

---

# Kapitel 6

## Einführung in Data Warehouses

Vorlesung: Dr. Matthias Schubert

Skript © 2009 Matthias Schubert

Dieses Skript basiert auf dem Skript zur Vorlesung Datenbanksysteme II von Prof. Dr. Christian Böhm gehalten im Sommersemester 2007 an der LMU München und dem Skript von Dr. Peer Kröger gehalten im Sommersemester 2008

<http://www.dbs.informatik.uni-muenchen.de/Lehre/DBSII>

---

## 6 Einführung in Data Warehouses

---

### Übersicht

- 6.1 Einleitung
- 6.2 Datenmodellierung
- 6.3 Anfragebearbeitung

## Übersicht

6.1 Einleitung

6.2 Datenmodellierung

6.3 Anfragebearbeitung

266

---

## 6.1 Einleitung

---

### Zwei Arten von DB-Anwendungen

- Online Transaction Processing (OLTP)
  - Routinetransaktionsverarbeitung
  - Realisierung des operationalen Tagesgeschäfts wie
    - “Buchen eines Flugs”
    - “Verarbeitung einer Bestellung”
    - “Ein- und Verkauf von Waren”
    - ...
  - Charakteristik:
    - Arbeitet auf dem jüngsten, aktuellsten Zustand der Daten
    - Änderungstransaktionen (kurze Lese-/Schreibzugriffe)
    - Zugriff auf sehr begrenzte Datenmenge
    - Sehr kurze Antwortzeiten erwünscht (ms-s)
  - OLTP-Datenbanken optimieren typischerweise den logischen und physischen DB-Entwurf hinsichtlich dieser Charakteristik

267

## 6.1 Einleitung

---

### Zwei Arten von DB-Anwendungen (cont.)

- Online Analytical Processing (OLAP)
  - Bilden Grundlage für strategische Unternehmensplanung (Decision Support)
  - Anwendungen wie
    - „Entwicklung der Auslastung der Transatlantik-Flüge über die letzten 2 Jahre?“
    - „Auswirkungen spezieller Marketingaktionen auf Verkaufszahlen der Produkte?“
    - „Voraussichtliche Verkaufszahl eines Produkts im nächsten Monat?“
    - ...
  - Charakteristik:
    - Arbeitet mit „historischen“ Daten (lange Lesetransaktionen)
    - Zugriff auf sehr große Datenmengen
    - Meist Integration, Konsolidierung und Aggregation der Daten
    - Mittlere Antwortzeiten akzeptabel (s-min)
  - OLAP-Datenbanken optimieren typischerweise den logischen und physischen DB-Entwurf hinsichtlich dieser Charakteristik

268

## 6.1 Einleitung

---

### Zwei Arten von DB-Anwendungen (cont.)

- OLTP- und OLAP-Anwendungen sollten nicht auf demselben Datenbestand ausgeführt werden
  - Unterschiedliche Optimierungsziele beim Entwurf
  - Komplexe OLAP-Anfragen könnten die Leistungsfähigkeit der OLTP-Anwendungen beeinträchtigen
- Data Warehouse
  - Datenbanksystem, indem alle Daten für OLAP-Anwendungen in konsolidierter Form gesammelt werden
  - Integration von Daten aus operationalen DBs aber auch aus Dateien (Excel, ...), ...
  - Daten werden dabei oft in aggregierter Form gehalten
  - Enthält historische Daten
  - Regelmäßige Updates (periodisch)

269

## 6.1 Einleitung

---

### Operationales DBS vs. Data Warehouse

	<b>operationales DBS</b>	<b>Data Warehouse</b>
<b>Ziel</b>	Abwicklung des Geschäfts	Analyse des Geschäfts
<b>Focus auf</b>	Detail-Daten	aggregierten Daten
<b>Versionen</b>	nur aktuelle Daten	gesamte Historie der Daten
<b>DB-Größe</b>	~ 1 GB	~ 1 TB
<b>DB-Operationen</b>	Updates und Anfragen	nur Anfragen
<b>Zugriffe pro Op.</b>	~ 10 Datensätze	~ 1.000.000 Datensätze
<b>Leistungsmaß</b>	Durchsatz	Antwortzeit

270

---

## 6.1 Einleitung

### Data Warehouses

- Begriff:

*A Data Warehouse is a subject-oriented, integrated, non-volatile, and time variant collection of data to support management decisions*

[W.H. Inmon, 1996]

271

## 6.1 Einleitung

---

### Data Warehouses (cont.)

- Begriff: *A Data Warehouse is a **subject-oriented**, integrated, non-volatile, and time variant collection of data to support management decisions*  
[W.H. Inmon, 1996]
- Fachorientierung (**subject-oriented**)
  - System dient der Modellierung eines spezifischen Anwendungsziel (meist Entscheidungsfindung in Unternehmen)
  - System enthält nur Daten, die für das Anwendungsziel nötig sind. Für das Anwendungsziel irrelevante Daten werden weggelassen.

272

## 6.1 Einleitung

---

### Data Warehouses (cont.)

- Begriff: *A Data Warehouse is a subject-oriented, **integrated**, non-volatile, and time variant collection of data to support management decisions.*  
[W.H. Inmon, 1996]
- Fachorientierung (subject-oriented)
  - System dient nicht der Erfüllung einer Aufgabe (z.B. Verwlatung von Personaldaten)
  - System dient der Modellierung eines spezifischen Anwendungsziel
- Integrierte Datenbasis (**integrated**)
  - Verarbeitung der Daten aus unterschiedlichen Datenquellen

273

## 6.1 Einleitung

---

### Data Warehouses (cont.)

- Begriff: *A Data Warehouse is a subject-oriented, integrated, **non-volatile**, and time variant collection of data to support management decisions.* [W.H. Inmon, 1996]
- Nicht-flüchtige Datenbasis (**non-volatile**)
  - Stabile, persistente Datenbasis
  - Daten im Data Warehouse werden nicht mehr entfernt oder geändert

274

## 6.1 Einleitung

---

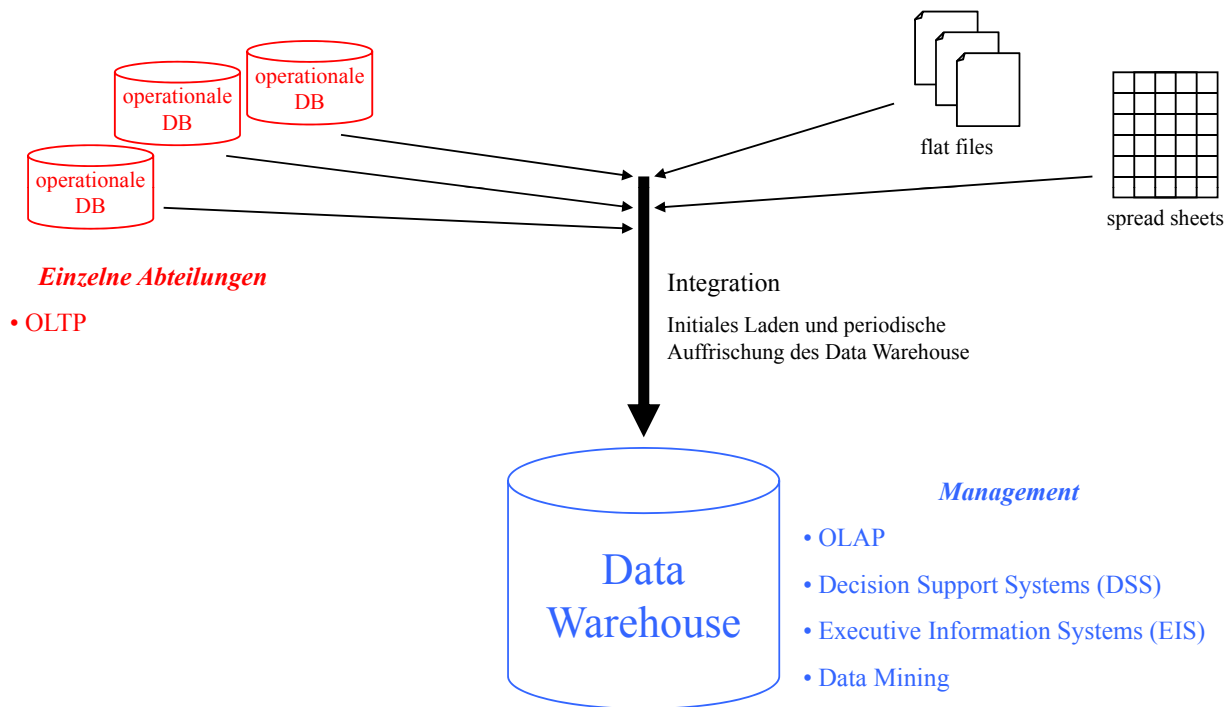
### Data Warehouses (cont.)

- Begriff: *A Data Warehouse is a subject-oriented, integrated, non-volatile, and **time variant** collection of data to support management decisions*  
[W.H. Inmon, 1996]
- Nicht-flüchtige Datenbasis (non-volatile)
  - Stabile, persistente Datenbasis
  - Daten im Data Warehouse werden nicht mehr entfernt oder geändert
- Historische Daten (**time variant**)
  - Vergleich der Daten über die Zeit möglich
  - Speicherung über längeren Zeitraum

275

## 6.1 Einleitung

### Architektur eines Data Warehouse



276

## 6.1 Einleitung

### Data Warehouses und Data Marts

- Manchmal kann es sinnvoll sein, nur eine inhaltlich beschränkte Sicht auf das Data Warehouse bereitzustellen (z.B. für eine Abteilung)
  - => Data Mart
  - Gründe:
    - Eigenständigkeit, Datenschutz, Lastenverteilung, ...
  - Realisierung:
    - Verteilung der DW-Datenbasis
  - Klassen:
    - Abhängige Data Marts: Verteilung eines bestehenden DWs
      - => Analysen auf DM konsistent zu Analysen auf gesamten DW
    - Unabhängige Data Marts: unabhängig voneinander entstandene „kleine“ DWs, nachträgliche Integration zum globalen DW
      - => unterschiedliche Analysesichten

277

# 6 Einführung in Data Warehouses

---

## Übersicht

6.1 Einleitung

6.2 Datenmodellierung

6.3 Anfragebearbeitung

278

---

## 6.2 Datenmodellierung

---

### Motivation

- Datenmodell sollte bzgl. Analyseprozess optimiert werden
- Datenanalyse im Entscheidungsprozess
  - Betriebswirtschaftliche Kennzahlen stehen im Mittelpunkt (z.B. Erlöse, Gewinne, Verluste, Umsätze, ...)  
=> **Fakten**
  - Betrachtung dieser Kennzahlen aus unterschiedlichen Perspektiven (z.B. zeitlich, regional, produktbezogen, ...)  
=> **Dimensionen**
  - Unterteilung der Auswertungsdimensionen möglich (z.B. zeitlich: Jahr, Quartal, Monat; regional: Bundesländer, Bezirke, Städte/Gemeinden; ...)  
=> **Hierarchien, Konsolidierungsebenen**

279



## 6.2 Datenmodellierung

---

### Kennzahlen/Fakten

- Kennzahlen/Fakten
  - Numerische Messgrößen
  - Beschreiben betriebswirtschaftliche Sachverhalte
- Beispiele: Umsatz, Gewinn, Verlust, ...
- Typen
  - Additiv: (additive) Berechnung zwischen sämtlichen Dimensionen möglich (z.B. Bestellmenge eines Artikels)
  - Semi-additiv: (additive) Berechnung möglich mit Ausnahme temporaler Dimension (z.B. Lagerbestand, Einwohnerzahl)
  - Nicht-Additiv: keine additive Berechnung möglich (z.B. Durchschnittswerte, prozentuale Werte, ...)

280

## 6.2 Datenmodellierung

---

### Dimensionen

- Dimension
  - Beschreibt mögliche Sicht auf die assoziierte Kennzahl
  - Endliche Menge von  $d \geq 2$  Dimensionselementen (Hierarchieobjekten), die eine semantische Beziehung aufweisen
  - Dient der orthogonalen Strukturierung des Datenraums
- Beispiele: Produkt, Geographie, Zeit

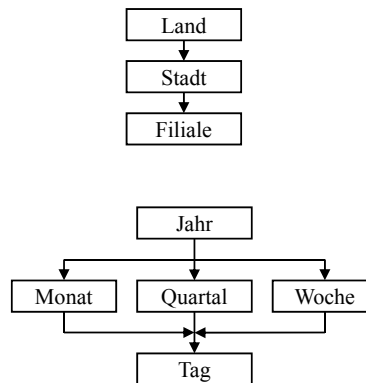
281

## 6.2 Datenmodellierung

---

### Hierarchien in Dimensionen

- Dimensionselemente sind Knoten einer Klassifikationshierarchie
- Klassifikationsstufe beschreibt Verdichtungsgrad
- Darstellung von Hierarchien in Dimensionen über Klassifikationsschema
- Formen
  - Einfache Hierarchien: höhere Ebene
    - enthält die aggregierten Werte genau
    - einer niedrigeren Hierarchiestufe
  - Parallele Hierarchien: innerhalb einer
    - Dimension sind mehrere verschiedene
    - Arten der Gruppierung möglich



282

## 6.2 Datenmodellierung

---

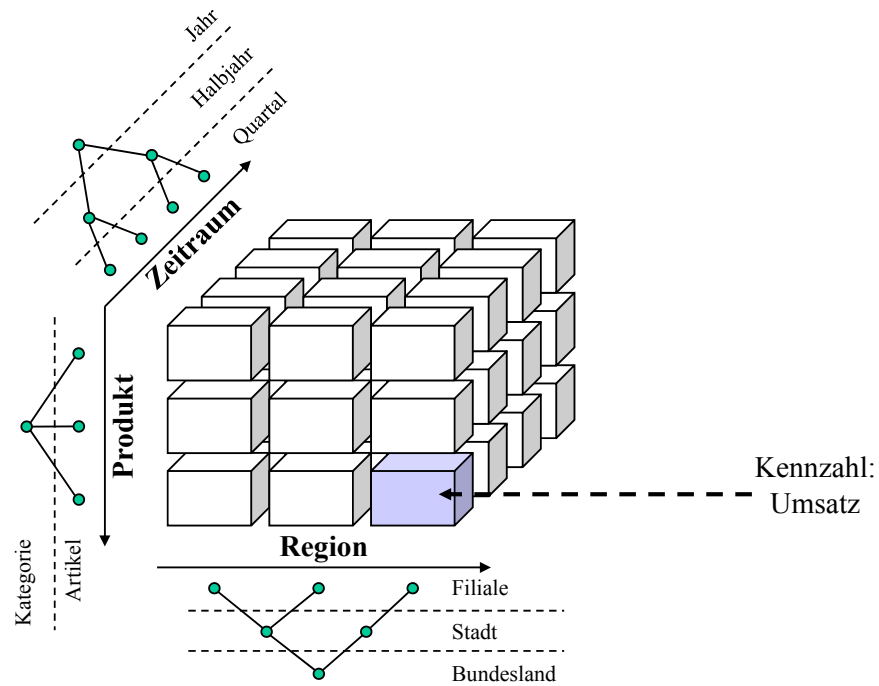
### Data-Cubes

- Grundlage der multidimensionalen Datenanalyse: Datenwürfel (**Data-Cube**)
- Kanten des Cubes: Dimensionen
- Zellen des Cubes: ein oder mehrere Kennzahlen (als Funktion der Dimension)
- Anzahl der Dimensionen: Dimensionalität des Cubes
- Visualisierung
  - 2 Dimensionen: Tabelle
  - 3 Dimensionen: Würfel
  - >3 Dimensionen: Multidimensionale Domänenstruktur

283

## 6.2 Datenmodellierung

Beispiel: 3D Data-Cube

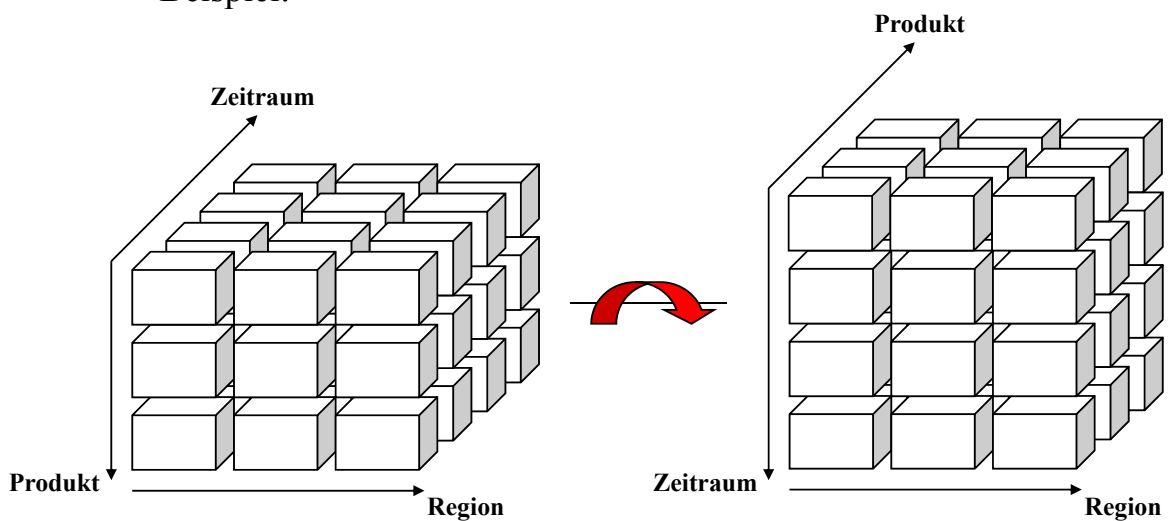


284

## 6.2 Datenmodellierung

Standardoperationen zur Datenanalyse

- Pivottisierung/Rotation
  - Drehen des Data-Cube durch Vertauschen der Dimensionen
  - Datenanalyse aus verschiedenen Perspektiven
  - Beispiel:



285

## 6.2 Datenmodellierung

### Standardoperationen zur Datenanalyse (cont.)

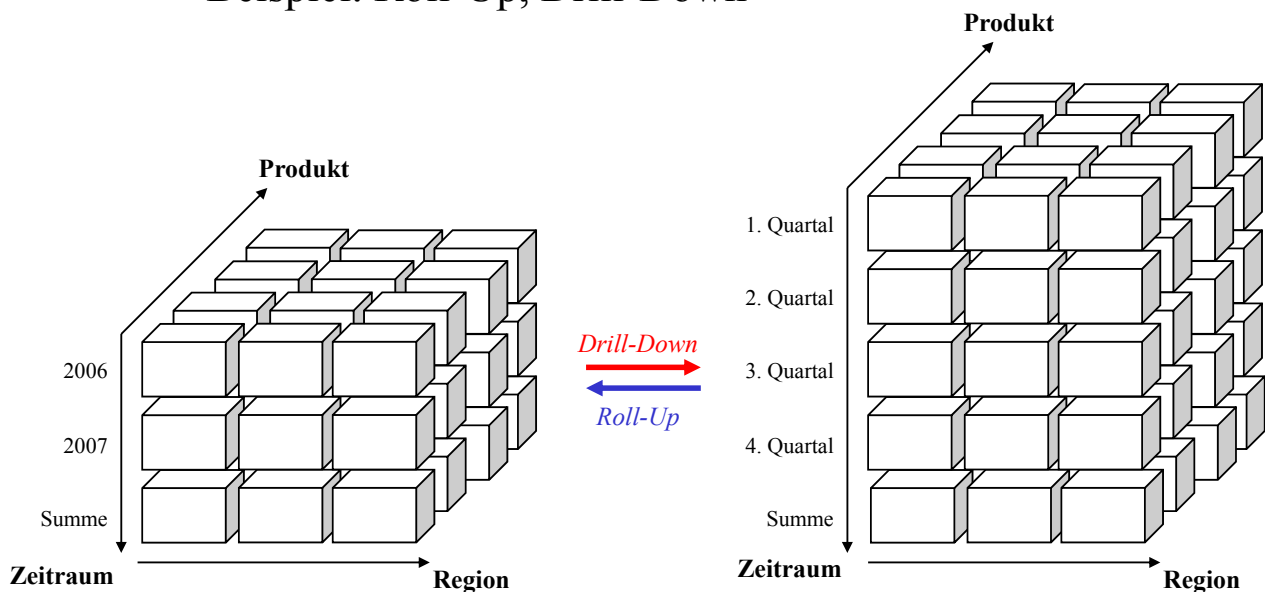
- Roll-Up
  - Erzeugen neuer Informationen durch Aggregation der Daten entlang der Klassifikationshierarchie in einer Dimension  
(z.B. Tag => Monat => Quartal => Jahr)
  - Dimensionalität bleibt erhalten
- Drill-Down
  - Komplementär zu Roll-Up
  - Navigation von aggregierten Daten zu Detail-Daten entlang der Klassifikationshierarchie
- Drill-Across
  - Wechsel von einem Cube zu einem anderen

286

## 6.2 Datenmodellierung

### Standardoperationen zur Datenanalyse (cont.)

- Beispiel: Roll-Up, Drill-Down



287

## 6.2 Datenmodellierung

### Standardoperationen zur Datenanalyse (cont.)

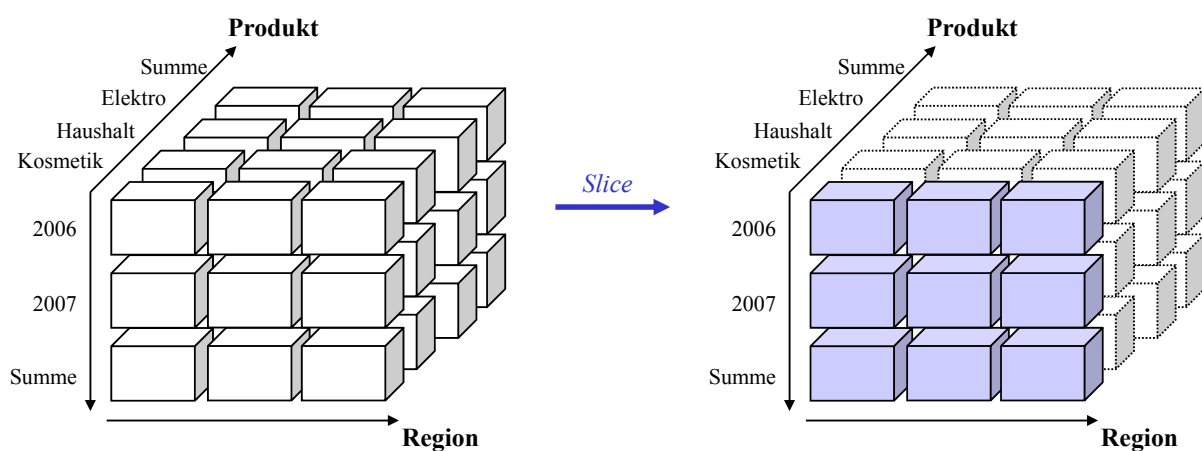
- Slice und Dice
  - Erzeugen individueller Sichten
    - Slice:
      - Herausschneiden von „Scheiben“ aus dem Cube (z.B. alle Werte eines Quartals)
      - Verringerung der Dimensionalität
    - Dice:
      - Herausschneiden eines „Teil-Cubes“ (z.B. Werte bestimmter Produkte und Regionen)
      - Erhaltung der Dimensionalität
      - Veränderung der Hierarchieobjekte

288

## 6.2 Datenmodellierung

### Standardoperationen zur Datenanalyse (cont.)

- Beispiel: Slice



289

## 6.2 Datenmodellierung

---

### Umsetzung des multidimensionalen Modells

- Interne Verwaltung der Daten durch
  - Relationale Strukturen (Tabellen)
    - Relationales OLAP (ROLAP)
    - Vorteile: Verfügbarkeit, Reife der Systeme
  - Multidimensionale Strukturen (direkte Speicherung)
    - Multidimensionales OLAP (MOLAP)
    - Vorteil: Wegfall der Transformation
- Wichtige Designaspekte
  - Speicherung
  - Anfragebearbeitung

290

## 6.2 Datenmodellierung

---

### Relationale Umsetzung: Anforderungen

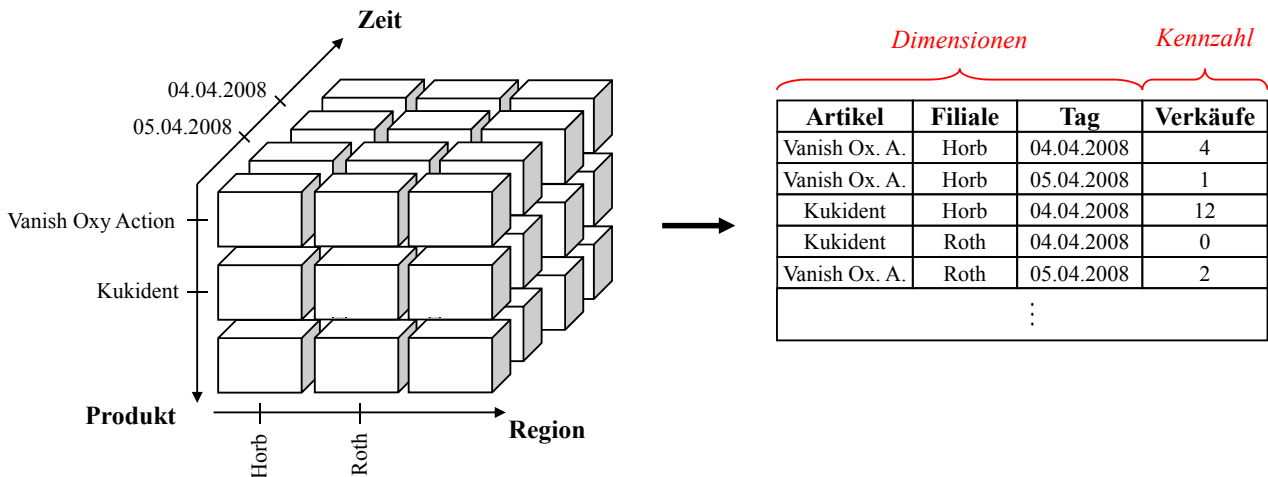
- Vermeidung des Verlusts anwendungsbezogener Semantik aus dem multidimensionalen Modell (z.B. Klassifikationshierarchien)
- Effiziente Übersetzung multidimensionaler Anfragen
- Effiziente Verarbeitung der übersetzten Anfragen
- Einfache Pflege der entstandenen Relationen (z.B. Laden neuer Daten)
- Berücksichtigung der Anfragecharakteristik und des Datenvolumens von Analyseanwendungen

291

## 6.2 Datenmodellierung

### Relationale Umsetzung: Faktentabelle

- Ausgangspunkt: Umsetzung des Data-Cubes ohne Klassifikationshierarchien
  - Dimensionen und Kennzahlen => Attribute der Relation
  - Zellen => Tupel der Relation



292

## 6.2 Datenmodellierung

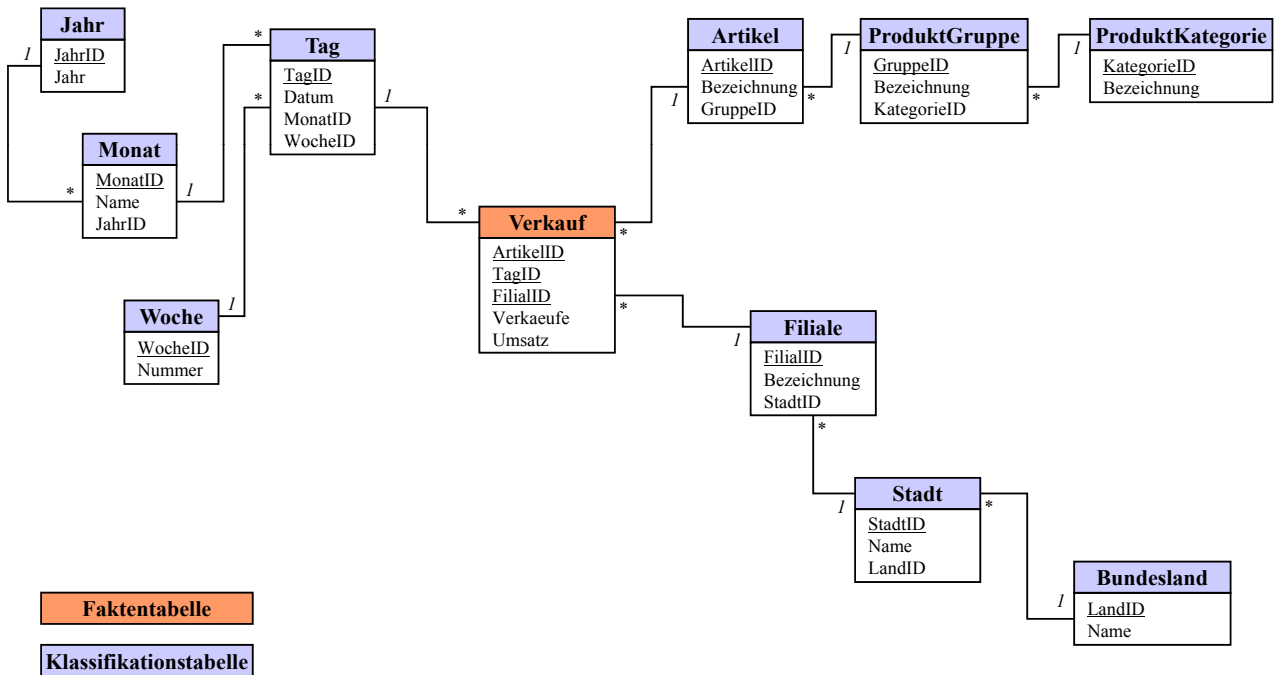
### Relationale Umsetzung: Snowflake Schema

- Abbildung von Klassifikationen?
- Eigene Tabelle für jede Klassifikationsstufe (Artikel, Produktgruppe, ...)
- Klassifikationstabellen enthalten
  - ID für entsprechenden Klassifikationsknoten
  - Beschreibende Attribute (Marke, Hersteller, Bezeichnung, ...)
  - Fremdschlüssel der direkt übergeordneten Klassifikationsstufe
- Faktentabelle enthält
  - Kenngrößen
  - Fremdschlüssel der jeweils niedrigsten Klassifikationsstufe der einzelnen Dimensionen
  - Fremdschlüssel bilden zusammengesetzte Primärschlüssel der Faktentabelle

293

## 6.2 Datenmodellierung

### Snowflake Schema: Beispiel



294

## 6.2 Datenmodellierung

### Relationale Umsetzung: Star Schema

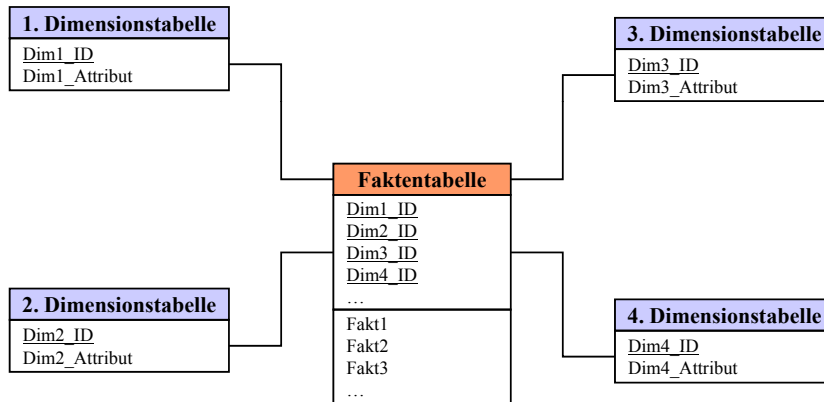
- Snowflake Schema ist normalisiert
  - Keine Update-Anomalien
  - ABER: Zusammenholen von Informationen erfordert Join über mehrere Tabellen
- Star Schema
  - Denormalisierung der zu einer Dimension gehörenden Tabellen
  - Für jede Dimension genau eine **Dimensionstabelle**
  - Redundanzen in der Dimensionstabelle für schnellere Anfragebearbeitung

295



## 6.2 Datenmodellierung

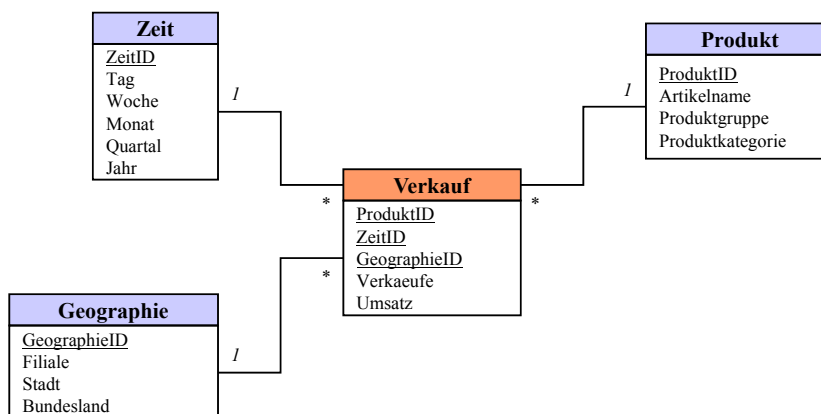
### Star Schema: Visualisierung



296

## 6.2 Datenmodellierung

### Star Schema: Beispiel



297

## 6.2 Datenmodellierung

---

### Relationale Umsetzung: Mischformen

- Idee: Abbildung einzelner Dimensionen anhand von Snowflake oder Star Schema

#### - Kriterien

- Änderungshäufigkeit der Dimensionen  
Reduzierung des Pflegeaufwands => Snowflake
- Anzahl der Klassifikationsstufen einer Dimension  
Höhere Effizienz durch größere Redundanz => Star
- ...

298

## 6.2 Datenmodellierung

---

### **Relationale Umsetzung: Begriff**

- Galaxie (Multi-Cube, Hyper-Cube)
  - Mehrere Faktentabellen im Star Schema teilweise mit gleichen Dimensionstabellen verknüpft
- Fact Constellation
  - Speicherung vorberechneter Aggregate in Faktentabelle (z.B. Umsatz für Region)

299

## 6.2 Datenmodellierung

---

### Relationale Umsetzung: Probleme

- Transformation multidimensionaler Anfragen in relationale Repräsentation nötig
- Einsatz komplexer Anfragewerkzeuge nötig (OLAP-Werkzeuge)
- Semantikverlust
  - Unterscheidung zwischen Kennzahlen und Dimensionen in der Faktentabelle nicht gegeben
  - Unterscheidung zwischen beschreibenden Attributen und Attributen zum Hierarchie-Aufbau in Dimensionstabellen nicht gegeben
- Daher:  
direkte multidimensionale Speicherung besser ???

300

## 6.2 Datenmodellierung

---

### Multidimensionale Umsetzung

- Idee:
  - Verwende entsprechende Datenstrukturen für Data-Cube und Dimensionen
  - Speicherung des Data-Cube als Array
  - Ordnung der Dimensionen nötig, damit Zellen des Data-Cube adressiert werden können
- Bemerkung
  - Häufig proprietäre Strukturen (und Systeme)

301

## 6.2 Datenmodellierung

---

### Multidimensionale Umsetzung (cont.)

- Datenstruktur für eine Dimension
  - Endliche geordnete Liste von Dimensionswerten (aller Klassifikationsstufen)
  - Dimensionswerte: einfache, atomare Datentypen (String, Integer, Date, ...)
- Datenstruktur für Cube
  - Für  $d$  Dimensionen:  $d$ -dimensionaler Raum
  - Bei  $m$  Werten in einer Dimension: Aufteilung des Würfels in  $m$  parallele Ebenen => endliche gleichgroße Liste von Ebenen je Dimension
  - Zelle eines  $d$ -dimensionalen Cubes wird eindeutig über  $d$  Dimensionswerten identifiziert
  - Pro Kennzahl in Zelle ein entsprechendes Array

302

---

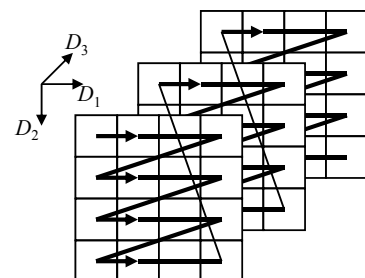
## 6.2 Datenmodellierung

---

### Multidimensionale Umsetzung (cont.)

- Speicherung des Data-Cube:
  - Linearisierung des  $d$ -dimensionalen Arrays in ein 1-dimensionales Array
  - Koordinaten der Würfelzellen (Dimensionen) „entsprechen“ Indizes des Arrays
  - Indexrechnung für Zelle mit Koordinaten  $z = x_1, \dots, x_d$

$$\begin{aligned} \text{Index}(z) &= x_1 + (x_2 - 1) \cdot |D_1| \\ &+ (x_3 - 1) \cdot |D_1| \cdot |D_2| \\ &\quad \vdots \\ &+ (x_d - 1) \cdot |D_1| \cdot \dots \cdot |D_{d-1}| \end{aligned}$$



303

## 6.2 Datenmodellierung

---

### Multidimensionale Umsetzung (cont.)

- Vorteile
  - Direkte OLAP-Unterstützung
  - Analytische Mächtigkeit
- Grenzen
  - Hohe Zahl an Plattenzugriffen bei ungünstiger Linearisierungsreihenfolge
  - Durch die Ordnung der Dimensionswerte (für Array-Abbildung nötig) keine einfache Änderung an Dimensionen möglich
  - Kein Standard für multidimensionale DBMS
- Oft: Hybride Speicherung HOLAP = MOLAP + ROLAP
  - Relationale Speicherung der Datenbasis
  - Multidimensionale Speicherung für häufig aggregierte Daten (z.B. angefragte (Teil-)Data Cubes)

304

---

## 6 Einführung in Data Warehouses

---

### Übersicht

6.1 Einleitung

6.2 Datenmodellierung

6.3 Anfragebearbeitung

305

## 6.3 Anfragebearbeitung

---

### Motivation

- Typische Anfragen an Data Warehouses beinhalten Aggregationen  
*Wieviele Artikel der Produktgruppe Elektrogeräte wurden im Januar 2000 pro Tag in den einzelnen Regionen in Bayern verkauft?*
- Charakteristik typischer DW-Anfragen
  - Große Menge vorhandener Fakten
  - Daraus nur ein bestimmter, in den meisten Dimensionen beschränkter Datenbereich angefragt
  - Problem: Aggregation auf großen Datenmengen  
z.B. 1TB Verkaufsdaten komplett einlesen dauert bei einer Leserate von 200 MB/s:  $1000000/200 \text{ s} = 5000 \text{ s}$   
d.h. ca. 83 min=> inakzeptabel!!!

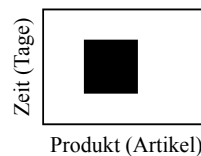
306

## 6.3 Anfragebearbeitung

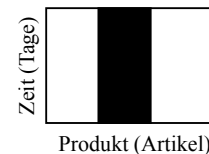
---

### Multidimensionale Anfragen

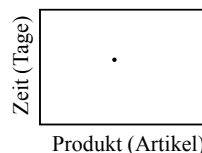
- Bereichsanfrage (range query)



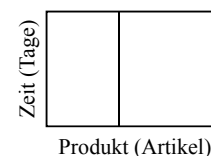
- Partielle Bereichsanfrage (partial range query)



- Punktanfrage (match query)



- Partielle Punktanfrage (partial match query)



- ...

307

## 6.3 Anfragebearbeitung

### Umsetzung

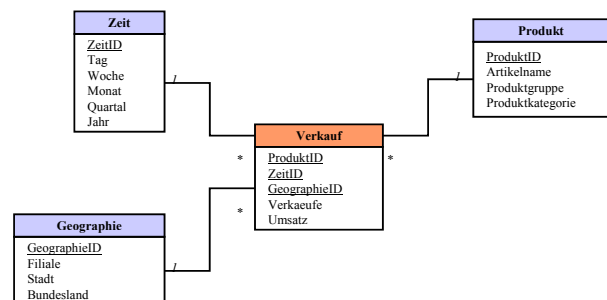
- Grundsätzlich abhängig von der Umsetzung des Schemas
  - Star Schema vs. Snowflake Schema
  - Multidimensionale Speicherung
- Häufige Anfrage-Muster auf relationaler Umsetzung
  - ( $n+1$ )-Wege-Join zwischen
    - $n$  Dimensionstabellen
    - Faktentabelle
  - Restriktionen über Dimensionstabellen  
(z.B. Region = Deutschland, Produktgruppe = Elektrogeräte, Jahr = 2000)

308

## 6.3 Anfragebearbeitung

### Star Join

- Beispiel  
*Wieviele Artikel der Produktgruppe  
Elektrogeräte wurden im Januar 2000  
pro Tag in den einzelnen Regionen  
in Bayern verkauft?*



```
SELECT      Geographie.Bundesland, Zeit.Monat, SUM(Verkaeufe)
FROM        Produkte, Zeit, Geographie, Verkauf
WHERE       Verkauf.ProduktID = Produkt.ProduktID
AND         Verkauf.ZeitID = Zeit.ZeitID
AND         Verkauf.GeographieID = Geographie.GeographieID
AND         Produkt.Produktgruppe = 'Elektrogeraete'
AND         Geographie.Bundesland = 'Bayern'
AND         Zeit.Monat = 'Januar 2000'

GROUP BY   Geographie.Region, Zeit.Tag
```

309

## 6.3 Anfragebearbeitung

---

### Star Join

- Allgemein:
  - SELECT-Klausel
    - Kenngrößen (evtl. aggregiert)
    - Ergebnisgranularität (der Dimensionen)  
z.B. Zeit.Monat, Geographie.Region
  - FROM-Klausel
    - Faktentabelle
    - Dimensionstabellen
  - WHERE-Klausel
    - Verbundbedingungen
    - Restriktionen in Dimensionen  
z.B. Produkt.Produktgruppe = 'Elektrogeraete', Zeit.Monat=  
'Januar 2000', ...

310

## 6.3 Anfragebearbeitung

---

### Star Join: Optimierung

- Star-Join ist ein typisches Muster für Anfragen in Data Warehouses
- Typische Charakteristik (wegen Star Schema)
  - Sehr große Faktentabelle
  - Relativ kleine, unabhängige Dimensionstabellen
- Heuristiken klassischer relationaler Optimierer schlagen hier meist fehl
  - Optimieren unter der Annahme, dass alle Relationen etwa gleich groß sind



## 6.3 Anfragebearbeitung

---

### Star Join: Optimierung (cont.)

- Auswertungsplan?
  - 4-Wege Join (Join über 4 Tabellen: Verkauf, Produkt, Zeit, Geographie)
  - In relationalen DBS kann typischerweise nur paarweise gejoint werden => Sequenz paarweiser Joins
  - Es gibt  $4! = 24$  viele mögliche Join-Reihenfolgen (= mögliche Auswertungspläne)
  - Heuristik zur Verringerung der Anzahl der Möglichkeiten:
    - Joins zwischen Relationen, die nicht durch Join-Bedingung in Anfrage verknüpft, NICHT betrachten

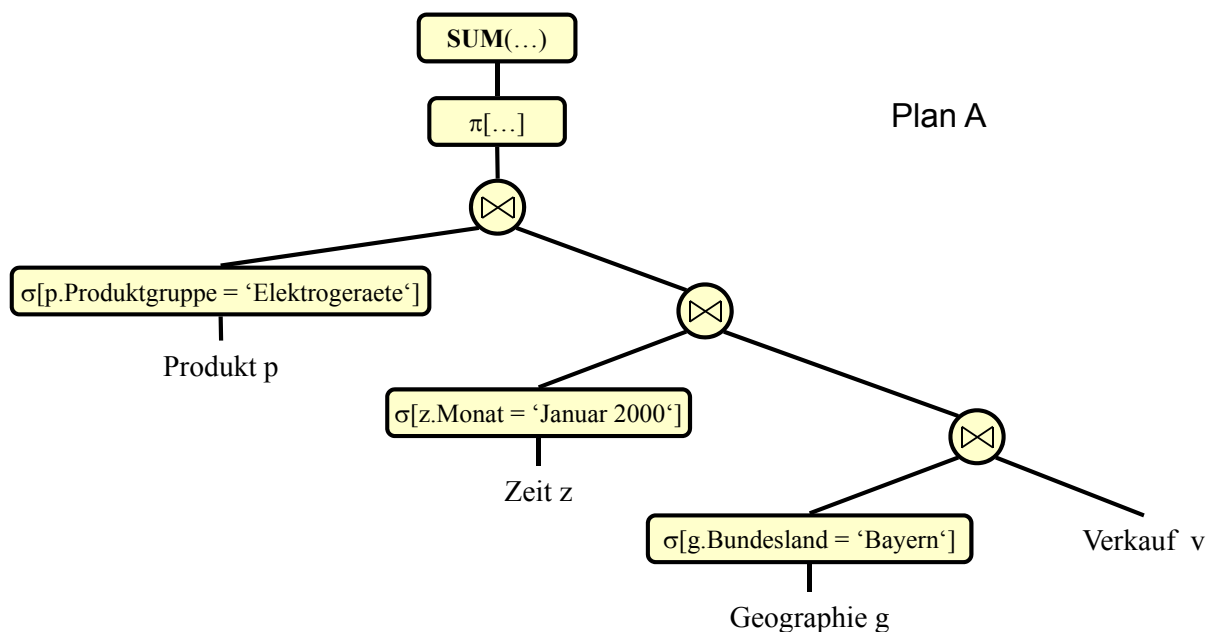
312

## 6.3 Anfragebearbeitung

---

### Star Join: Optimierung (cont.)

- Optimierter kanonischer Auswertungsplan:



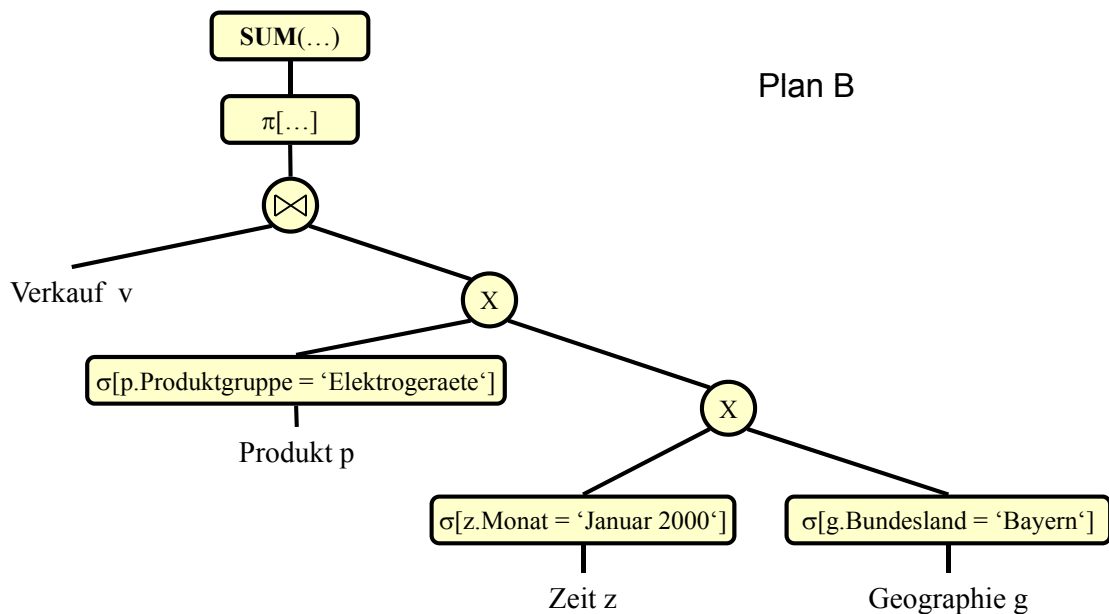
313

## 6.3 Anfragebearbeitung

---

### Star Join: Optimierung (cont.)

- Alternativer Auswertungsplan, der üblicherweise nicht betrachtet wird



314

## 6.3 Anfragebearbeitung

---

### Star Join: Optimierung (cont.)

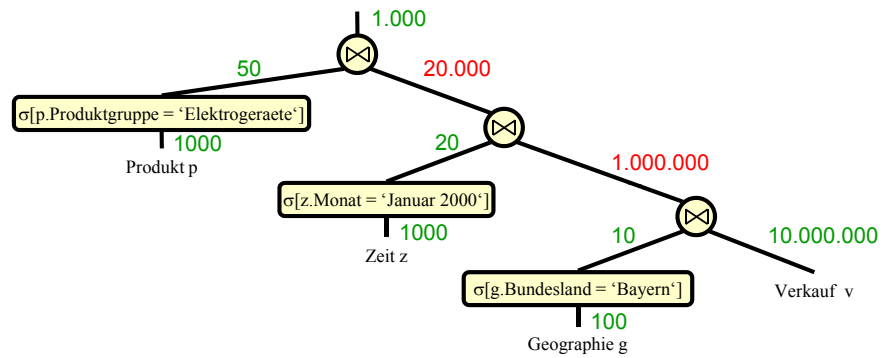
- Vergleich von Plan A und B
- Szenario:
  - Tabelle Verkauf: 10.000.000 Datensätze
  - 10 Geschäfte in Bayern (von 100)
  - 20 Verkaufstage im Januar 2000 (von 1000 gespeicherten Tagen)
  - 50 Produkte in Produktgruppe „Elektrogeräte“ (von 1000)
  - Gleichverteilung/gleiche Selektivität der einzelnen Ausprägungen

315

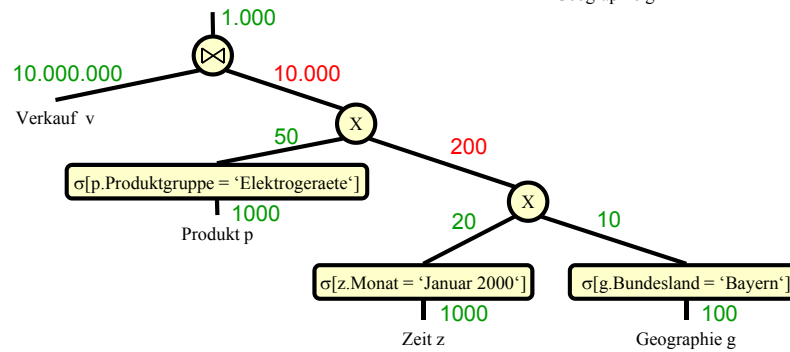
## 6.3 Anfragebearbeitung

### Star Join: Optimierung (cont.)

- Plan A



- Plan B



316

## 6.3 Anfragebearbeitung

### Roll-UP/Drill-Down

Verdichtungsgrad wird durch **GROUP BY**-Klausel spezifiziert:

- Mehr Attribute in **GROUP BY**  
=> weniger starke Verdichtung  
=> Drill-Down
- Weniger Attribute in **GROUP BY**  
=> stärkere Verdichtung  
=> Roll-Up

317

## 6.3 Anfragebearbeitung

---

### Weitere Optimierungs-Heuristiken

- Materialisierung von Aggregaten
  - Aggregation ist sehr zeitaufwendig
  - Berechne häufig verwendete Aggregationen einmal und materialisiere deren Ergebnis
- **CUBE**-Operator (z.B in SQL-Server, DB2, Oracle 8i)
  - Aggregationen mit drill-down/roll-up entlang aller Dimensionen, die in **GROUP BY**-Klausel vorkommen
  - Ermöglicht einfachere Formulierung dieser Aggregation
  - Ermöglicht Optimierung dieser Aggregation
    - „normalerweise“ müsste Faktentabelle mehrmals gelesen werden, da Aggregation durch mehrere mit **UNION** verknüpfte Unteranfragen berechnet werden müsste
    - Durch **CUBE**-Operator: nur einmaliges lesen der Faktentabelle