
Kapitel 6

Einführung in Data Warehouses

Skript zur Vorlesung: Datenbanksysteme II
Sommersemester 2008, LMU München

© 2008 Dr. Peer Kröger

Dieses Skript basiert zu einem Teil auf dem Skript zur Vorlesung Datenbanksysteme II von Prof. Dr. Christian Böhm gehalten im Sommersemester 2007 an der LMU München

Übersicht

6.1 Einleitung

6.2 Datenmodellierung

6.3 Anfragebearbeitung

Übersicht

6.1 Einleitung

6.2 Datenmodellierung

6.3 Anfragebearbeitung

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

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

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)

Operationales DBS vs. Data Warehouse

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

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]

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.

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. Verwaltung von Personaldaten)
- System dient der Modellierung eines spezifischen Anwendungsziel

– Integrierte Datenbasis (**integrated**)

- Verarbeitung der Daten aus unterschiedlichen Datenquellen

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

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]

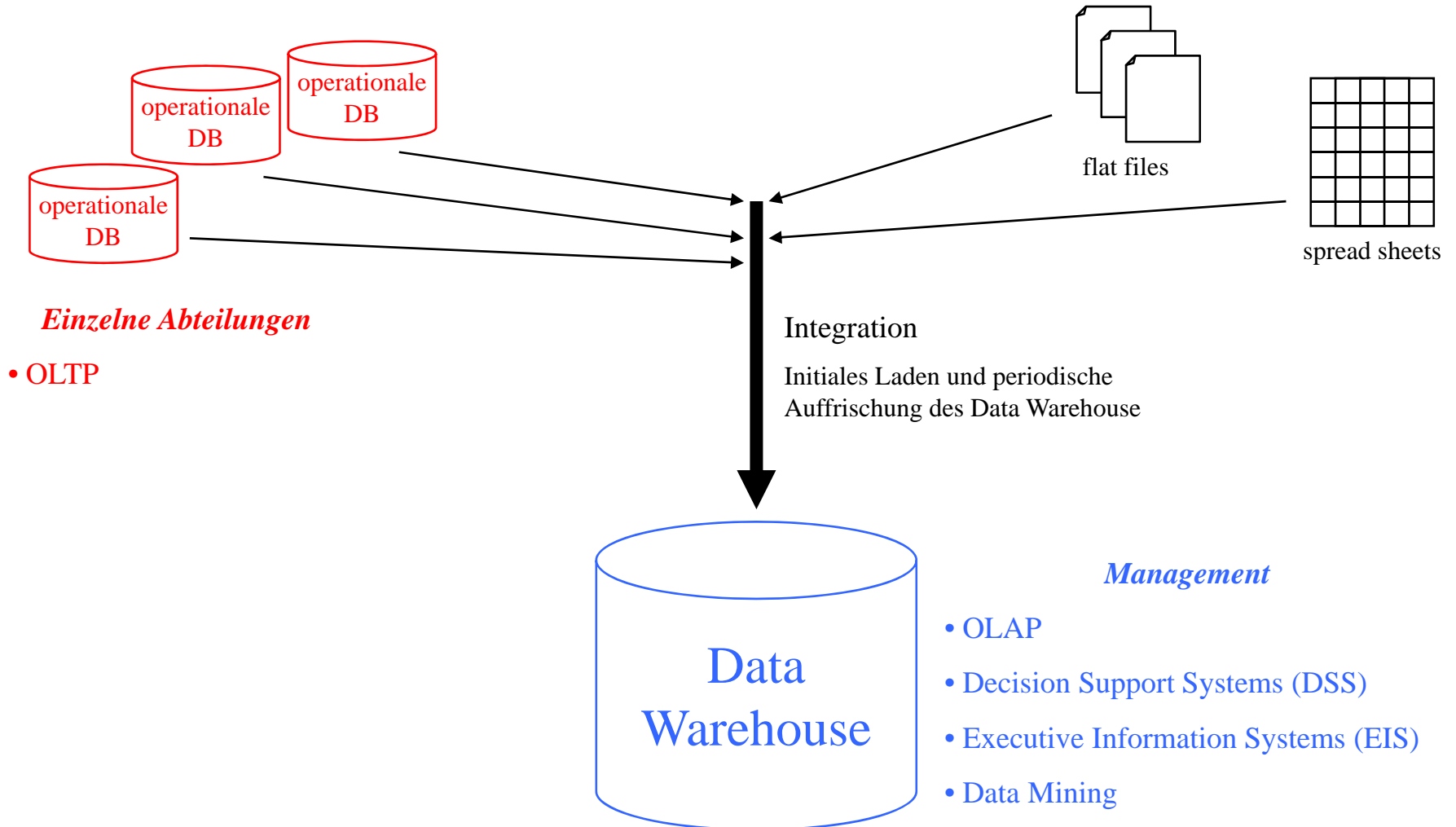
– Nicht-flüchtige Datenbasis (non-volatile)

- 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

Architektur eines Data Warehouse



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

Übersicht

6.1 Einleitung

6.2 Datenmodellierung

6.3 Anfragebearbeitung

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**

Kennzahlen/Fakten

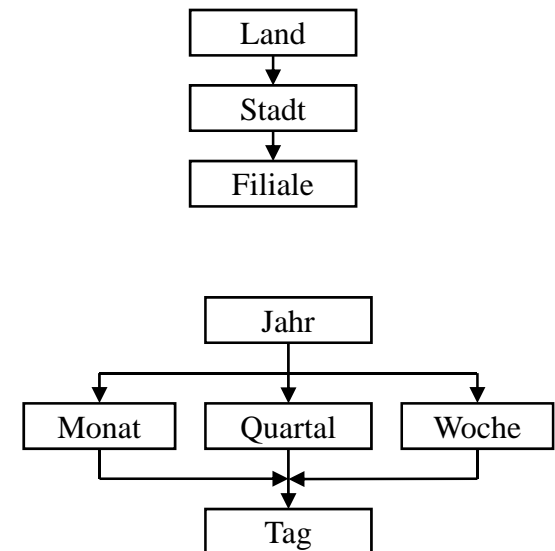
- Kennzahlen/Fakten
 - Numerische Messgrößen
 - Beschreiben betriebswirtschaftliche Sachverhalte
- Beispiele: Umsatz, Gewinn, Verlust, ...
- Typen
 - Additiv: (additive) Berechnung zwischen sämtlichen Dimensionen möglich (z.B. Bestellmenge eines Artikels)
 - Semi-additiv: (additive) Berechnung möglich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

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

Hierarchien in Dimensionen

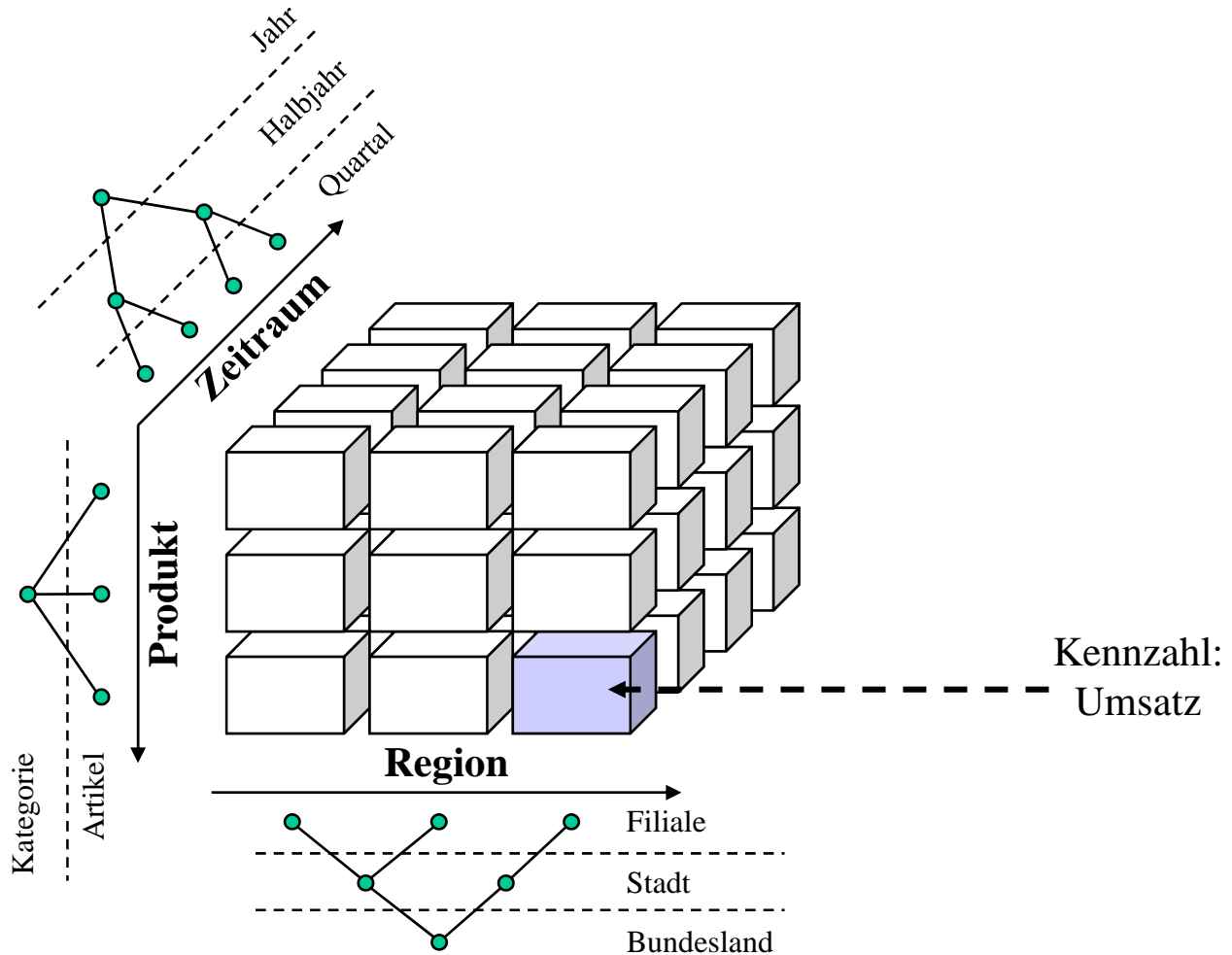
- Dimensionselemente sind Knoten einer Klassifikationshierarchie
- Klassifikationsstufe beschreibt Verdichtungsgrad
- Darstellung von Hierarchien in Dimensionen über Klassifikationsschema
- Formen
 - Einfache Hierarchien: höhere Ebene enthält die aggregierten Werte genau einer niedrigeren Hierarchiestufe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



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

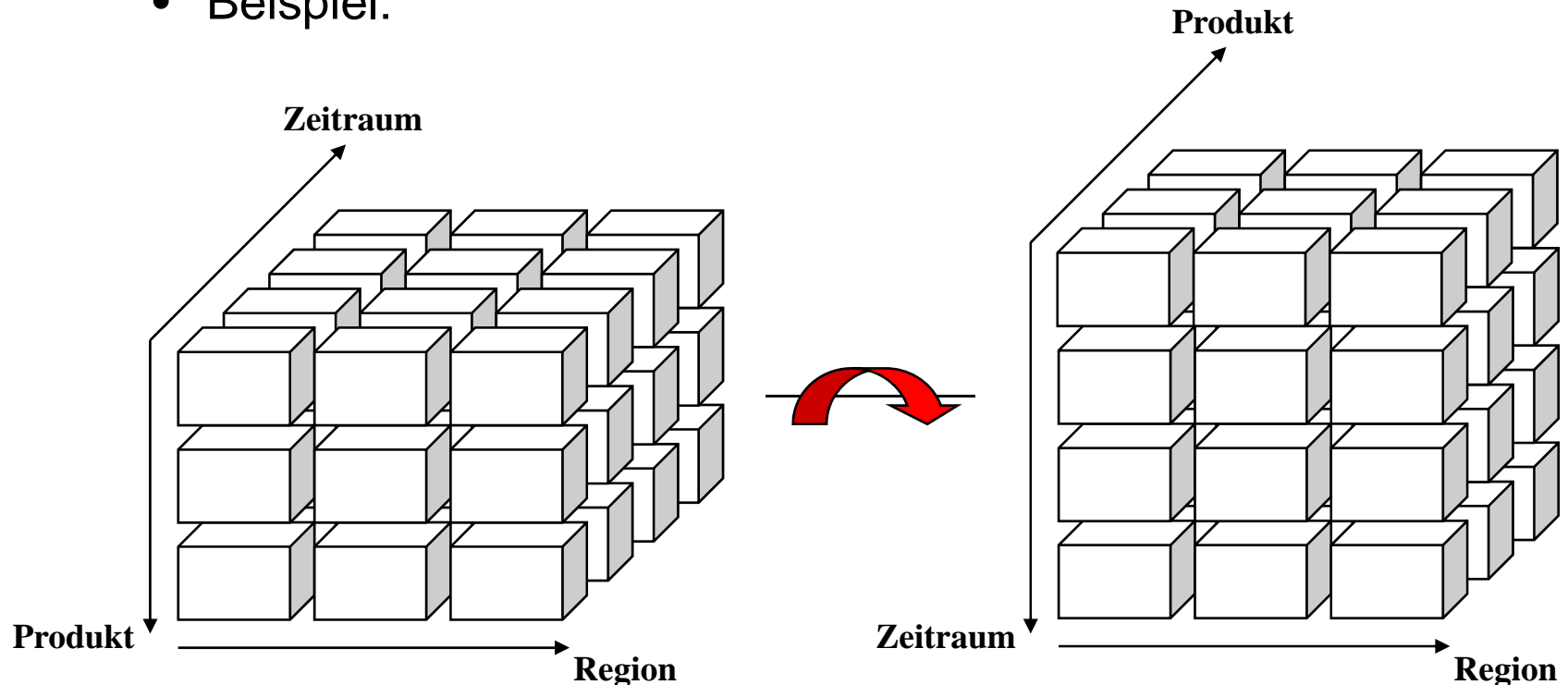
Beispiel: 3D Data-Cube



Standardoperationen zur Datenanalyse

– Pivotisierung/Rotation

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



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

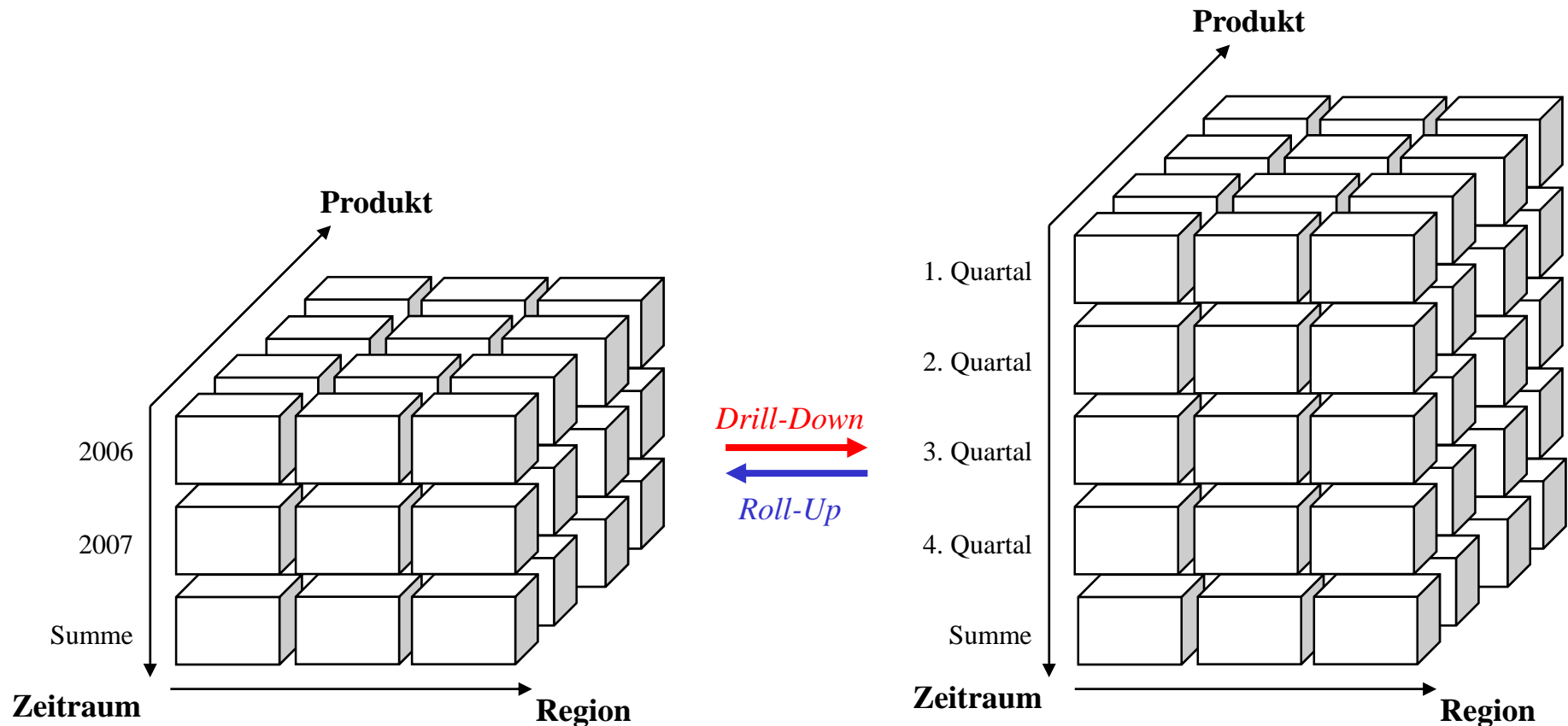
- Komplementär zu Roll-Up
- Navigation von aggregierten Daten zu Detail-Daten entlang der Klassifikationshierarchie

– Drill-Across

- Wechsel von einem Cube zu einem anderen

Standardoperationen zur Datenanalyse (cont.)

- Beispiel: Roll-Up, Drill-Down



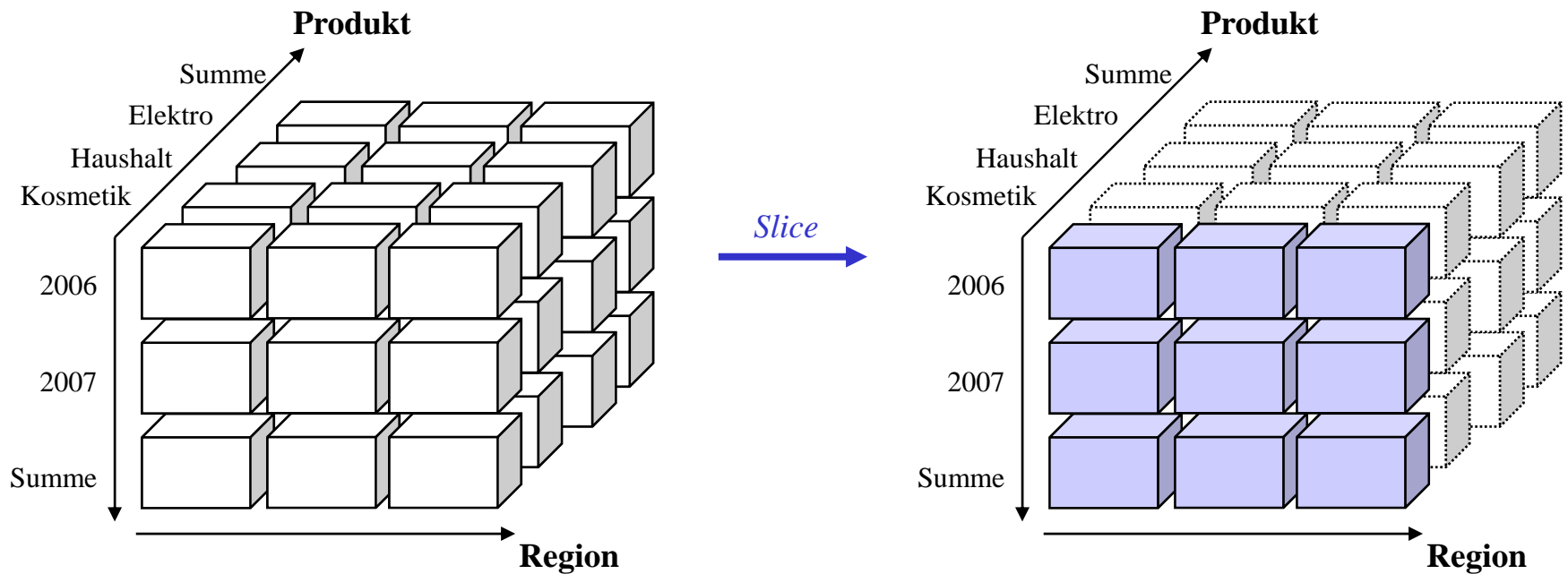
Standardoperationen zur Datenanalyse (cont.)

– 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

Standardoperationen zur Datenanalyse (cont.)

- Beispiel: Slice



Umsetzung des multidimensionalen Modells

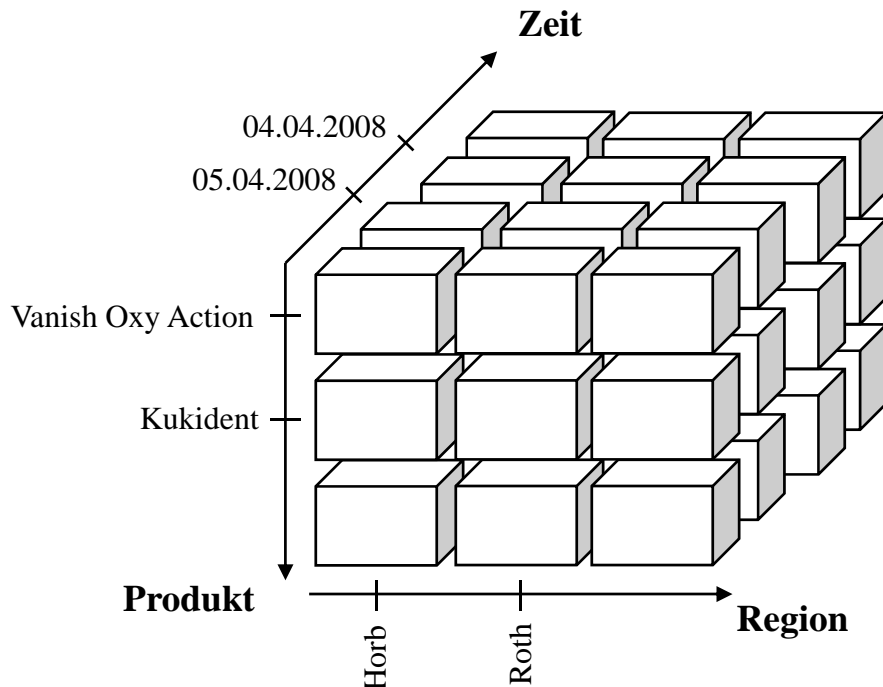
- Interne Verwaltung der Daten durch
 - Relationale Strukturen (Tabellen)
 - Relationales OLAP (ROLAP)
 - Vorteile: Verfügbarkeit, Reife der Systeme
 - Multidimensionale Strukturen (direkte Speicherung)
 - Multidimensionales OLAP (MOLAP)
 - Vorteil: Wegfall der Transformation
- Wichtige Designaspekte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

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

Relationale Umsetzung: Faktentabelle

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

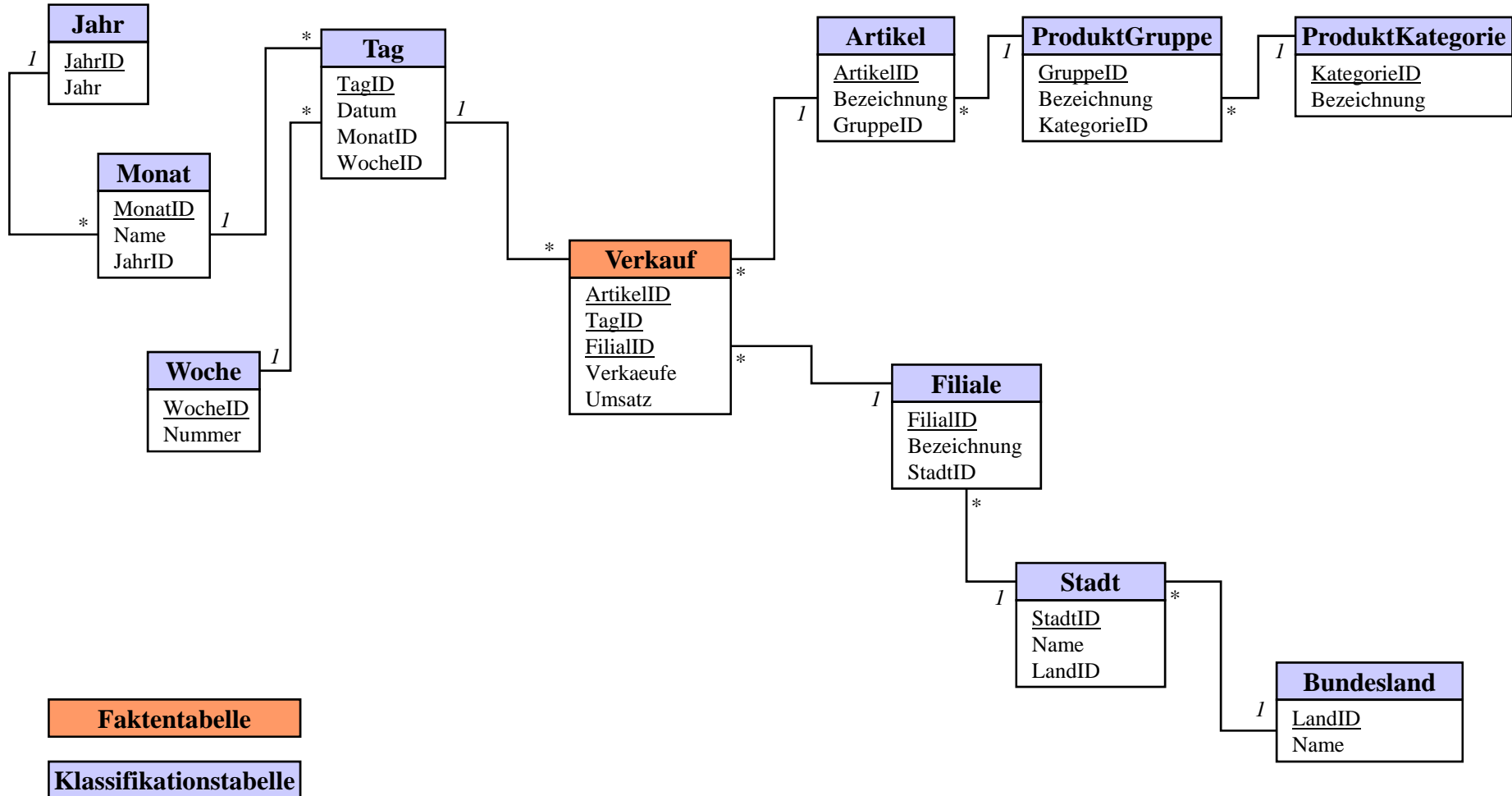


<i>Dimensionen</i>			<i>Kennzahl</i>
Artikel	Filiale	Tag	Verkäufe
Vanish Ox. A.	Horb	04.04.2008	4
Vanish Ox. A.	Horb	05.04.2008	1
Kukident	Horb	04.04.2008	12
Kukident	Roth	04.04.2008	0
Vanish Ox. A.	Roth	05.04.2008	2
			⋮

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

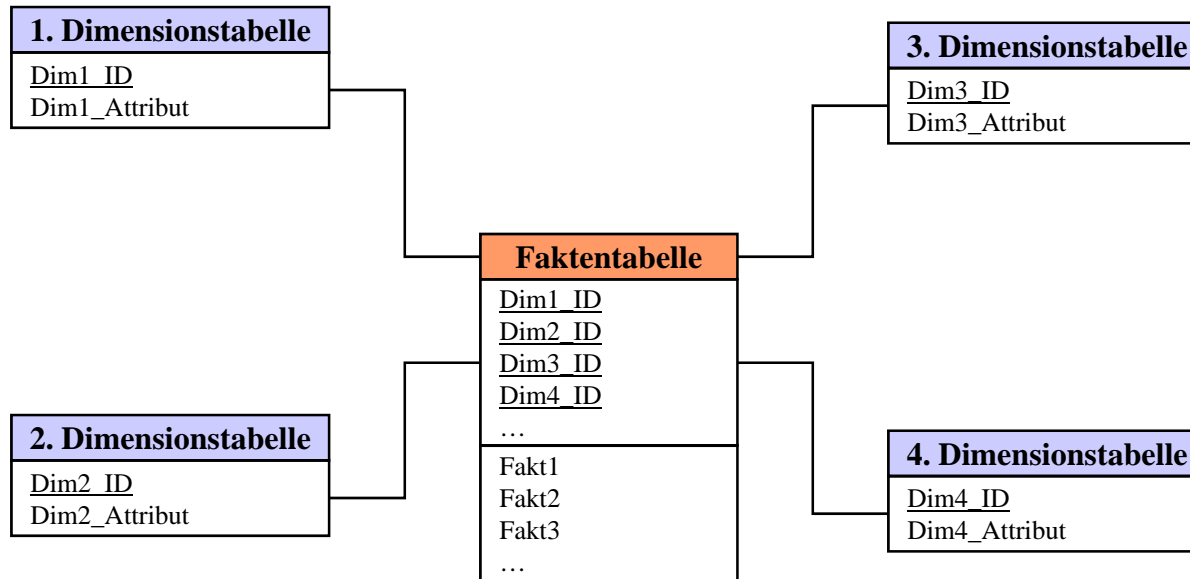
Snowflake Schema: Beispiel



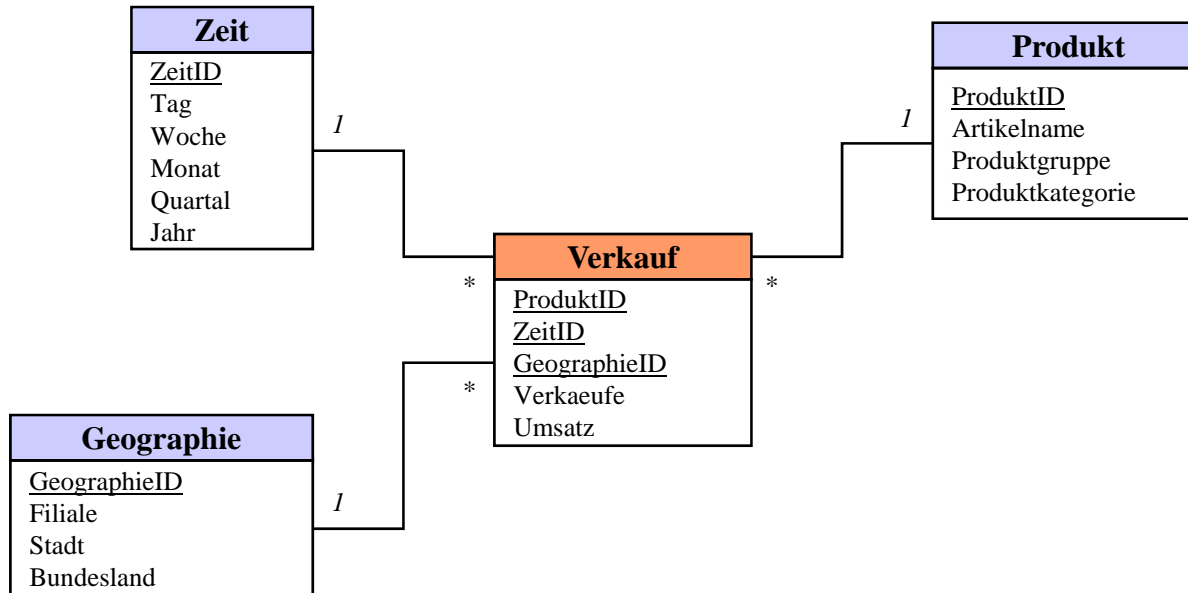
Relationale Umsetzung: Star Schema

- 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

Star Schema: Visualisierung



Star Schema: Beispiel



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
 - ...

Relationale Umsetzung: Begriffe

- 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)

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 ???

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)

Multidimensionale Umsetzung (cont.)

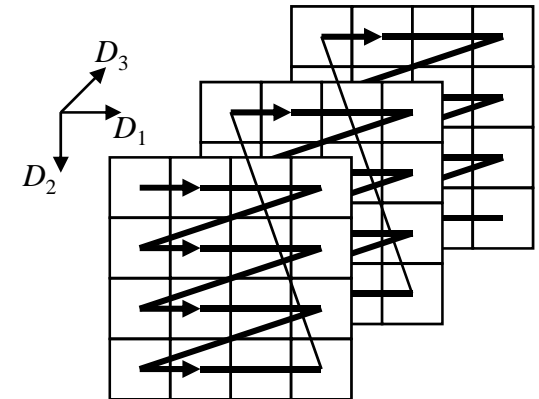
- Datenstruktur für eine Dimension
 - Endliche geordnete Liste von Dimensionswerten (aller Klassifikationsstufen)
 - Dimensionswerte: einfache, atomare Datentypen (String, Integer, Date, ...)
- Datenstruktur für Cube
 - Für d Dimensionen: d -dimensionaler Raum
 - Bei m Werten in einer Dimension: Aufteilung 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

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
- Indexberechnung für Zelle mit Koordinaten $z = x_1, \dots, x_d$

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



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)