DWH
- Producer - Consumer
  - Quelle informiert über Änderungen
  - DWH konsumiert Änderungen

# ETL - Transformation

- Transformation der Daten in eine „DWH-gerechte“ Form
  - Schema, Format, Semantik
  - Laden soll so schnell wie möglich gehen
  - Erledigung vieler Teilschritte außerhalb des DWH
- Arten von Transformationen
  - Schema-/ Formattransformationen
  - Datentransformationen
- Transformationen möglich an zwei Stellen
  - Transformation der Quell-Extrakte in Load-Files
  - Transformation von Staging-Area nach Basis-DB

# ETL - Laden

- 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, Transaktionen, ...
  - BULK Loader Funktionen
    - DB-spezifische Erweiterungen zum Laden großer Datenmengen
  - Benutzung von Anwendungsschnittstellen
    - Bei manchen Produkten notwendig (SAP)

# 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 Filialen | |
|---|---|
| Laden mit voller Qualitätskontrolle | 330h = 14d |
| Laden mit partieller Datenverbesserung | 67 h = 2,8d |
| Nur Laden | 25h = 1d |