

# Data Warehousing und Data Mining

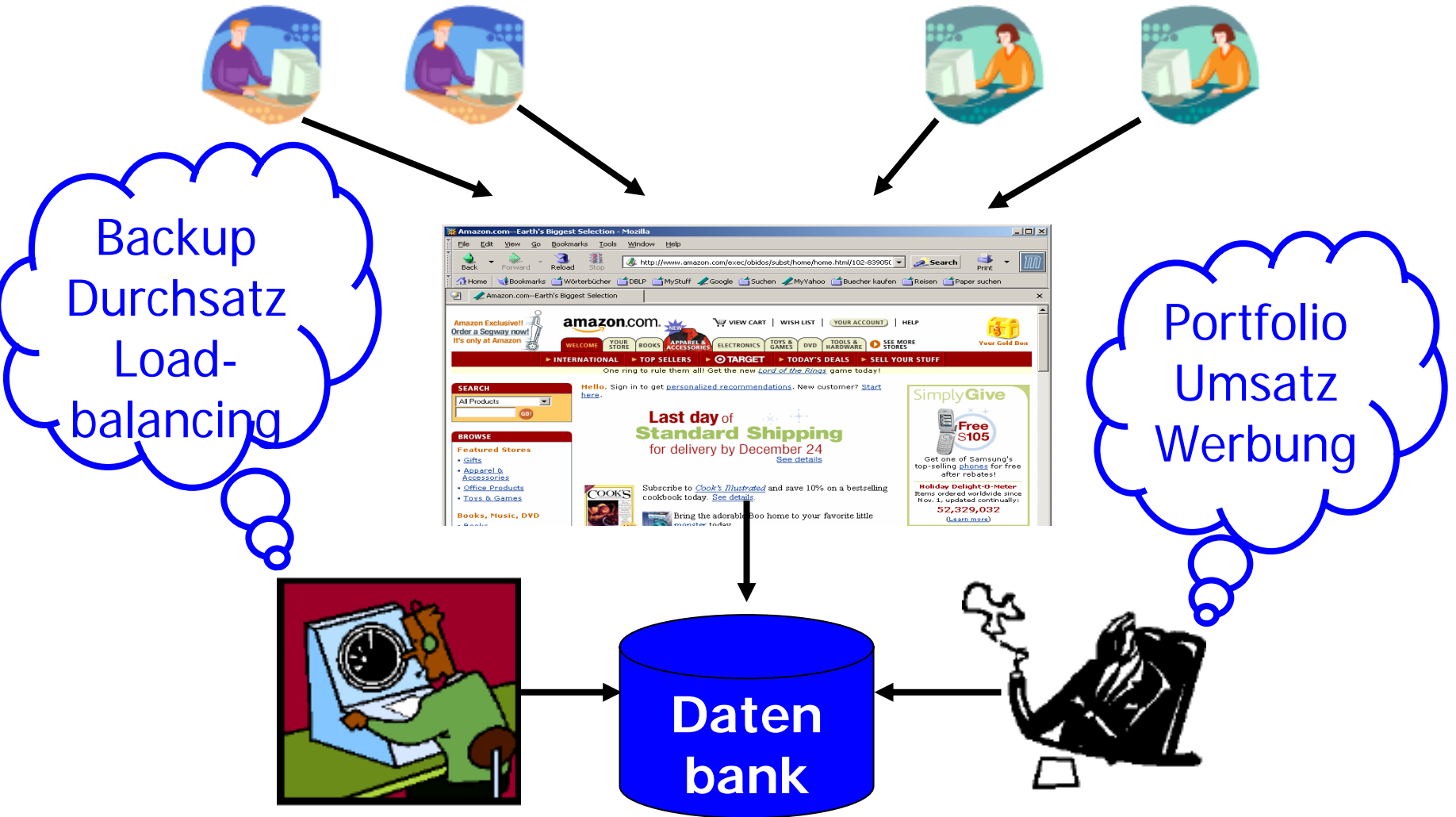
Architektur und Komponenten von  
Data Warehouses

Ulf Leser

Wissensmanagement in der  
Bioinformatik

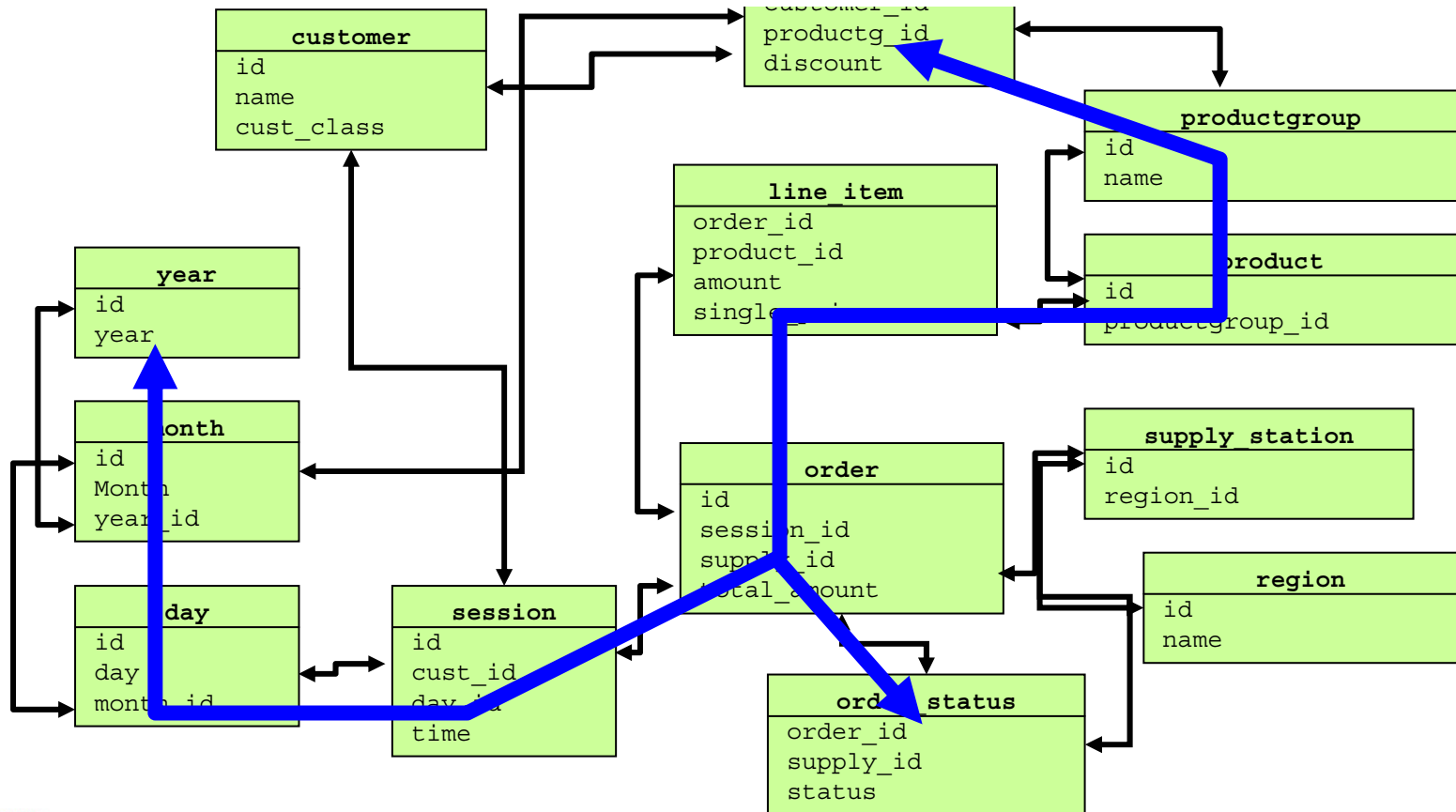


# Bücher im Internet bestellen



# Fragen eines Marketingleiters

Wie viele abgeschlossene Bestellungen haben wir jeweils im Monat vor Weihnachten, aufgeschlüsselt nach Produktgruppen und Promotion?



# Ergebnis

```
SELECT Y.year, PG.name, DI.disc, count(*)
FROM year Y, month M, day D, session S,
line_item I, order O, product P,
productgroup PG, discount DI,
order_status OS
WHERE M.year_id = Y.id and
D.month_id = M.id and
S.day_id = D.id and
O.session_id = S.id and
I.order_id = O.id and
I.product_id = P.id and
P.productgroup_id = PG.id and
DI.productgroup_id = PG.id and
O.id = OS.order_id
D.day < 24 and M.month = 12 and
OS.status = 'FINISHED'
PG.name, DI.discount
DI.discount
```

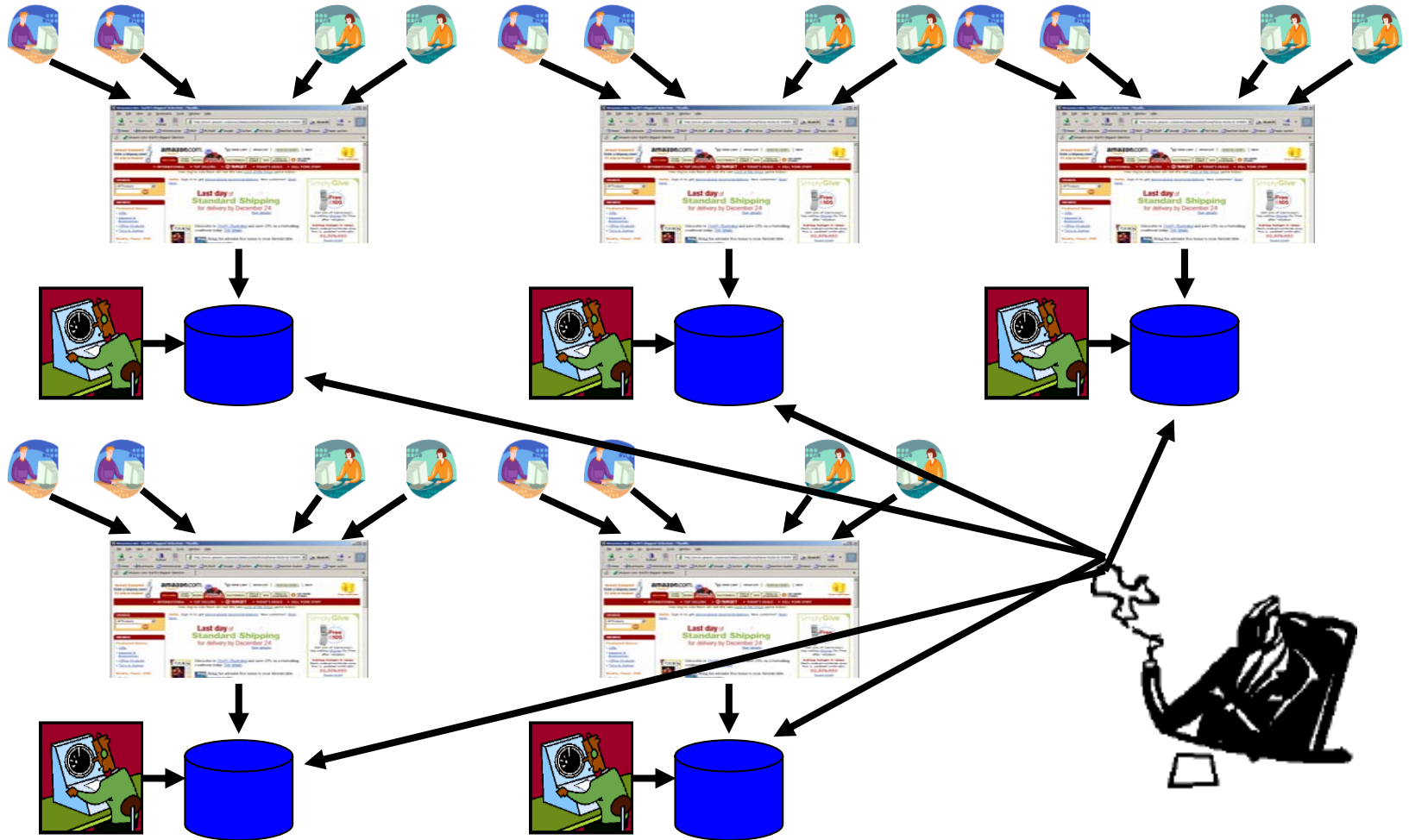
## 9 Joins

- year: 10 Records
- month: 120 Records
- day: 3650 Records
- session: 36.000.000
- order: 37.000.000
- line\_item: 72.000.000
- order\_status: 37.000.000
- product: 200.000
- productgroup: 100
- discount: 50

## Problem!

- Schwierig zu optimieren (Join-Order)
- Ja nach Ausführungsplan riesige Zwischenergebnisse
- Ähnliche Anfragen – ähnlich riesige Zwischenergebnisse

# In Wahrheit ...

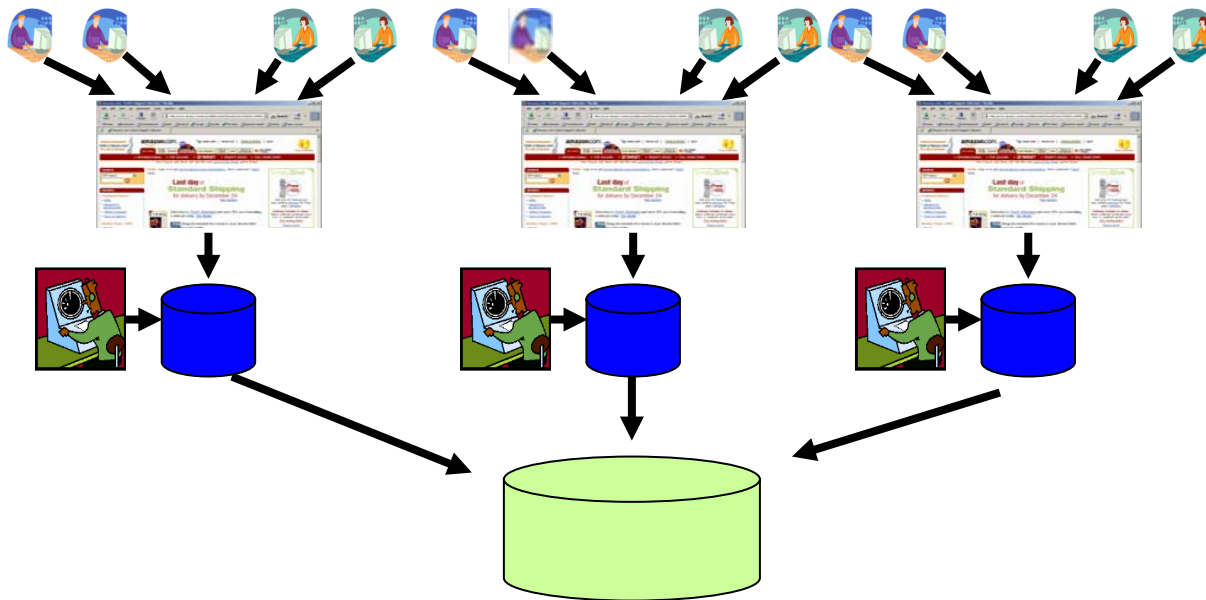


# Probleme

---

- Count über Union über verteilte Datenbanken?
  - **Heterogenitätsproblem**
    - Quellen werden Schemata verändern
    - Länderspezifischer Eigenheiten (MWST, Versandkosten, Sonderaktionen, ...)
    - Oftmals verborgene Änderungen in der Semantik der Daten
- Berechnung der Zwischenergebnisse bei jeder Anfrage?
  - **Datenmengenproblem**
    - Erfordert Transport großer Datenmengen durchs Netz
    - Historische Sicht - **Datenmengen wachsen immer weiter**
    - Operative Systeme brauchen die historischen Daten nicht
      - Ziel: Frühes löschen (abgeschlossene Bestellungen)
    - Manager brauchen viele der operativen Daten nicht
      - Ziel: Alles aufheben

# Besser: Data Warehouse



- Redundante Datenhaltung
- Spezielle Modellierung
- Transformierte und selektierte Daten
- Asynchrone Aktualisierung

# Definition DWH

---

- *A DWH is a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decision* [Inm96]
  - Subj-orient.: 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

# 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 jew. 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

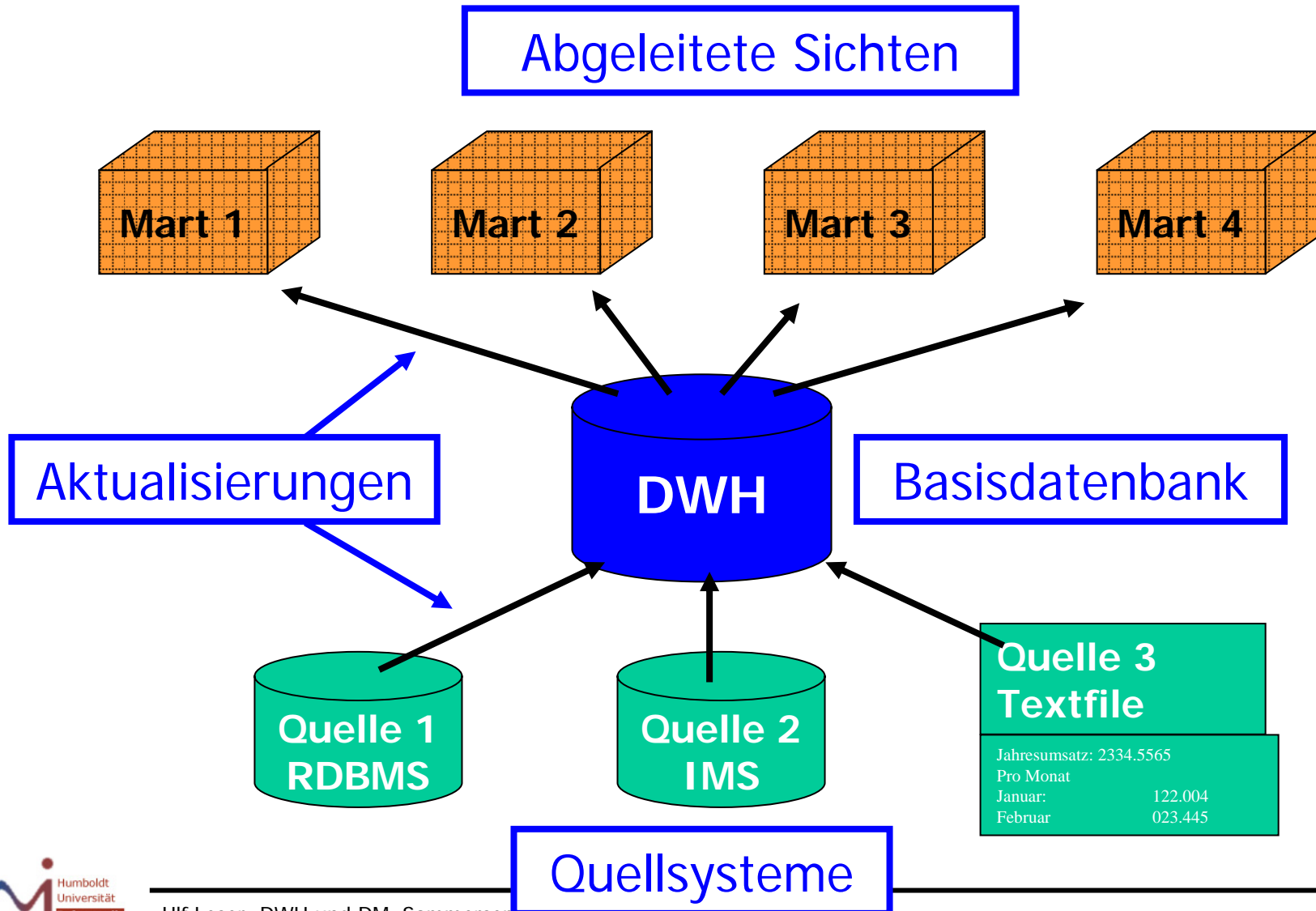
	OLTP	OLAP
Typische Operationen	Insert, Update, Delete, Select	Select Bulk-Inserts
Transaktionen	Viele und kurz	Lesetransaktionen
Typische Anfragen	Einfache Queries, Primärschlüsselzugriff, Schnelle Abfolgen von Selects/inserts/updates/deletes	Komplexe Queries: Aggregate, Gruppierung, Subselects, etc.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

# Inhalt dieser Vorlesung

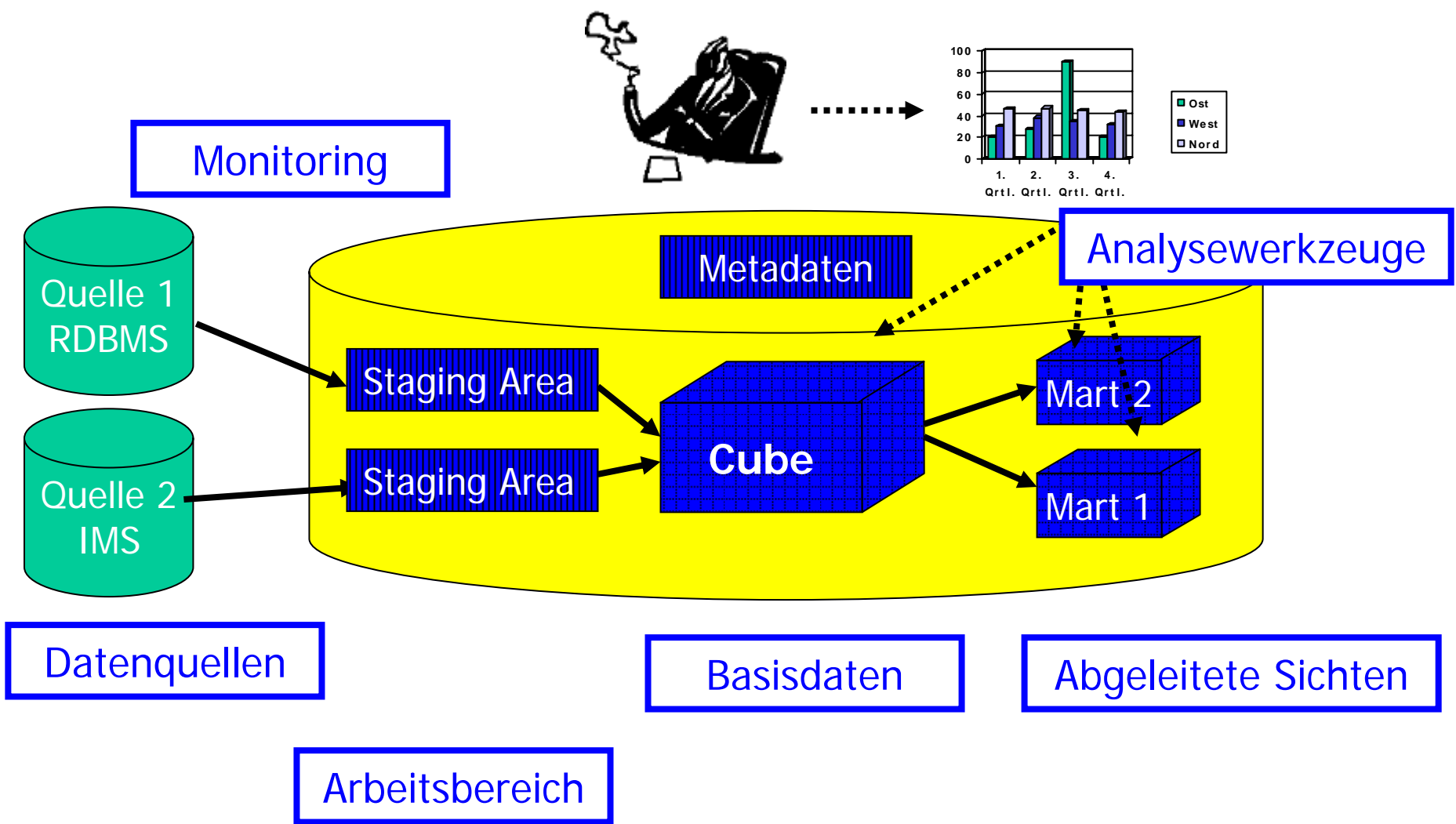
---

- Architektur
- Komponenten
- Prozesse

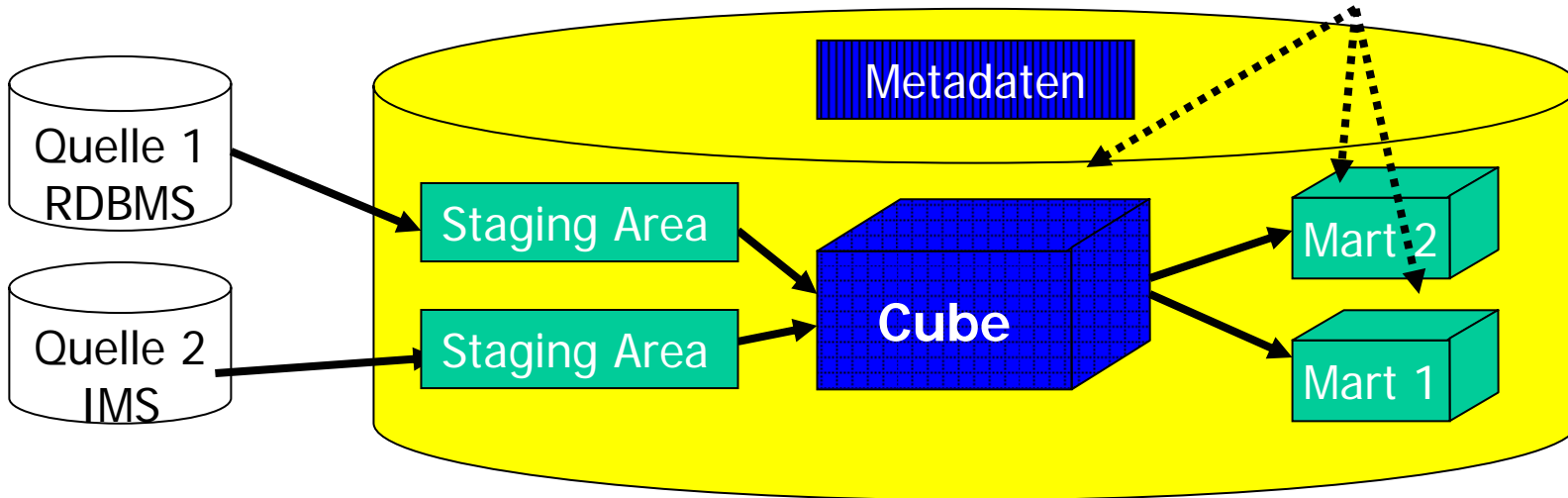
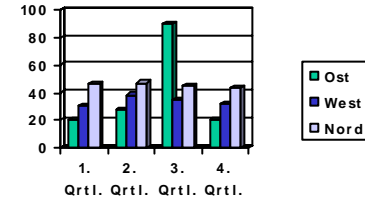
# DWH Grobarchitektur: Hubs and Spokes



# DWH Architektur & Komponenten



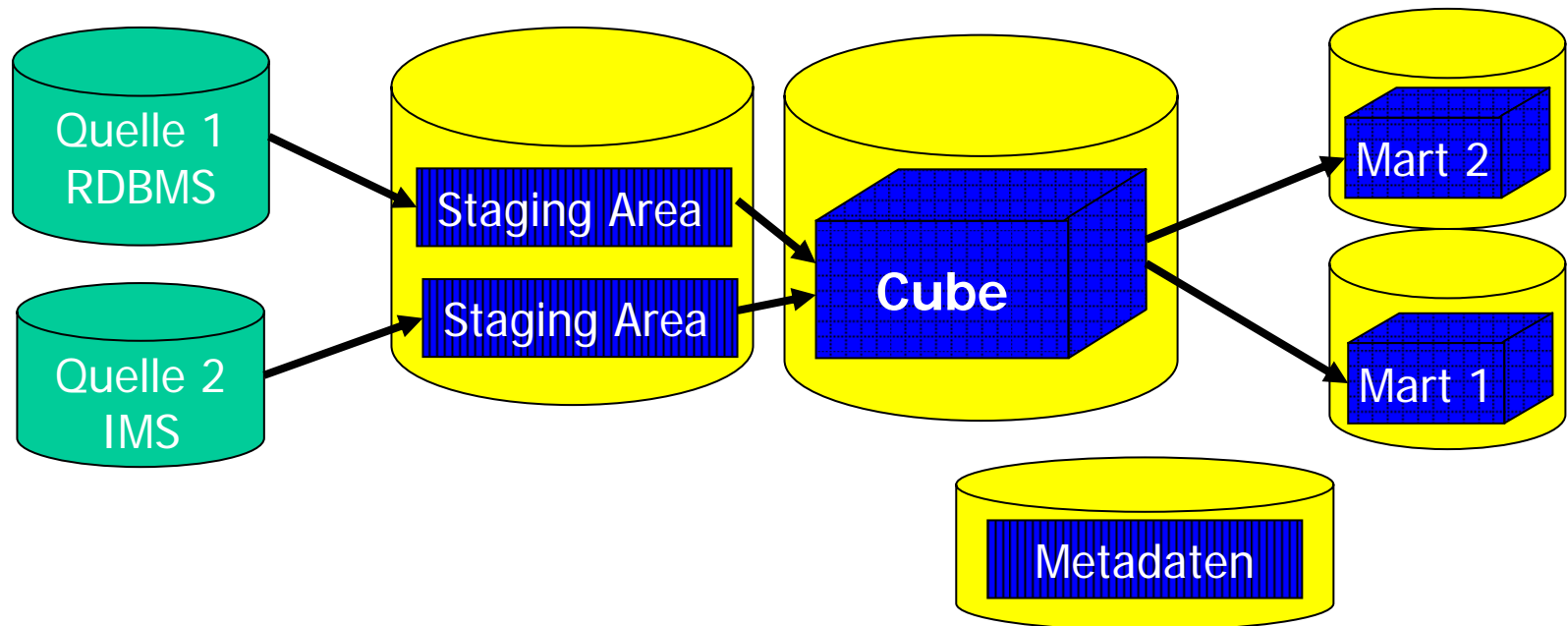
# Langlebigkeit



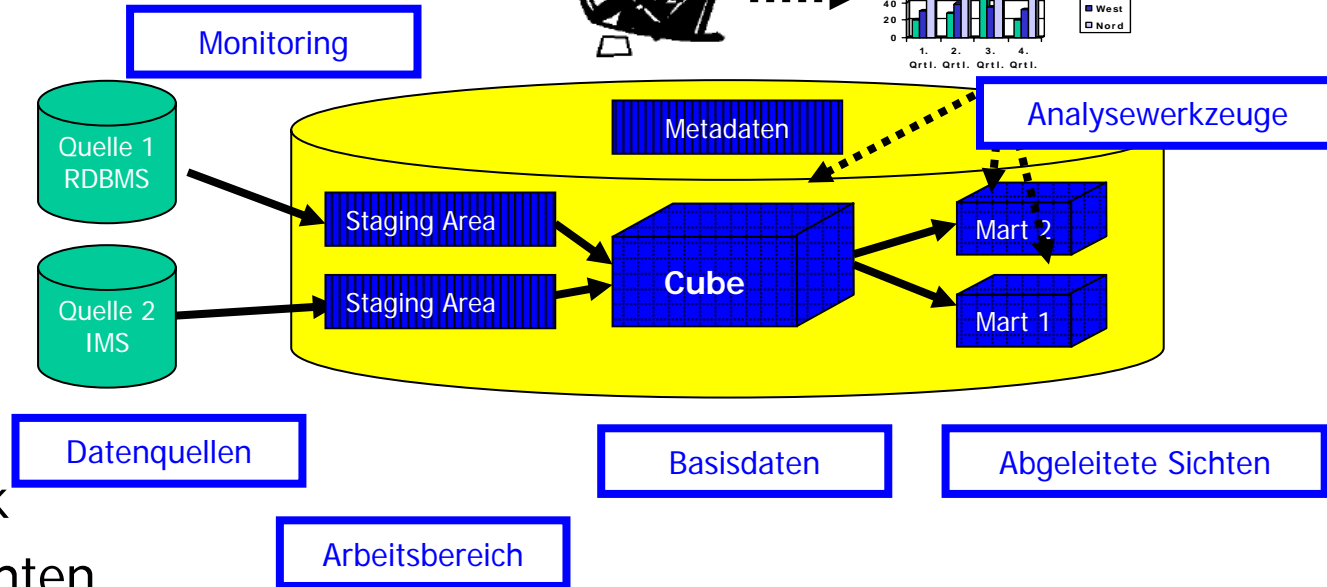
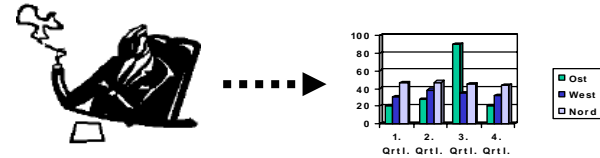
Flüchtig  
Persistent

# Alternativen

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

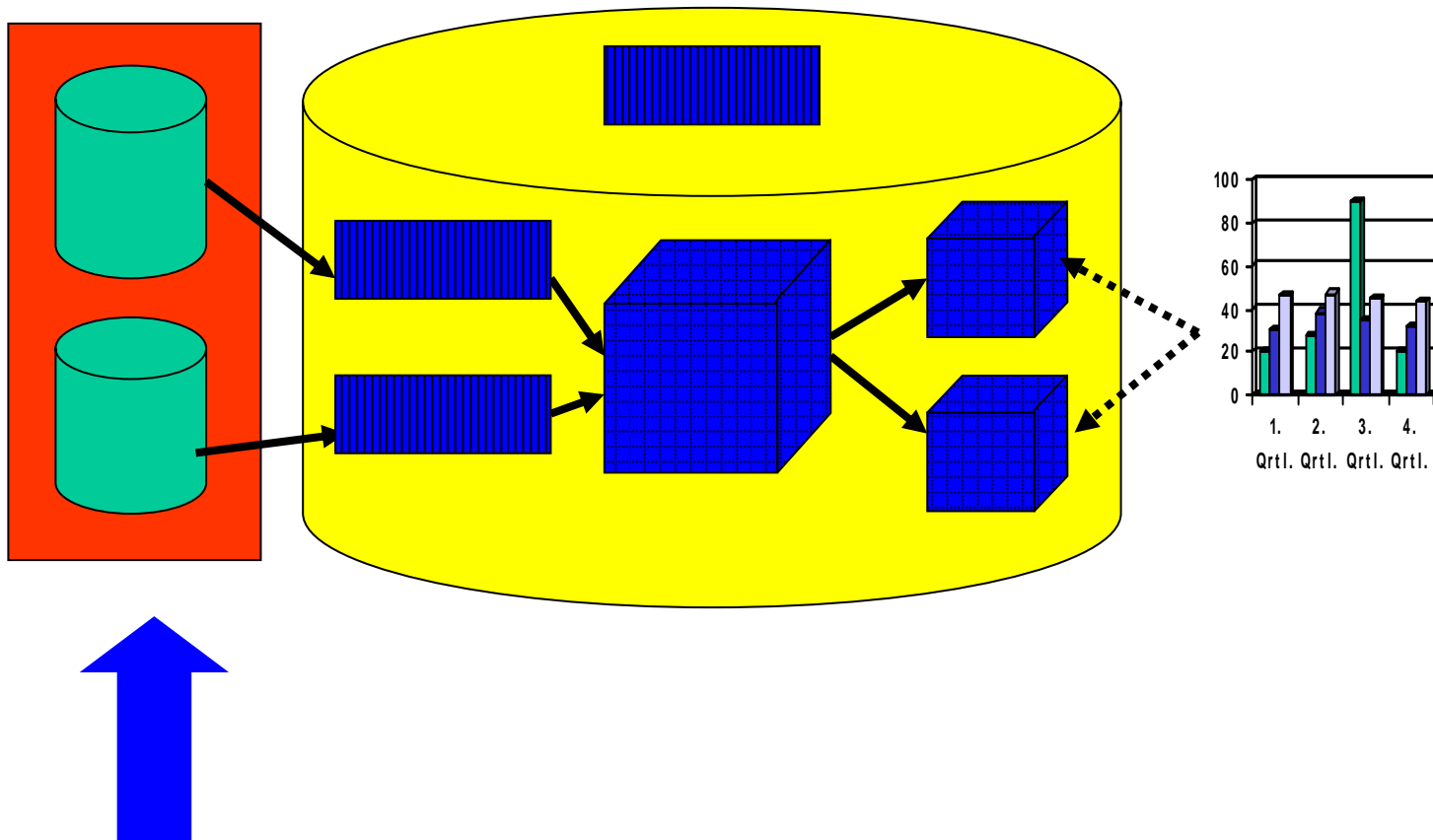


# Inhalt dieser Vorlesung



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

# 1. Datenquellen



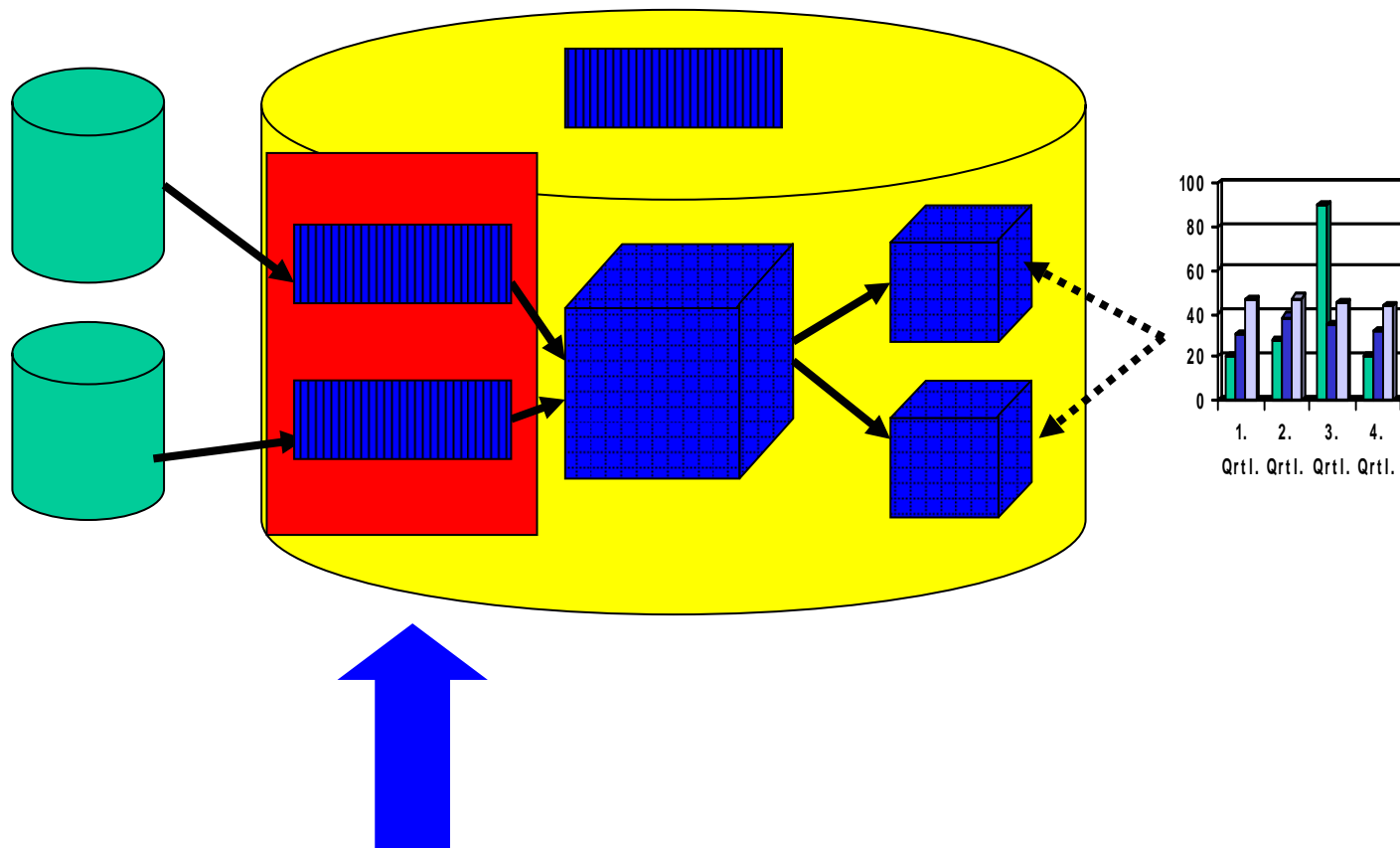
# Datenquellen

---

- 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)

---

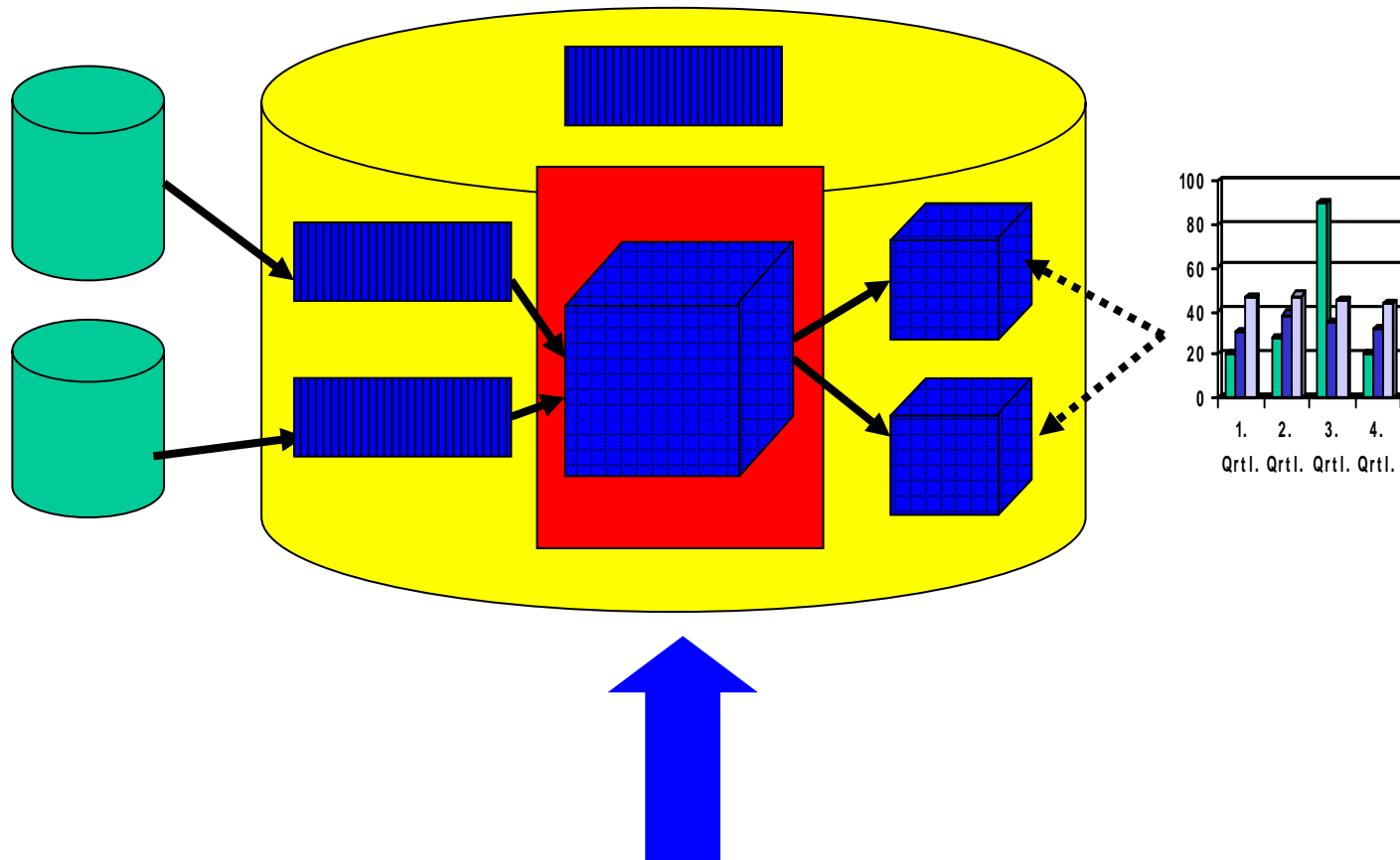


# Arbeitsbereich

---

- 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, ...

SAP R/3, Oracle

Manugistics,  
Commerce-One

Siebel, SAP

Intershop, ...

# 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, ...

# Schwierigkeiten

---

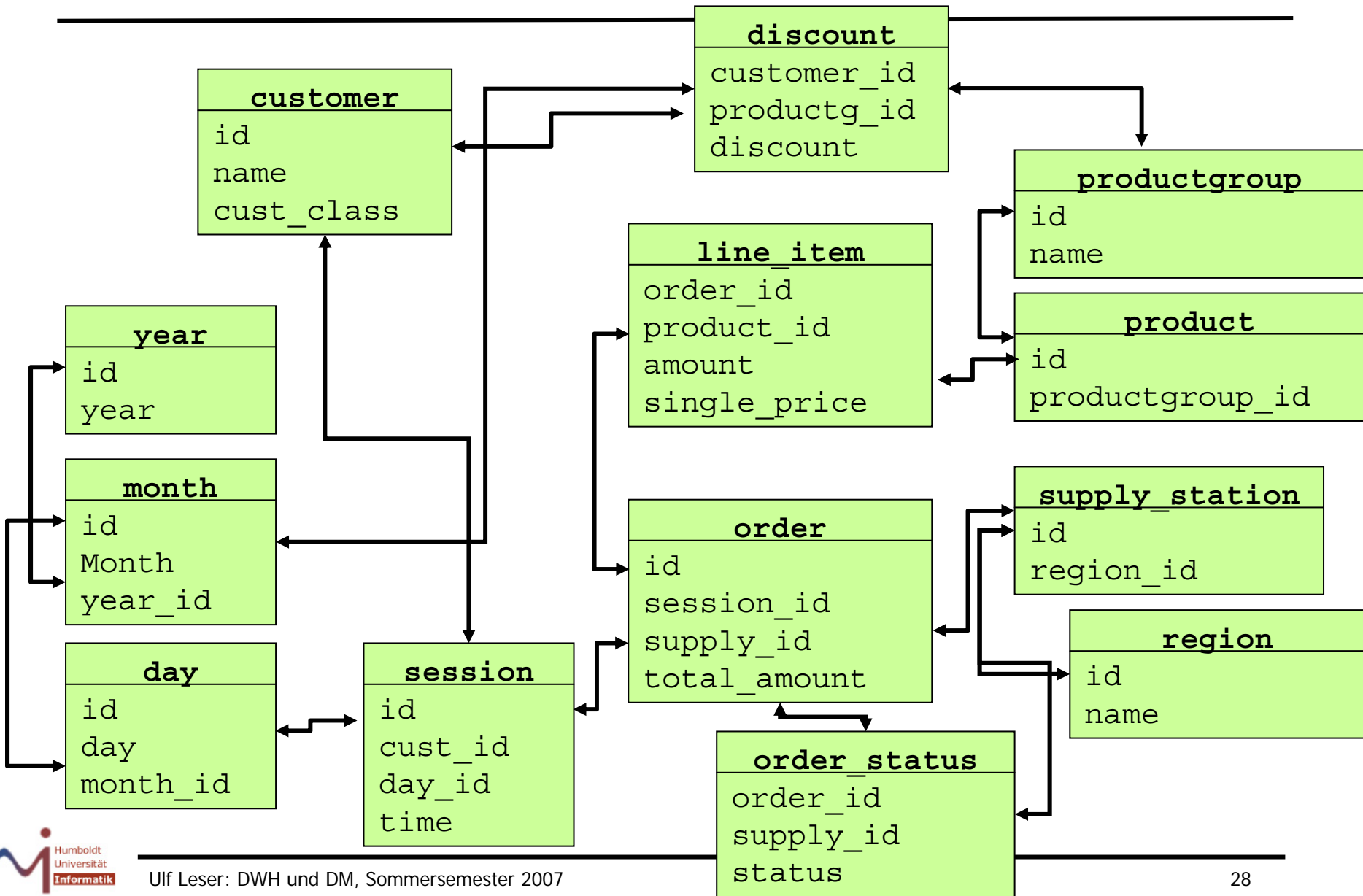
- Hauptproblem: Semantik von Schemaelementen
  - Relationales Datenmodell ist semantikarm
  - Was speichert die Relation **A20RR**?
  - Was speichert das Feld **kunde.name**?
    - Vorname? Nachname? Titel? ...
  - Was ist Umsatz?
    - Brutto? Netto? Nach Abzug Rabatte? Nach Abzug Steuern? ...
- Schemaintegration bisher nicht automatisierbar
- Halbautomatische, interaktive Systeme
- Aktives Forschungsfeld: Schema Matching
  - „Raten“ von Korrespondenzen
  - Instanz-, namens- oder strukturbasiert

# Analyseorientiertes DWH

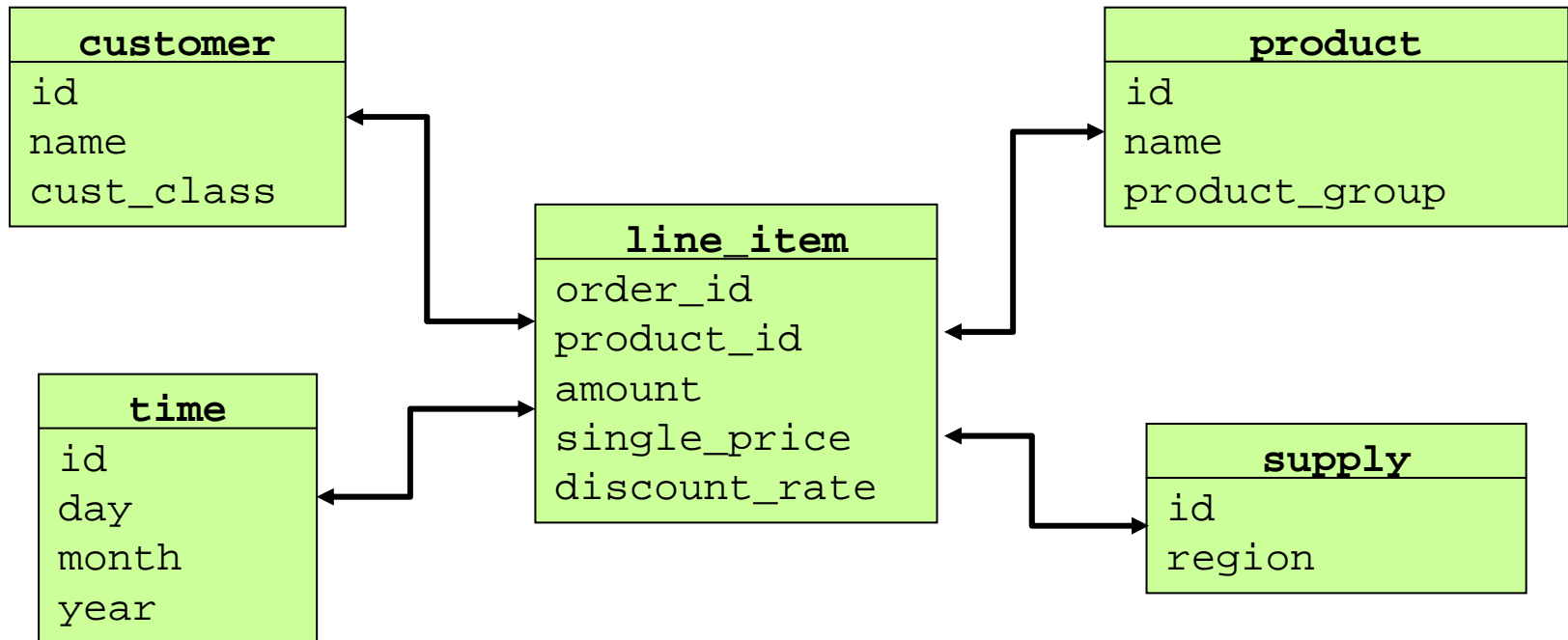
---

- Klassische Datenmodellierung
  - Ziele: Redundanzvermeidung / Integritätswahrung / hoher Durchsatz bei nebenläufigem Zugriff
  - Normalformen, Fremdschlüssel, Satzsperrren
  - 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

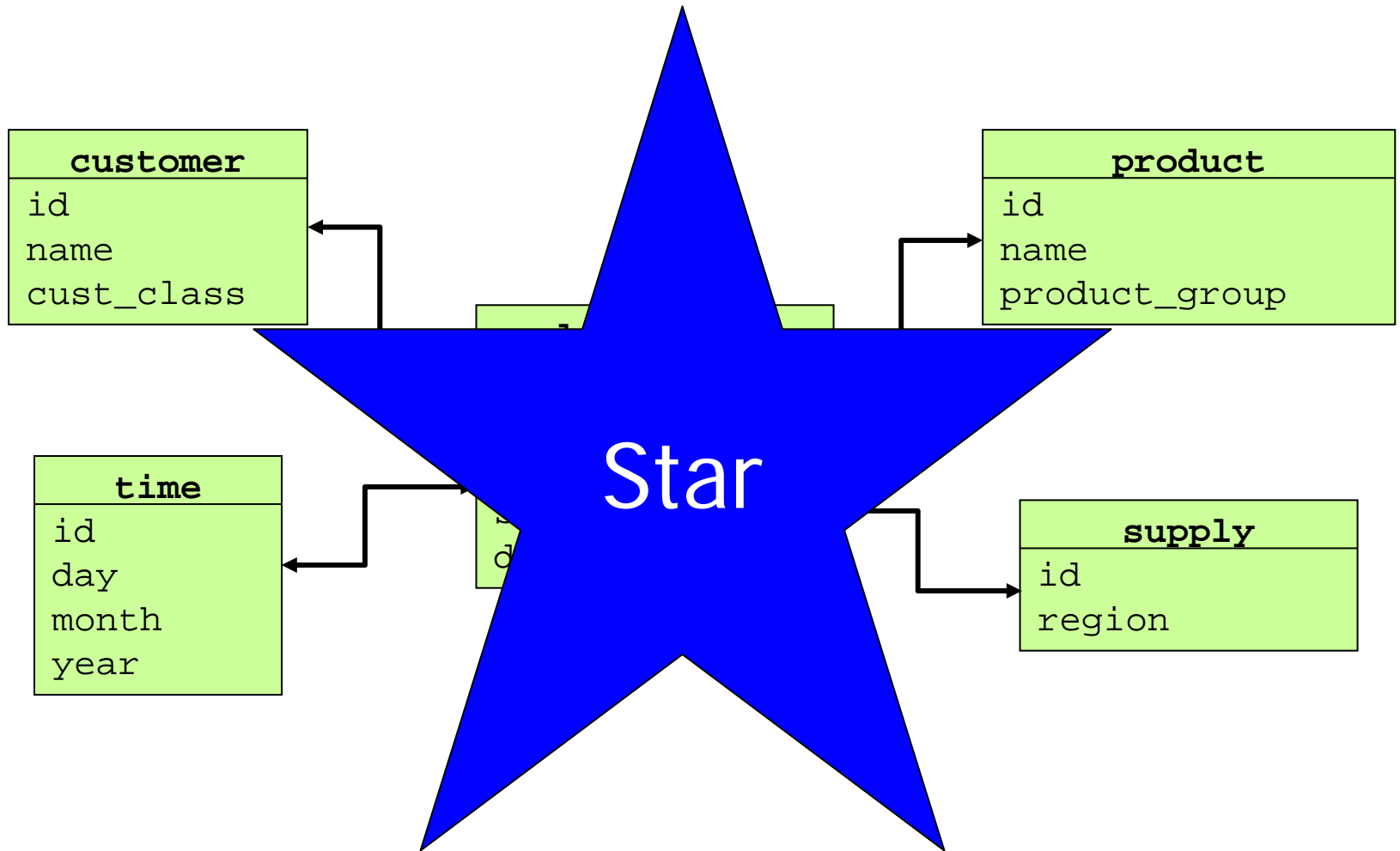


# Multidimensionales Schema

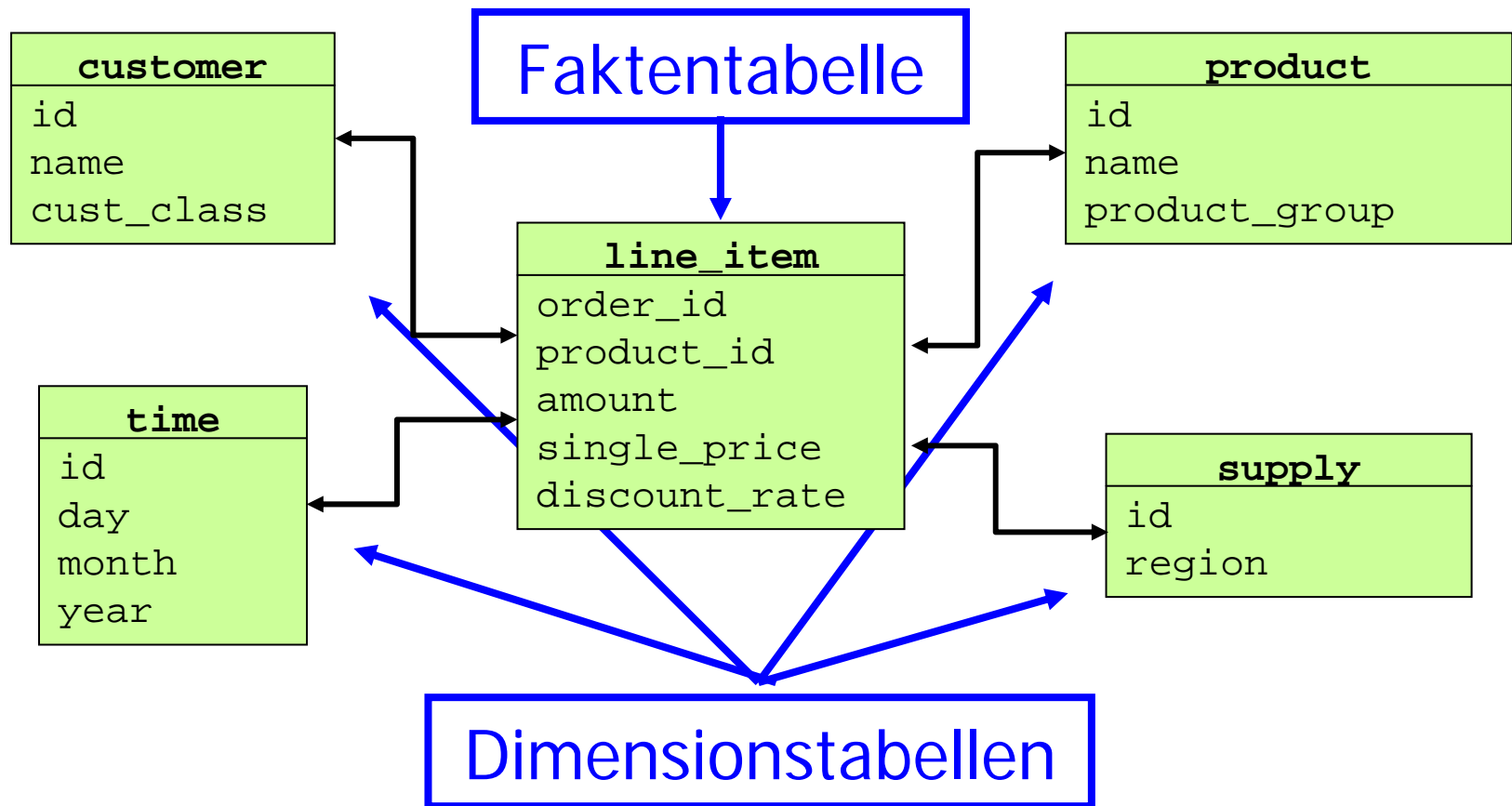


- Technische Informationen raus (Session)
- Nur abgeschlossene Bestellungen aufnehmen (Orderstatus)
- Zusammenfassen (discount\_rate)
- Denormalisieren (überall)
- Konzentration auf **Businessobjekte und -prozesse**

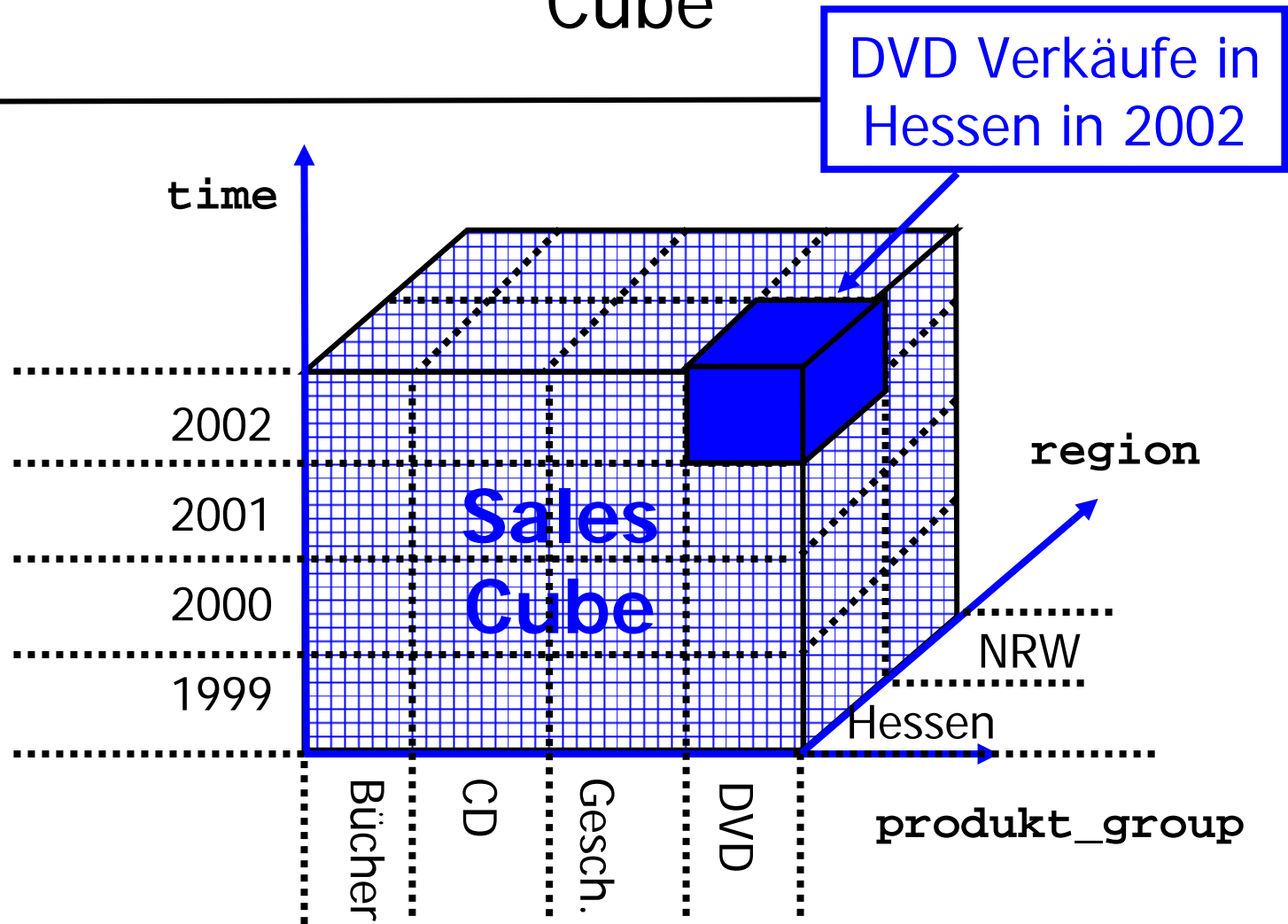
# Starschema



# Dimensionen und Fakten

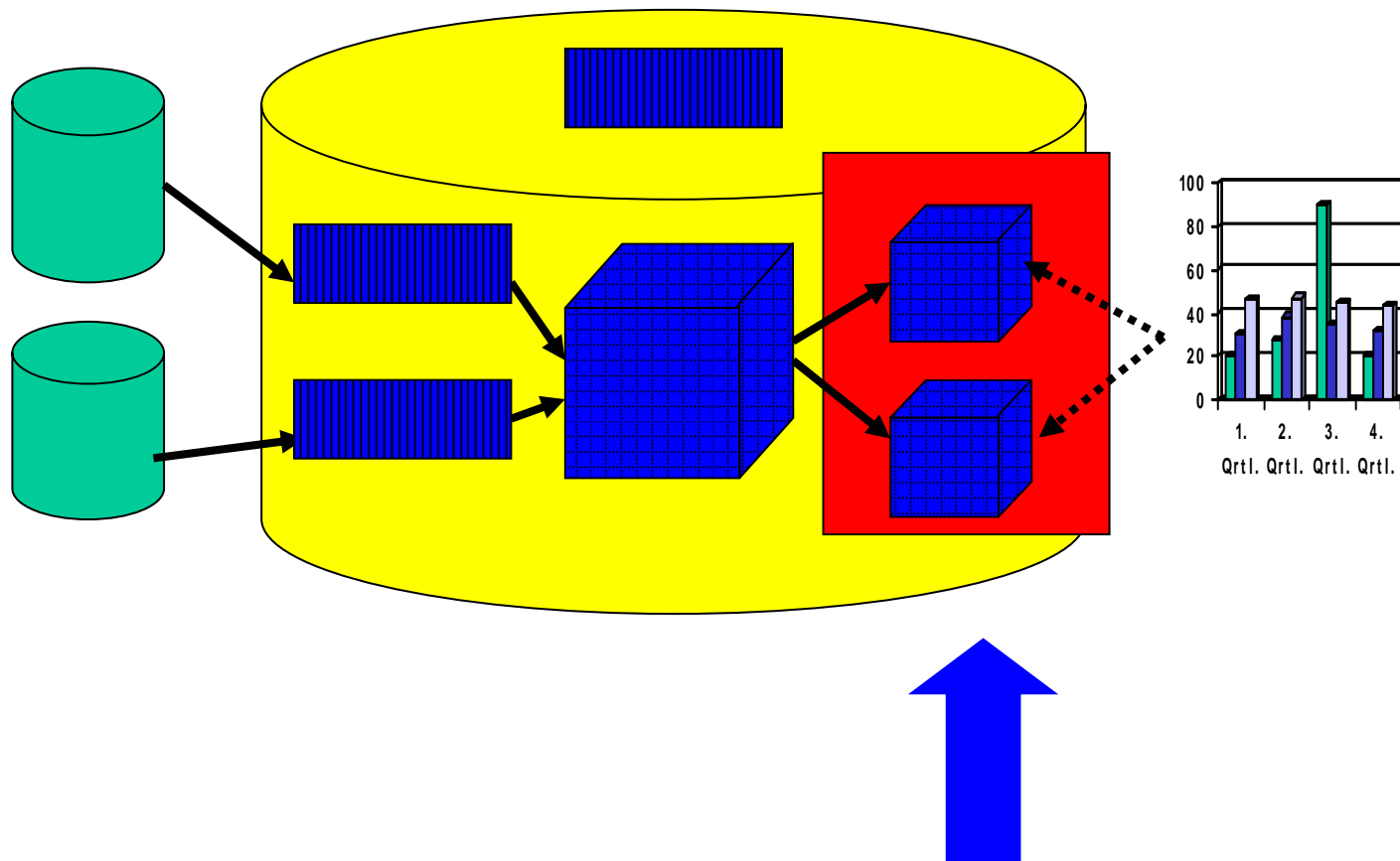


# Cube



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

# 4. Abgeleitete Sichten



# Abgeleitete Sichten

---

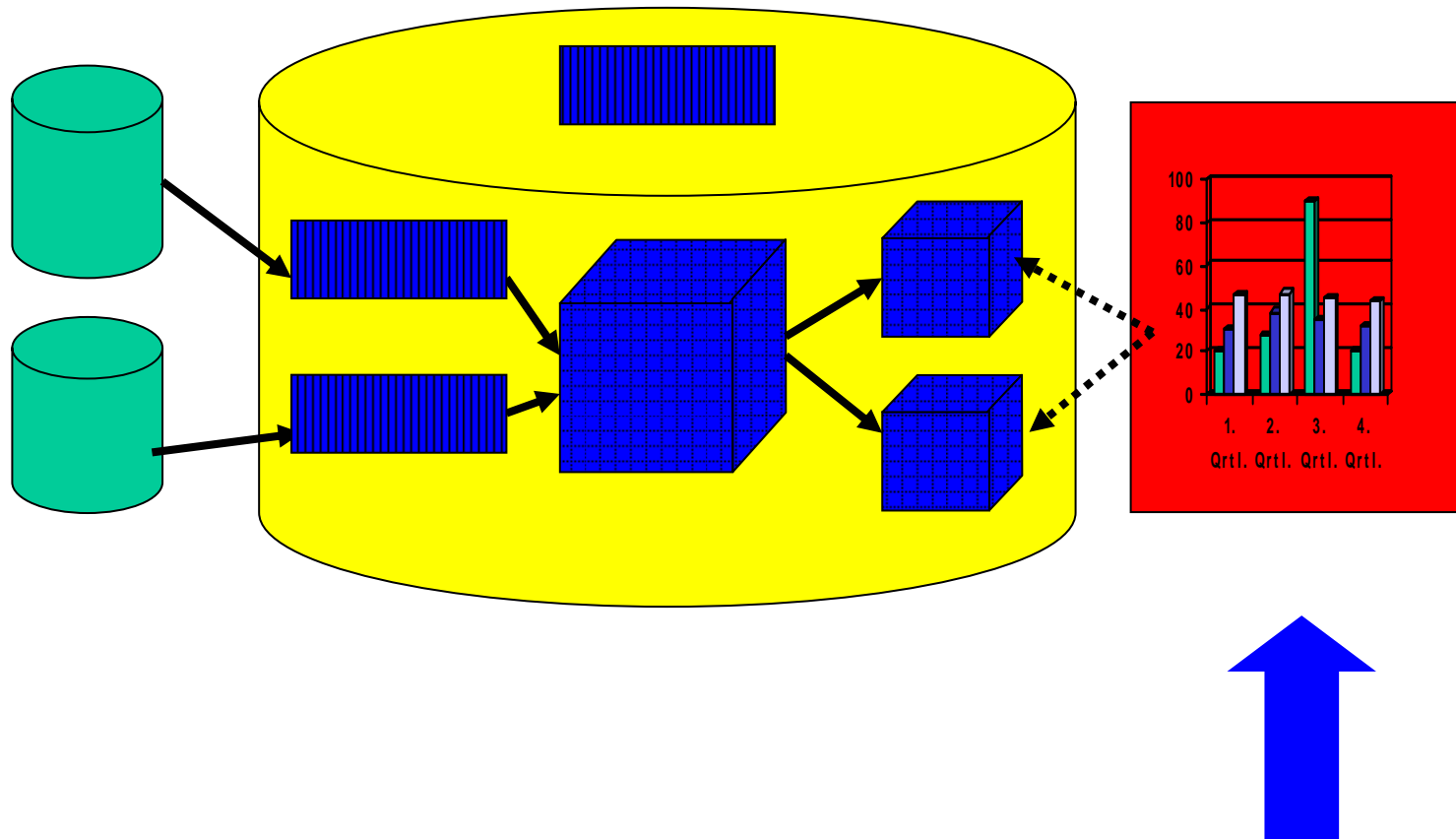
- Analysten benötigt spezielle Daten
  - Aggregiert
    - Alle Verkäufe in Norddeutschland nach Lieferanten
    - Alle Verkäufe nach Niederlassung und Produkten
  - Ausgewählt
    - Alle Verkäufe in Niederlassung X
    - Alle Verkäufe von Lieferant Y
- Probleme bei Auswertung auf Cube
  - Wiederholte Durchforstung sehr großer Datenbestände notwendig
  - Hohe Detailstufe des Cubes für viele Anfragen nicht notwendig
- Vorab-Erstellung von abgeleiteten Daten
  - Data Marts
  - Prä-aggregierte, angereicherte und gefilterte Sichten

# Abgeleitete Sichten –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?
  - Ableitbarkeit von (Teil-)Anfragen aus Sichten
- Auswahl der Sichten
  - Trade-Off Platzverbrauch, Aktualisierungskosten und Anfragebeschleunigung

# Datenanalyse



# 5. Datenanalyse

---

- „Einfache“ statistische Auswertung
  - OLAP Operationen
  - Reports
- OLAP Werkzeuge
  - Häufig proprietäre Systeme, eigene (geheime) Indexstrukturen
  - SAS, SPSS, BusinessObjects, Cognos, Excel, ...
- Funktionalität
  - Grafische Werkzeuge
  - Interaktive Datenauswahl, Filtering, Chaining, ...
  - Navigation, spez. im Cube
  - Präsentation: Grafiken, Tabellen, Reports, ...
- Die allermeisten Analysen sind **Standardreports**
  - Versus Ad-hoc Anfragen

# Data Mining

---

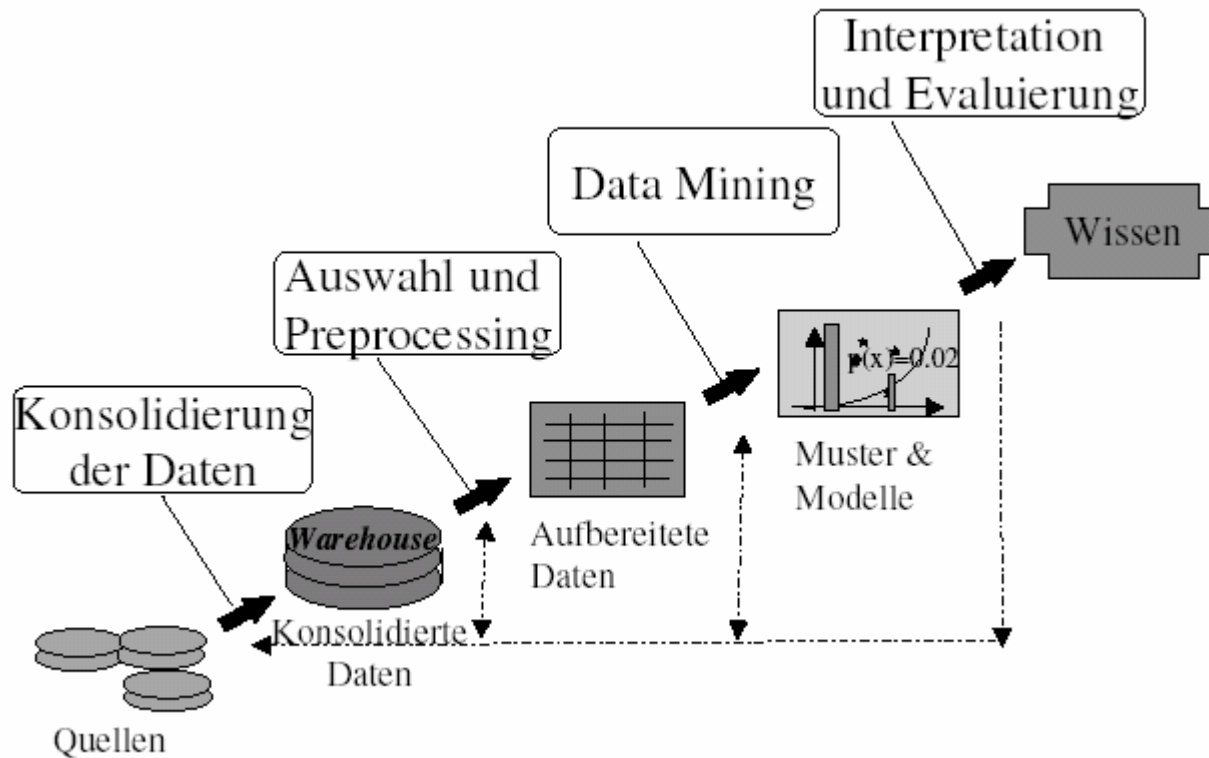
- „Finden verborgender, nicht-trivialer Informationen“
- Bereiche
  - Statistische Analyse
  - Maschinelle Lernverfahren
  - Knowledge Discovery in Databases (KDD)
- Suche nach Auffälligkeiten, Mustern, Regeln
  - Viele Kunden, die Windeln kaufen, kaufen auch Bier
- Suche nach Erklärungsmodellen
  -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

# Deskriptiv oder Präskriptiv

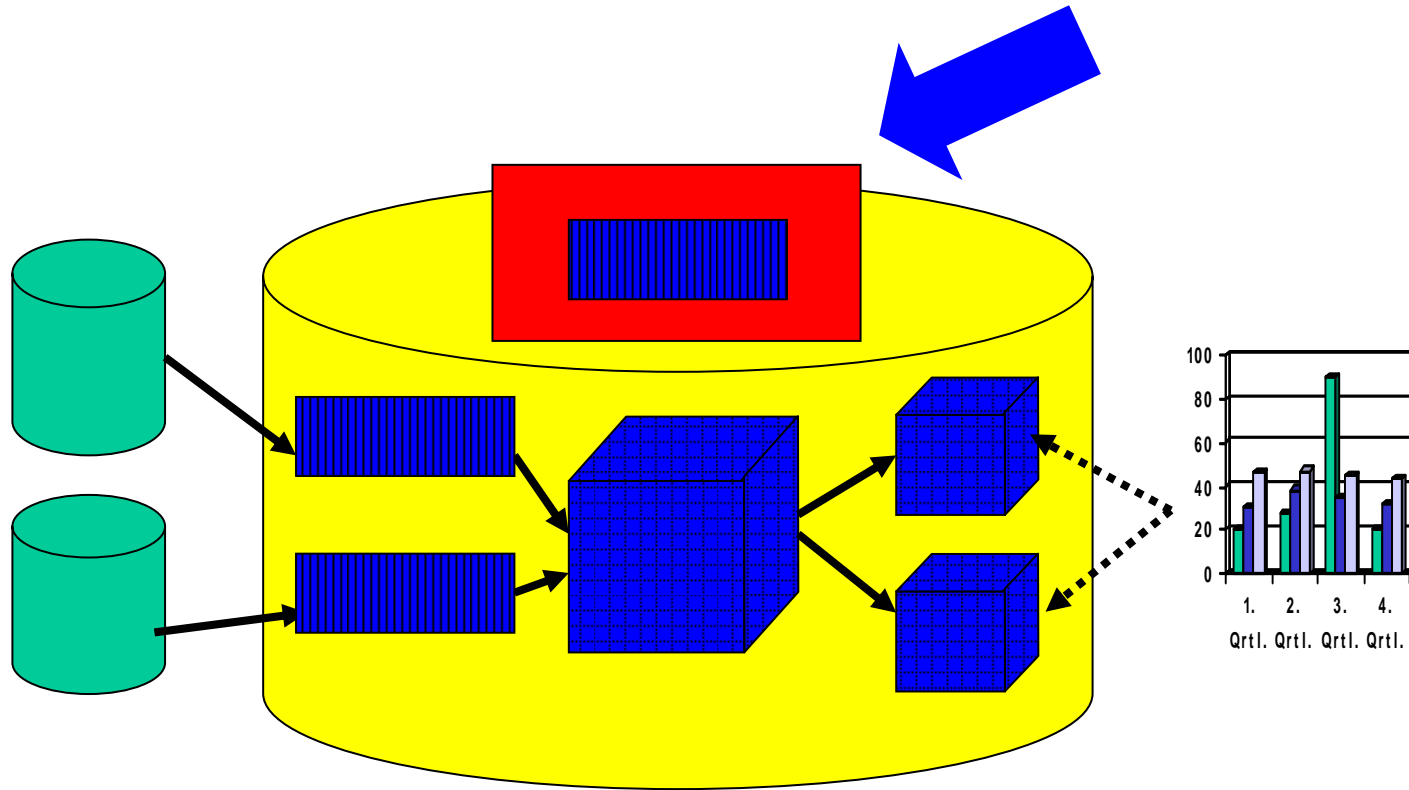
---

- Deskriptive Verfahren
  - Zusammenfassen der wesentlichen Charakteristika von Daten
  - Finden von Ausreißern
  - Finden von Abhängigkeiten
    - Warenkorbanalyse, Assoziationsregeln
  - Finden von Gruppen (Clustering)
    - Kunden mit gleichen Eigenschaften
- Präskriptive Verfahren
  - Ableitung neuen Wissens auf den Daten
  - Basiert auf Modellbildung (durch deskriptive Verfahren)
  - Klassifikation
    - Entscheidungsbäume, Regelsysteme
  - Vorhersage
    - Extrapolation, Regression

# Data Mining Prozess



# 6. Metadatenrepository

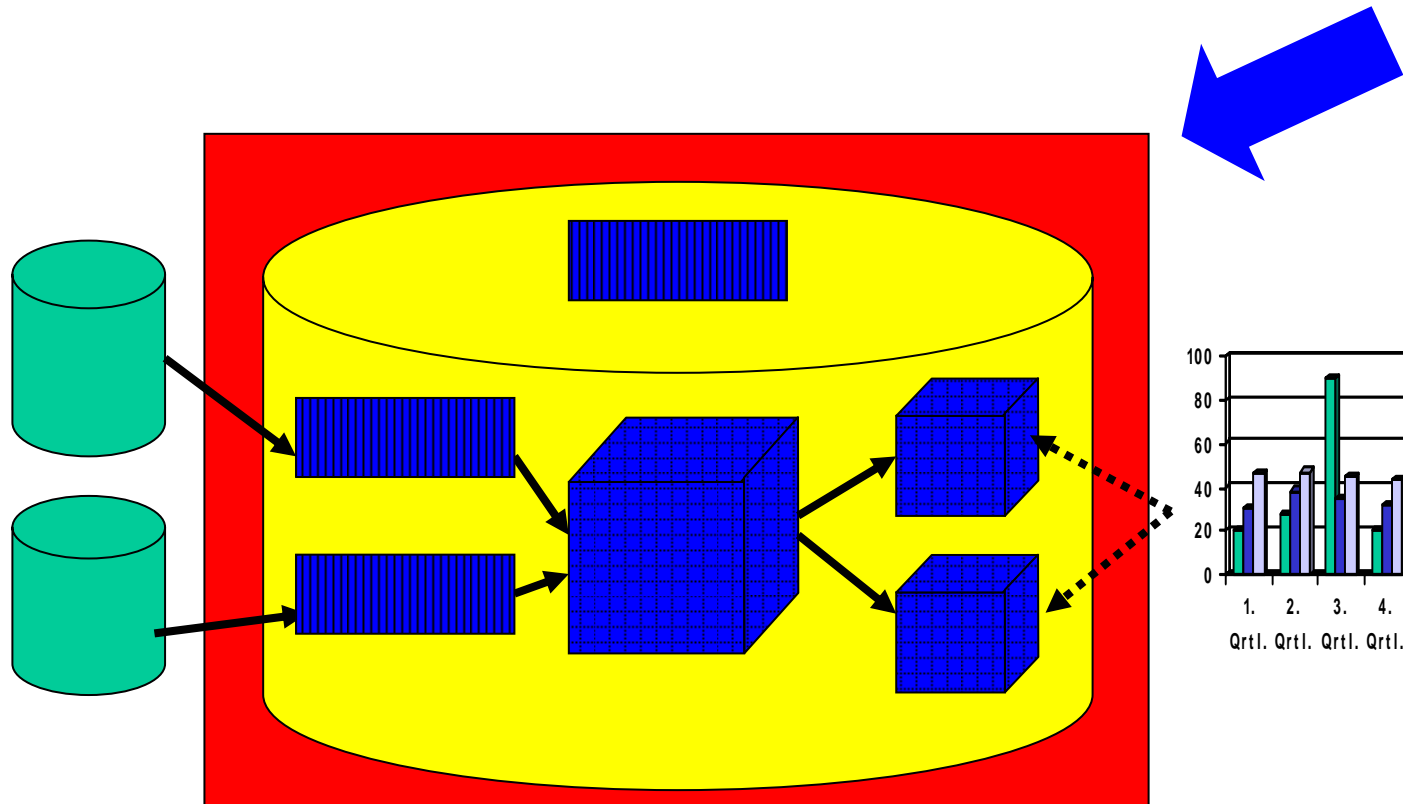


# Metadatenrepository

---

- „... 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
  - Vermeidung von Fehlinterpretationen
  - Technische Beschreibung des DWH
- Produkte: Platinum CA, Microsoft, Oracle, ...
- Standards: IRDS, OIM, CWM, ...

# 7. DWH Manager

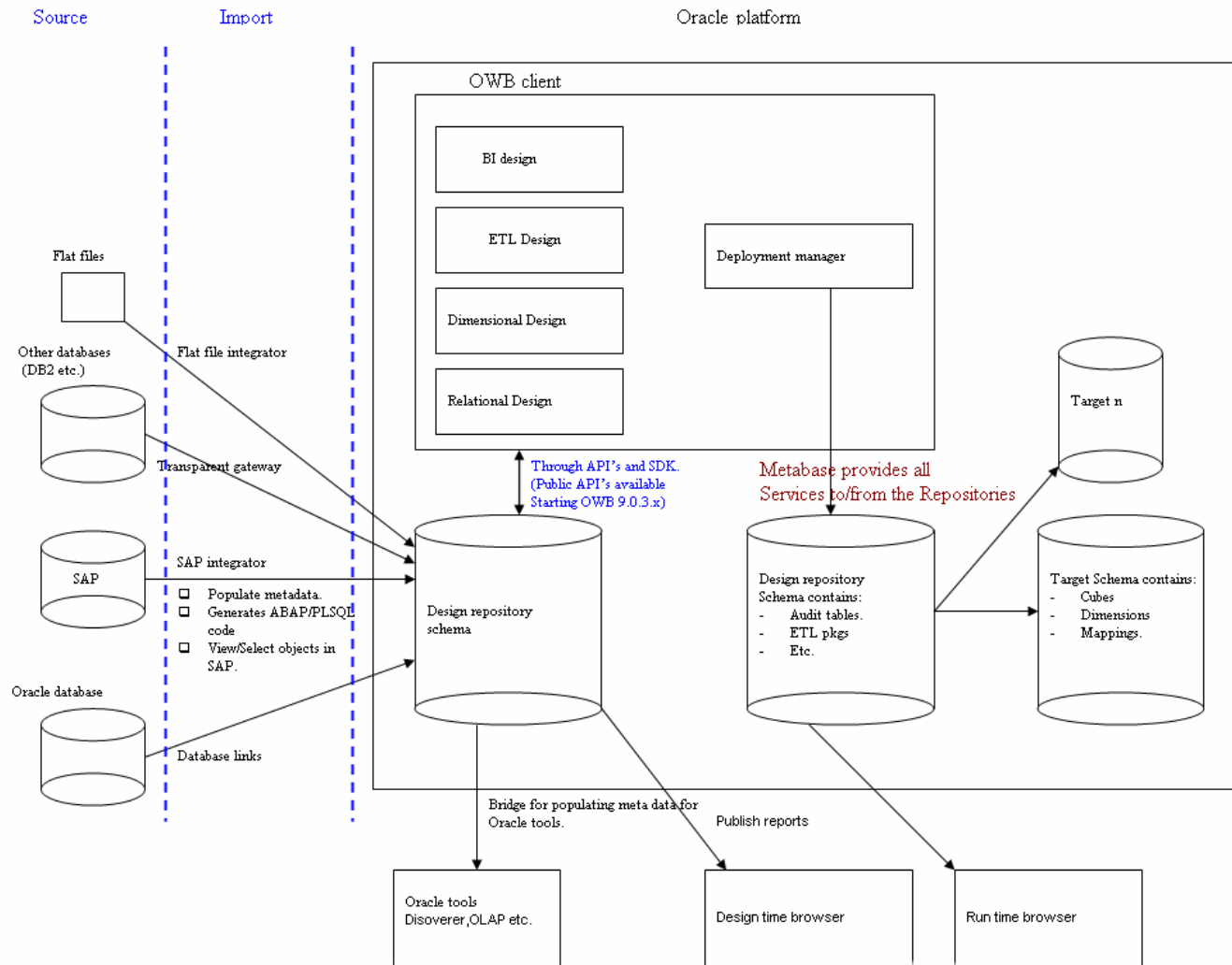


# DWH Manager

---

- Häufig „virtuelle“ Komponente
  - 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

# Oracle Warehouse Builder

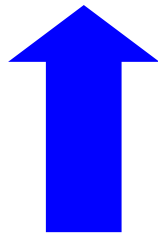
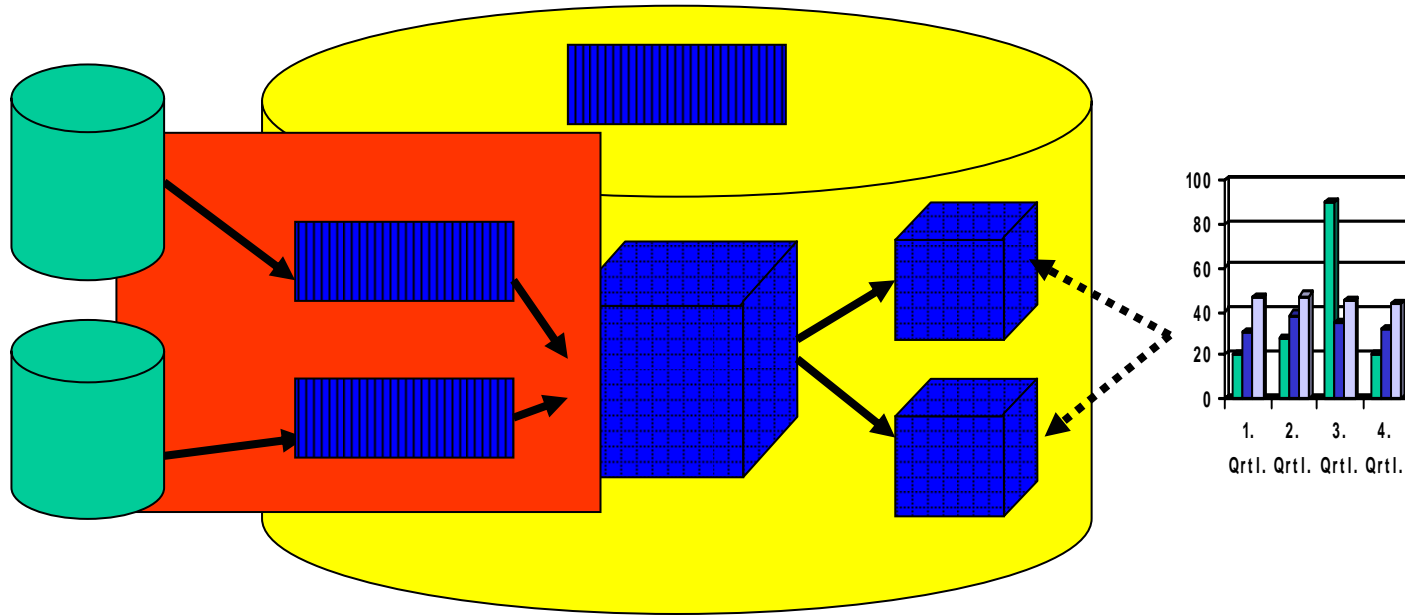


# Inhalt dieser Vorlesung

---

- Architektur
- Komponenten
- Prozesse
  - ETL
  - (Aktualisierung materialisierter Sichten)
  - (Entwicklung, Betrieb, Qualitätskontrolle, ...)

# 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

---

- 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,...

# 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