

Teil III

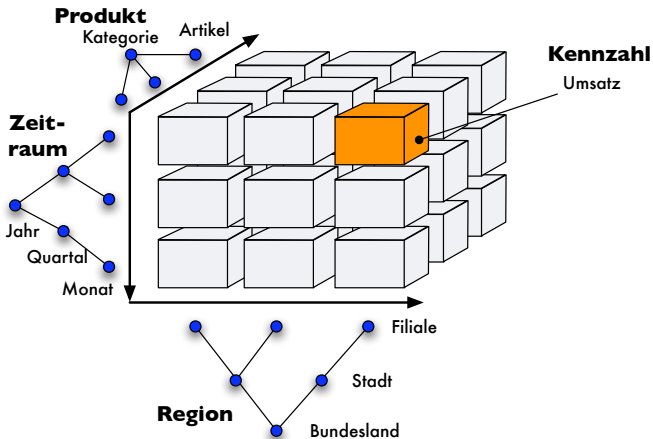
Multidimensionales Datenmodell

Multidimensionales Datenmodell

- 1 Grundbegriffe
- 2 Der Würfel
- 3 Konzeptuelle Modellierung
- 4 Operationen zur Datenanalyse
- 5 Relationale Umsetzung des multidimensionalen Datenmodells

Grundbegriffe

- Dimensionen
- Fakten / Kennzahlen



Motivation

- Datenmodell ausgerichtet auf Unterstützung der Analyse
- Datenanalyse im Entscheidungsprozess
 - ▶ Betriebswirtschaftliche Kennzahlen stehen im Mittelpunkt
 - **Fakten**
 - ★ Gewinn
 - ★ Umsatz
 - ★ Kosten, etc.
 - ▶ Betrachtung der Kennzahlen aus unterschiedlichen Perspektiven
 - **Dimensionen**
 - ★ Zeit
 - ★ Raum
 - ★ Sache
 - ▶ Unterteilung der Auswertedimensionen möglich
 - **Hierarchien** oder **Konsolidierungsebenen**
 - ★ Jahr
 - ★ Quartal
 - ★ Monat

Verfügbare Informationen

- Qualifizierend
 - ▶ Repräsentiert durch „Kategorienattribute“
 - ▶ Daten zur Nutzung als Navigationsraster („Drill-Pfade“)
 - ▶ Modelliert als Begriffshierarchien im Rahmen von Dimensionen
- Quantifizierend
 - ▶ Bilden Gegenstand der Auswertung („Summenattribute“ oder andere arithmetische Operationen)
 - ▶ Zellen eines Würfels, mit Dimensionen als Kanten

Dimensionen

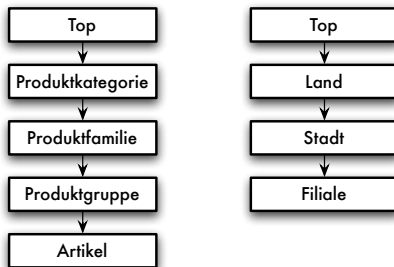
- Dimension:
 - ▶ Beschreibt mögliche Sicht auf assoziierte Kennzahlen
 - ▶ Endliche Menge von n ($n \geq 2$) Dimensionselementen (Hierarchieobjekten), die eine semantische Beziehung aufweisen
 - ▶ Dienen der orthogonalen Strukturierung des Datenraums
- Beispiele:
 - ▶ Produkt,
 - ▶ Filialstruktur,
 - ▶ Geschäftsjahr

Hierarchien in Dimensionen

- Dimensionselemente:
 - ▶ Knoten einer Klassifikationshierarchie
 - ▶ Klassifikationsstufe beschreibt Verdichtungsgrad
 - ▶ Darstellung von Dimensionen über Klassifikationsschema (Schema von Klassifikationshierarchien)
- Formen:
 - ▶ Einfache Hierarchien
 - ▶ Parallele Hierarchien

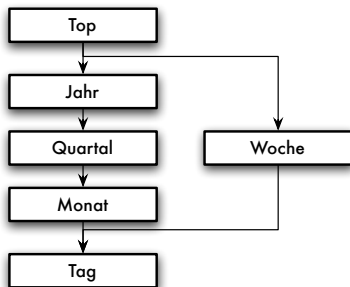
Einfache Hierarchie

- Höhere Hierarchieebene enthält die aggregierten Werte genau einer niedrigeren Hierarchiestufe
- Oberster Knoten: *Top*
 - ▶ Enthält Verdichtung auf einen einzelnen Wert für die Dimension



Parallele Hierarchie

- Innerhalb einer Dimension sind mehrere unabhängige Arten der Gruppierung möglich
- Keine hierarchische Beziehung zwischen parallelen Zweigen
- Parallelhierarchie
 - ▶ Pfad im Klassifikationsschema
 - ▶ Konsolidierungspfad



Schema einer Dimension D

- Partiiell geordnete Menge von Kategorienattributen ($\{D^1, \dots, D^n, Top_D\}; \rightarrow$)
 - ▶ Generisches maximales Element Top_D
 - ▶ Funktionale Abhängigkeit \rightarrow
- Top_D wird von allen Attributen funktional bestimmt:

$$\forall i, 1 \leq i \leq n : D_i \rightarrow Top_D$$

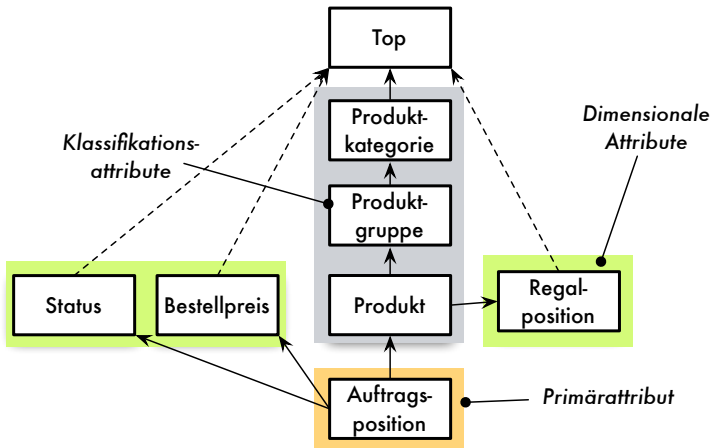
- Es gibt genau ein D_i , das alle anderen Kategorieattribute bestimmt
 - ▶ Gibt feinste Granularität einer Dimension vor

$$\exists i, 1 \leq i \leq n, \forall j, 1 < j \leq n, i \neq j : D_i \rightarrow D_j$$

Kategorienattribute

- Inhaltliche Verfeinerung durch unterschiedliche Rollen:
- Primärattribut
 - ▶ Kategorienattribut, das alle anderen Attribute einer Dimension bestimmt
 - ▶ Definiert maximale Feinheit
 - ▶ Beispiel: „Auftragsposition“
- Klassifikationsattribut
 - ▶ Element der Menge, die mehrstufige Kategorisierung (Klassifikationshierarchie) bilden
 - ▶ Beispiel: „Produkt“, „Produktgruppe“, „Produktkategorie“
- Dimensionales Attribut
 - ▶ Element der Menge der Attribute, die vom Primärattribut oder einem Klassifikationsattribut bestimmt werden und nur Top_D bestimmen
 - ▶ Beispiel: „Regalposition“

Struktur einer Dimension: Beispiel



Kennzahlen

- Kennzahlen / Fakten (engl. *facts*):
 - ▶ (Verdichtete) numerische Messgrößen
 - ▶ Beschreiben betriebswirtschaftliche Sachverhalte
- **Fakt**: Maßzahl (engl. *measure*)
- **Kennzahl**:
 - ▶ Aus Fakten konstruiert (abgeleitete Kennzahl)
 - ▶ Durch Anwendung arithmetischer Operationen
- Beispiele:
 - ▶ Umsatz, Gewinn, Kosten
 - ▶ Deckungsbeitrag, ROI (Return on Investment)
 - ▶ Fluktuationsquote, Umsatzsteigerung

Fakt: Schema

- Schema wird durch mehrere Bestandteile spezifiziert
- **Granularität** $G = \{G_1, \dots, G_k\}$
 - ▶ G ist Teilmenge aller Kategorienattribute aller im Schema existierenden Dimensionsschemata DS_1, \dots, DS_n
 - ★ $\forall i, 1 \leq i \leq k, \exists j, 1 \leq j \leq n : G_i \in DS_j$
 - ★ $\forall i, 1 \leq i \leq k, \forall j, 1 \leq j \leq k, i \neq j : G_i \not\rightarrow G_j$
(keine funktionalen Abhängigkeit zwischen Kategorienattributen einer Granularität)
 - ▶ „Detailliertheitsgrad“ der Fakten
- Summationstyp *SumTyp*
 - ▶ Bestandsgröße
 - ▶ Stromgröße
 - ▶ Einheit

Kennzahl

- Kennzahl M ist definiert durch
 - ▶ Granularität G
 - ▶ Berechnungsvorschrift $f()$ über Fakten
 - ▶ Summationstyp $SumTyp$
- Berechnung über nichtleerer Teilmenge der im Schema existierenden Fakten
- $M = (G, f(F_1, \dots, F_k), SumTyp)$

Kennzahl: Bildung von $f()$

- Skalarfunktionen

- ▶ $+$, $-$, $*$, $/$, mod

- ▶ Beispiel: $Umsatzsteueranteil = Menge * Preis * Steuersatz$

- Aggregatfunktionen

- ▶ Funktion $H()$ zur Verdichtung eines Datenbestandes, indem aus n Einzelwerten ein Aggregatwert ermittelt wird

$$H : 2^{dom(X_1) \times \dots \times dom(X_n)} \rightarrow dom(Y)$$

- ▶ Bsp.: $SUM()$, $AVG()$, $MIN()$, $MAX()$, $COUNT()$

- Ordnungsbasierte Funktionen

- ▶ Definition von Kennzahlen auf Basis zuvor definierter Ordnungen

- ▶ Bsp.: Kumulation, $TOP(n)$, $MEDIAN()$

Summationstyps

- Zuweisung eines **Summationstyps** charakterisiert erlaubte Aggregationsoperationen
- *FLOW*
 - ▶ Maß zum Zeitpunkt
 - ▶ Beliebig aggregierbar
 - ▶ Beispiel: Bestellmenge eines Artikels pro Tag
- *STOCK*
 - ▶ Maß über Zeitraum
 - ▶ Beliebig aggregierbar mit Ausnahme temporaler Dimension
 - ▶ Beispiel: Lagerbestand, Einwohnerzahl
- *VALUE – PER – UNIT (VPU)*
 - ▶ Maß mit Einheit zum Zeitpunkt
 - ▶ Aktuelle Zustände, die nicht summierbar sind
 - ▶ Zulässig nur: *MIN()*, *MAX()*, *AVG()*
 - ▶ Beispiele: Preis, Wechselkurs, Steuersatz

Summierbarkeit

| | <i>FLOW</i> | <i>STOCK</i> | | <i>VPU</i> |
|----------------|-------------|---------------------------------------|----|------------|
| | | Aggregation über temporale Dimension? | | |
| | | nein | ja | |
| <i>MIN/MAX</i> | + | + | | + |
| <i>SUM</i> | + | + | - | - |
| <i>AVG</i> | + | + | | + |
| <i>COUNT</i> | + | + | | + |

Disjunktheit

- Konkreter Wert einer Kennzahl geht **exakt einmal** in Ergebnis ein
- Bsp.: Studierende im Grundstudium

| Studierende | 2008 | 2009 | 2010 | Gesamt |
|--------------------|-------------|-------------|-------------|---------------|
| Informatik | 15 | 17 | 13 | 28 |
| BWL | 10 | 15 | 11 | 21 |
| Gesamt | 25 | 31 | 22 | 49 |

- Anzahl der Informatik-Studenten (bei 2-jährigem Grundstudium)?
- Anzahl der Studierenden insgesamt (bei Doppelstudium)?

Vollständigkeit

- Kennzahlen auf höherer Aggregationsebene lassen sich komplett aus Werten tieferer Stufen berechnen

| Restaurants | 2005 | 2006 |
|--------------------|-------------|-------------|
| Magdeburg | 52 | 50 |
| Ilmenau | 45 | 46 |
| Sonstige | 20 | 22 |
| <i>Gesamt</i> | <i>117</i> | <i>118</i> |

- Sonstige = Landgasthöfe im Umland

Aggregatfunktionen

X ist Klassifikationsknoten und (X_1, X_2, \dots, X_n) ist Partitionierung

- **Distributive Aggregatfunktion:**
 $\exists g : f(X) = f(g(X_1), g(X_2), \dots, g(X_n))$
- **Algebraische Aggregatsfunktion:** f ist berechnbar aus fester Menge G
- **Holistische Aggregatsfunktion:** f kann nur aus den Grundelementen von X berechnet werden

| Aggregatstyp | Beispiel |
|--------------|--|
| Distributiv | $SUM()$, $COUNT()$, $MAX()$, $MIN()$ |
| Algebraisch | $AVG()$ mit $g_1 := SUM()$ und $g_2 := COUNT()$, $STDDEV()$ |
| Holistisch | $MEDIAN()$, $RANK()$, $PERCENTILE()$ |

Der Würfel

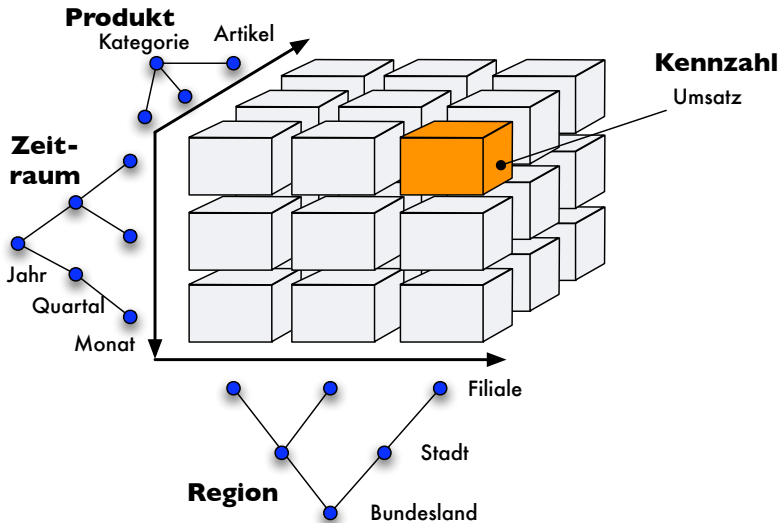
- **Würfel** (engl. **cube**, eigentlich Quader): Grundlage der multidimensionalen Analyse
- Kanten → **Dimensionen**
- Zellen → ein oder mehrere **Kennzahlen** (als Funktion der Dimensionen)
- Anzahl der Dimensionen → **Dimensionalität**
- Visualisierung
 - ▶ 2 Dimensionen: Tabelle
 - ▶ 3 Dimensionen: Würfel
 - ▶ > 3 Dimensionen: Multidimensionale Domänenstruktur
- Schema C eines Würfels
 - ▶ Menge der Dimensionen(-schemata) DS
 - ▶ Menge der Kennzahlen M
- $C = (DS, M) = (\{D^1, \dots, D^n\}, \{M^1, \dots, M^m\})$

Orthogonalität

- In multidimensionalen Schemata gilt **Orthogonalität**, d.h.
 - ▶ Keine funktionalen Abhängigkeiten zwischen Attributen unterschiedlicher Dimensionen

$$\forall i, 1 \leq i \leq n, \forall j, 1 \leq j \leq n, i \neq j \neg \exists k, l : D^i . D_k \rightarrow D^j . D_l$$

Multidimensionaler Datenwürfel



Konzeptuelle Modellierung

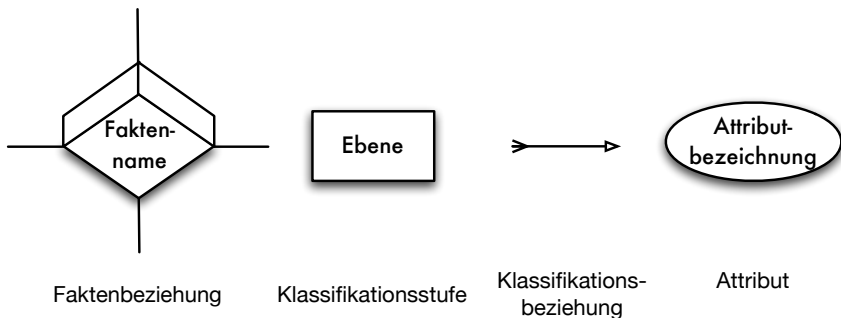
Formale Beschreibung des Fachproblems und der im Anwendungsbereich benötigten Informationsstrukturen

- Probleme konventioneller Entwurfstechniken (ER, UML):
 - ▶ Unzureichende Semantik für multidimensionales Datenmodell
 - ▶ Hier: Verzicht auf universelle Anwendbarkeit, stattdessen Konzentration auf Analyse
 - ▶ Beispiel: Klassifikationsstufe, Fakt → Entity ?

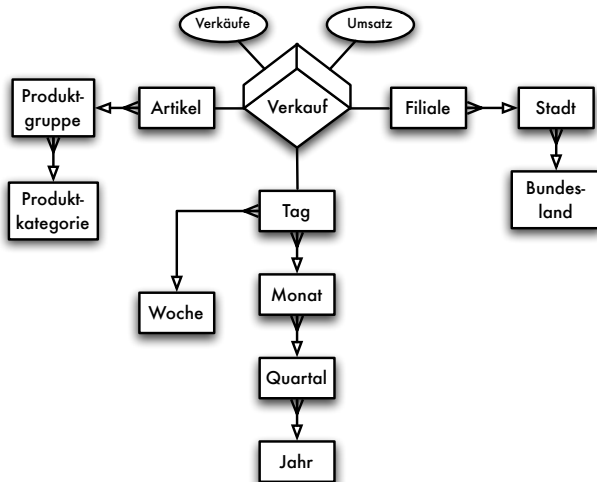
ME/R-Modell

- Multidimensional Entity/Relationship Model
[Sapia et. al. (1998)]
- Erweiterung des klassischen ER-Modells
 - ▶ Entity-Menge „Dimension Level“ (Klassifikationsstufe)
 - ★ Keine explizite Modellierung von Dimensionen
 - ▶ n-äre Beziehungsmenge „Fact“
 - ★ Kennzahlen als Attribute der Beziehung
 - ▶ Binäre Beziehungsmenge „Classification“ bzw. „Roll-Up“ (Verbindung von Klassifikationsstufen)
 - ★ Definiert gerichteten, nicht-zyklischen Graphen

ME/R: Notationen



ME/R: Beispiel

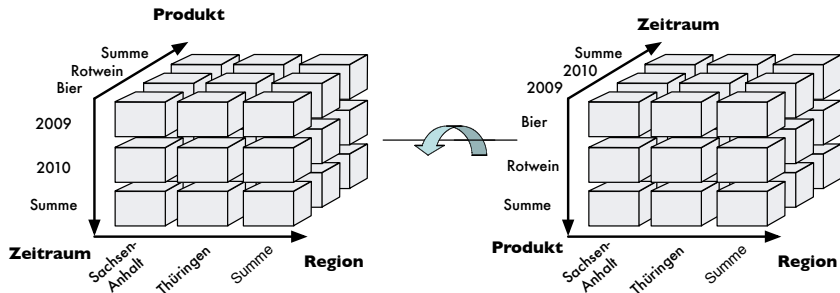


Operationen zur Datenanalyse

- OLAP-Operationen auf multidimensionalen Datenstrukturen
- Standardoperationen
 - ▶ Pivotierung
 - ▶ Roll-Up, Drill-Down
 - ▶ Drill-Across
 - ▶ Slice und Dice

Pivotierung / Rotation

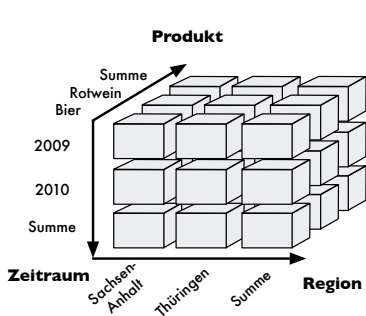
- Drehen des Würfels durch Vertauschen der Dimensionen
- Analyse der Daten aus verschiedenen Perspektiven



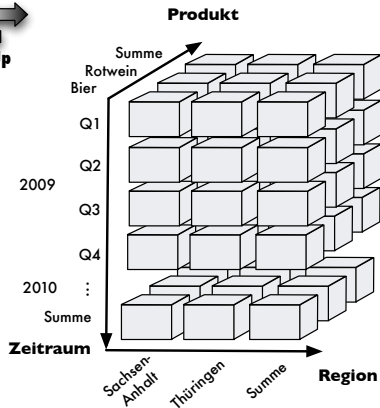
Roll-Up, Drill-Down, Drill-Across

- **Roll-Up:**
 - ▶ Erzeugen neuer Informationen durch Aggregation der Daten entlang des Konsolidierungspfades
 - ▶ Dimensionalität bleibt erhalten
 - ▶ Beispiel: Tag → Monat → Quartal → Jahr
- **Drill-Down:**
 - ▶ Komplementär zu Roll-Up
 - ▶ Navigation von aggregierten Daten zu Detail-Daten entlang der Klassifikationshierarchie
- **Drill-Across:**
 - ▶ Wechsel von einem Würfel zu einem anderen

Roll-Up und Drill-Down



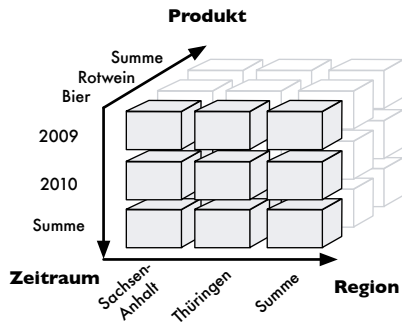
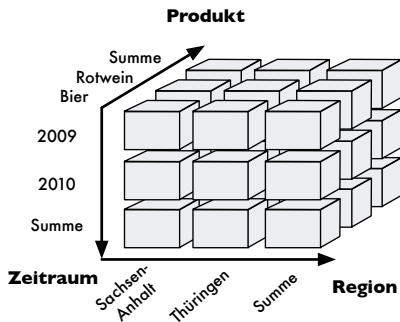
Drill Down
→
←
Roll Up



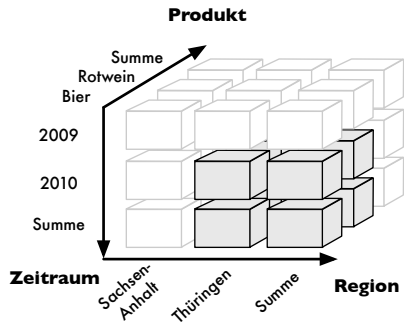
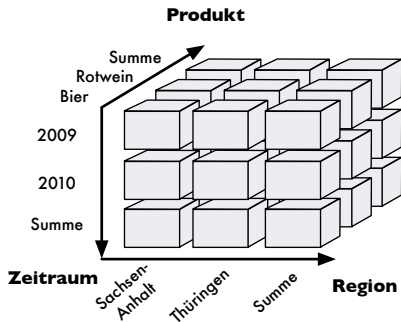
Slice und Dice

- Erzeugen individueller Sichten
- **Slice:**
 - ▶ Herausschneiden von „Scheiben“ aus dem Würfel
 - ▶ Verringerung der Dimensionalität durch Konditionierung der Dimensionen
 - ▶ Beispiel: alle Werte des aktuellen Jahres
 - ▶ *Entspricht der relationalen Selektion in den Dimensionen*
- **Dice:**
 - ▶ Herausschneiden einen „Teilwürfels“
 - ▶ Erhaltung der Dimensionalität, Veränderung der Hierarchieobjekte
 - ▶ Beispiel: die Werte bestimmter Produkte oder Regionen
 - ▶ *Entspricht der relationalen Selektion mehrerer Dimensionen*

Slice



Dice



Problem des Dimensionswechsel

- Austausch oder Weglassen einer Dimension kann Bestandteil der Analyse sein
- Beispiel: statt Cube by Produkt, Ort und Zeit kann Kunde, Ort und Zeit interessant sein.

| Produkt | Filiale | Tag | Verk. |
|--------------|-----------|----------|-------|
| Rotwein | Magdeburg | 18.02.10 | 145 |
| Weißbier | Magdeburg | 18.02.10 | 267 |
| Rotwein | Ilmenau | 18.02.10 | 70 |
| ... | | | |
| Kundengruppe | Filiale | Tag | Verk. |
| Weintrinker | Magdeburg | 18.02.10 | 210 |
| Bayer | Magdeburg | 18.02.10 | 407 |
| Genießer | Ilmenau | 18.02.10 | 35 |
| ... | | | |

Was bedeutet ein Wechsel der Dimension?

Marginalisierung

- Randsummenbildung bei Entfernen einer Dimension
- *Entspricht der relationalen Projektion*
- Abhängig vom Typen der Aggregatfunktion → Interpretationsproblem

| | Anteil Bierflaschen | |
|--------------|---------------------|---------|
| | Magdeburg | Ilmenau |
| 1. Halbwoche | 95% | 88% |
| 2. Halbwoche | 68% | 54% |
| GesamtWoche | 74% | 80% |

Simpsons Paradox

- Algebraische Funktion muss durch den Datenbestand (BDB) berechnet werden
- Gewichte der einzelnen Werte müssen mit beachtet werden

| | Magdeburg | | | Ilmenau | | |
|--------------|--------------|--------|------------|--------------|--------|------------|
| | Bierflaschen | Gesamt | Bieranteil | Bierflaschen | Gesamt | Bieranteil |
| 1. Halbwoche | 200 | 210 | 95,2% | 750 | 850 | 88,2% |
| 2. Halbwoche | 510 | 750 | 68,0% | 150 | 280 | 53,6% |
| GesamtWoche | 710 | 960 | 74,0% | 900 | 1130 | 79,6% |

Umsetzung des multidimensionalen Datenmodells

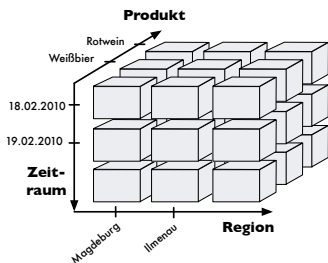
- **Multidimensionale Sicht**
 - ▶ Modellierung der Daten
 - ▶ Anfrageformulierung
- **Interne Verwaltung der Daten erfordert Umsetzung auf**
 - ▶ **Relationale Strukturen (Tabellen)**
 - **ROLAP (relationales OLAP)**
 - ★ Verfügbarkeit, Reife der Systeme
 - ▶ **Multidimensionale Strukturen (direkte Speicherung)**
 - **MOLAP (multidimensionales OLAP)**
 - ★ Wegfall der Transformation bzw. Vorwegberechnung
 - ★ Mehrdimensionale Arrays (Fakten) und assoziierte Dimensionslisten
 - ▶ **Hybride Struktur (Mischform)**
 - **HOLAP (hybrides OLAP)**
 - ★ Detaildaten relational abgespeichert (ROLAP)
 - ★ Aggregate werden multidimensional abgespeichert (MOLAP)
- **Aspekte**
 - ▶ Speicherung
 - ▶ Anfrageformulierung bzw. -ausführung

Relationale Speicherung

- Vermeidung des Verlustes anwendungs-bezogener 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 Datenwürfels ohne Klassifikationshierarchien
 - ▶ Dimensionen, Kennzahlen → Spalten der Relation
 - ▶ Zelle → Tupel

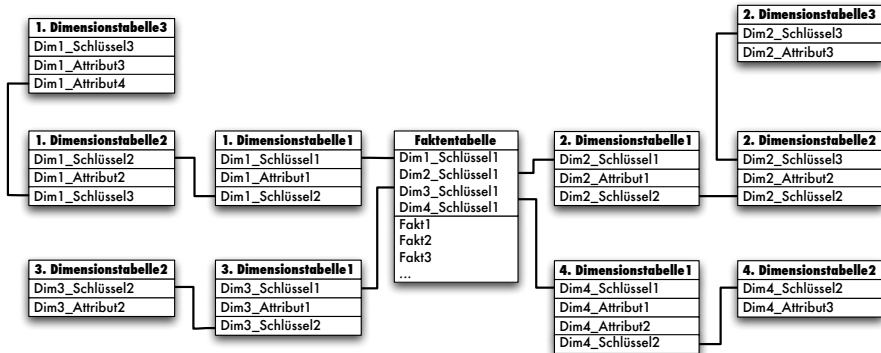


| Produkt | Filiale | Tag | Verk. |
|----------|-----------|----------|-------|
| Rotwein | Magdeburg | 18.02.10 | 145 |
| Weißbier | Magdeburg | 18.02.10 | 267 |
| Rotwein | Ilmenau | 18.02.10 | 70 |
| ... | | | |

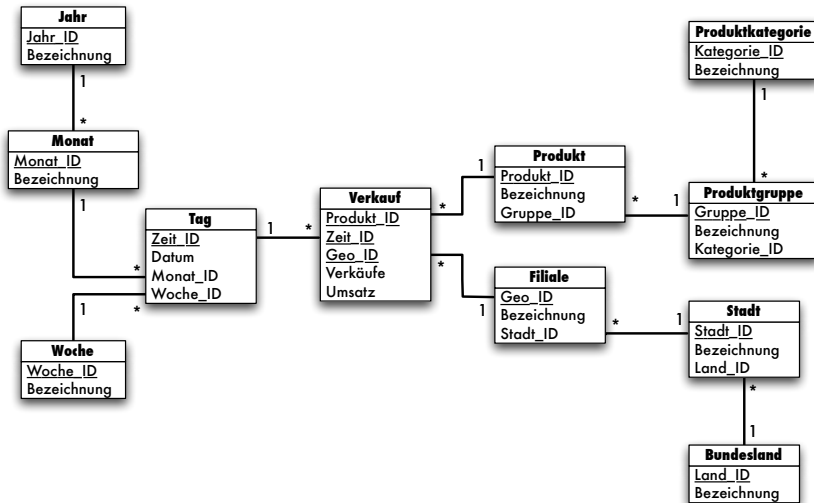
Snowflake-Schema

- Abbildung von Klassifikationen: eigene Tabelle für jede Klassifikationsstufe (z.B. Artikel, Produktgruppe, etc.)
- Dimensionstabelle enthält
 - ▶ ID für Klassifikationsknoten
 - ▶ Beschreibendes Attribut (z.B. Marke, Hersteller, Bezeichnung)
 - ▶ Fremdschlüssel der direkt übergeordneten Klassifikationsstufe
- Faktentabelle enthält (neben Kenngrößen):
 - ▶ Fremdschlüssel der jeweils niedrigsten Klassifikationsstufe
 - ▶ Fremdschlüssel bilden zusammengesetzte Primärschlüssel für Faktentabelle

Snowflake-Schema: Muster



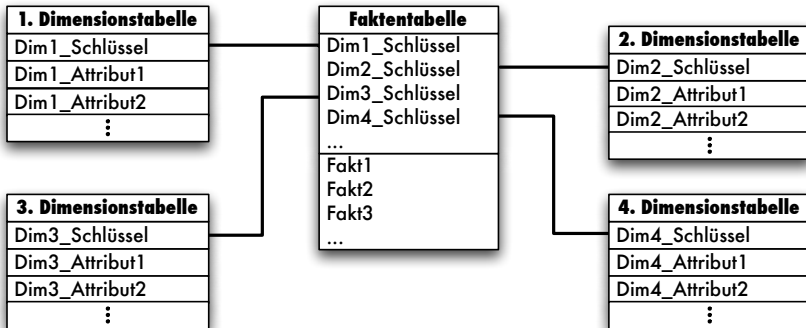
Snowflake-Schema: Beispiel



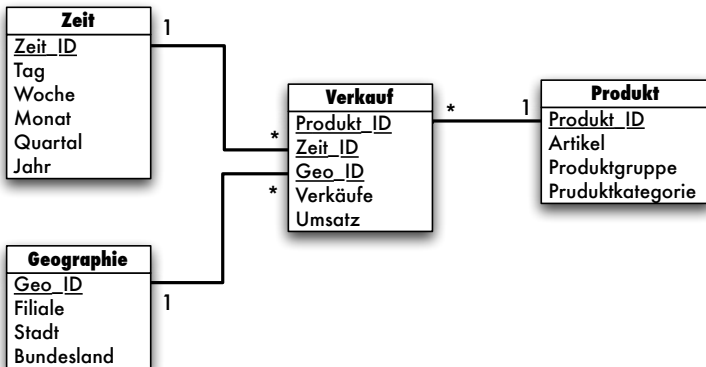
Star-Schema

- Snowflake-Schema ist normalisiert: Vermeidung von Update-Anomalien → **3. NF**
 - ▶ Aber: erfordert Join über mehrere Tabellen!
- Star-Schema:
 - ▶ Denormalisierung der zu einer Dimension gehörenden Tabellen → **1. NF**
 - ▶ Für jede Dimension genau eine Dimensionstabelle
 - ▶ Redundanzen in der Dimensionstabelle für schnellere Anfragebearbeitung
 - ▶ Beispiel: Artikel, Produkt, Produktgruppe etc. als Spalten in einer Tabelle Produkt

Star-Schema: Muster



Star-Schema: Beispiel



Star-Schema formal

- Multidimensionales Schema mit n Dimensionen
 - ▶ Dimensionstabellen D^1, \dots, D^n der Form $D^i(Dim_{i_Key}, A_{i,1}, \dots, A_{i,k_i})$
 - ▶ Faktentabelle $F(Dim_1_Key, \dots, Dim_n_Key, f_1, \dots, f_m)$ mit m Fakten
- Jeder Teil des kompositen Primärschlüssels der Faktentabelle ist Fremdschlüssel zum Primärschlüsselattribut der korrespondierenden Dimension

CREATE DIMENSION in Oracle

- Fremdschlüsselbedingungen in SQL ausdrückbar
- Aber: funktionale Beziehungen zwischen Attributen innerhalb einer Dimension nicht spezifizierbar
- Oracle-Erweiterung: `CREATE DIMENSION`
 - ▶ „informative“ Zusicherung
 - ▶ Korrektheit wird vom DBS nicht überprüft
 - ▶ Nutzung beim Query Rewriting über materialisierten Sichten

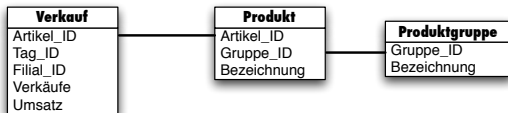
Beispiel CREATE DIMENSION

```
CREATE DIMENSION Bestellung_DIM
  LEVEL Bestellung IS ORDERS.ORDERKEY
  LEVEL Kunde IS ORDERS.CUSTOMERKEY
  LEVEL Region IS ORDERS.NATIONKEY
  LEVEL Gebiet IS ORDERS.REGIONKEY
HIERARCHY Bestellung_Hierarchie (
  Bestellung CHILD OF
  Kunde CHILD OF
  Region CHILD OF
  Gebiet )
ATTRIBUTE Bestellung DETERMINES
  (Bestell_Status, Bestell_Datum, ...)
ATTRIBUTE Kunde DETERMINES
  (Kundenname, Kundenadresse, ...);
```

Schlüsselworte

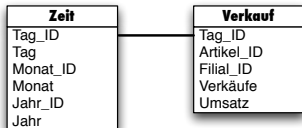
- LEVEL
 - ▶ Definiert Klassifikationsstufen
- HIERARCHY
 - ▶ Festlegung der Abhängigkeiten der Klassifikationsstufen
- ATTRIBUTE . . . DETERMINES
 - ▶ Definiert Abhängigkeit zwischen Klassifikationsattribut und dimensional Attributen

Snowflake mit CREATE DIMENSION



```
CREATE DIMENSION dim_Produkt
  LEVEL Produkt IS (Produkt.Artikel_ID)
  LEVEL Produktgruppe IS (Produktgruppe.Gruppe_ID)
HIERARCHY pg_rollup (
  Produkt CHILD OF Produktgruppe)
JOIN KEY (Produkt.Gruppe_ID)
REFERENCES Produktgruppe.Gruppe_ID
```

Star mit CREATE DIMENSION



```

CREATE DIMENSION dim_Zeit
  LEVEL Tag IS (Zeit.Tag_ID)
  LEVEL Monat IS (Zeit.Monat_ID)
  LEVEL Jahr IS (Zeit.Jahr_ID)
HIERARCHY zeit_rollup (
  Tag CHILD OF Monat CHILD OF Jahr)
ATTRIBUTE Tag_ID DETERMINES (Tag)
ATTRIBUTE Monat_ID DETERMINES (Monat)
ATTRIBUTE Jahr_ID DETERMINES (Jahr)
  
```

Vergleich Star und Snowflake

- Charakteristika von DW-Anwendungen
 - ▶ Typischerweise Einschränkungen in Anfragen auf höherer Granularitätsstufe (Join-Operationen)
 - ▶ Geringes Datenvolumen der Dimensionstabellen im Vergleich zu Faktentabellen
 - ▶ Seltene Änderungen an Klassifikationen (Gefahr von Update-Anomalien)
- Vorteile des Star-Schemas
 - ▶ Einfache Struktur (vereinfachte Anfrageformulierung)
 - ▶ Einfache und flexible Darstellung von Klassifikationshierarchien (Spalten in Dimensionstabellen)
 - ▶ Effiziente Anfrageverarbeitung innerhalb einer Dimension (keine Join-Operation notwendig)

Annahmen für Kostenbetrachtungen

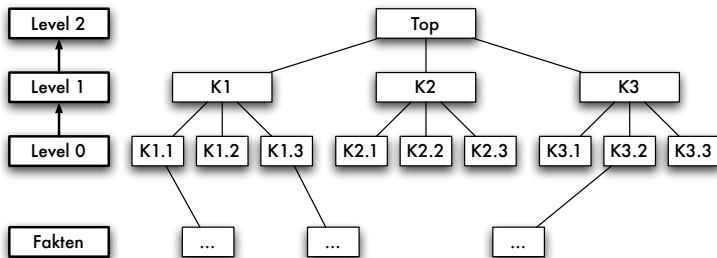
- n Dimensionen (D^n), je K Klassifikationsstufen plus Top
- Jeder Klassifikationsknoten hat 3 Kinder
 - ▶ Level $i = K$: $1 = 3^0$ Knoten (Top)
 - ▶ Level $i = K - 1$: $3 = 3^1$ Knoten
 - ▶ Level $i = K - 2$: $9 = 3^2$ Knoten
 - ▶ ...
 - ▶ Level $i = 0$: Höchste Granularität, 3^K Knoten
 - ▶ $N_D = \sum_{i=0 \dots K} 3^i$ Knoten pro Dimension

$$N_D = \sum_{i=0 \dots K} 3^i$$

- M Fakten, gleichverteilt in Dimensionen
- Attribut: b Bytes; Knoten haben nur ID; m Faktattribute

Volle Klassifikationshierarchie

- Zu jedem Knoten gibt es (gleich viele) Fakten



Kosten Star vs. Snowflake

- Speicherplatz Snowflake:

$$(((n + m) \cdot M) + n \cdot N_D) \cdot b$$

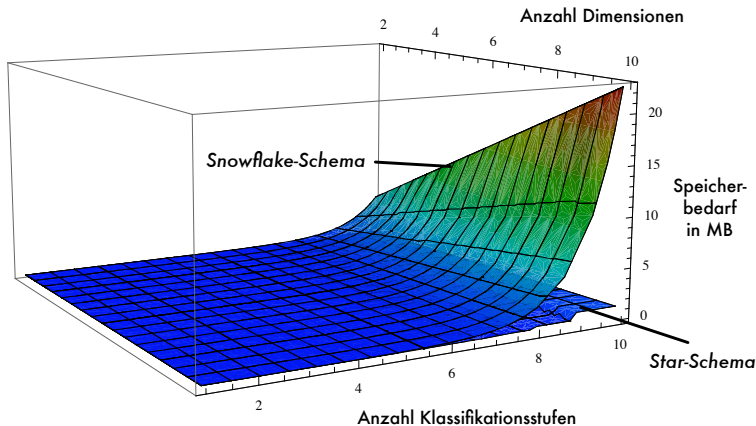
- ▶ n Anzahl Fremdschlüssel in Faktentabelle
- ▶ m Anzahl Faktenattribute
- ▶ $n \cdot N_D$ Ein Tupel pro Klassifikationsknoten

- Speicherplatz Star:

$$(((n + m) \cdot M) + n \cdot 3^K \cdot K) \cdot b$$

- ▶ $n \cdot 3^K$ Ein Tupel pro Klassifikationsknoten Level 0
- ▶ K Ein Attribut pro Klassifikationsstufe

Kosten Star vs. Snowflake



Anfrage Star und Snowflake

- Anfrage: Verkäufe der Produktgruppe „Soft-Drink“ pro Filiale und Jahr
- Snowflake-Schema:

```
SELECT F.Bezeichnung, J.Jahr, SUM(Verkäufe)
FROM Verkauf V, Filiale F, Artikel A,
      Produktgruppe PG, Tag T, Monat M, Jahr J
WHERE V.Artikel_ID = A.Artikel_ID AND
      A.Gruppe_ID = PG.Gruppe_ID AND
      PG.Bezeichnung = 'Soft-Drink' AND
      V.Tag_ID = T.Tag_ID AND
      T.Monat_ID = M.Monat_ID AND
      M.Jahr_ID = J.Jahr_ID AND
      V.Filial_ID = F.Filial_ID
GROUP BY F.Bezeichnung, J.Jahr
```

- Anzahl der Joins: 6
(steigt linear mit Anzahl der Aggregationspfade)

Anfrage Star und Snowflake (2)

- Anfrage für Star-Schema:

```
SELECT G.Filiale, Z.Jahr, SUM(Verkäufe)
FROM Verkauf V, Geographie G, Produkt P,
      Zeit Z
WHERE V.Produkt_ID = P.Produkt_ID AND
V.Zeit_ID = Z.Zeit_ID AND
V.Geographie_ID = G.Geographie_ID AND
P.Produktgruppe = 'Soft-Drink'
GROUP BY G.Filiale, Z.Jahr
```

- Anzahl der Joins: 3
(unabhängig von der Länge der Aggregationspfade)

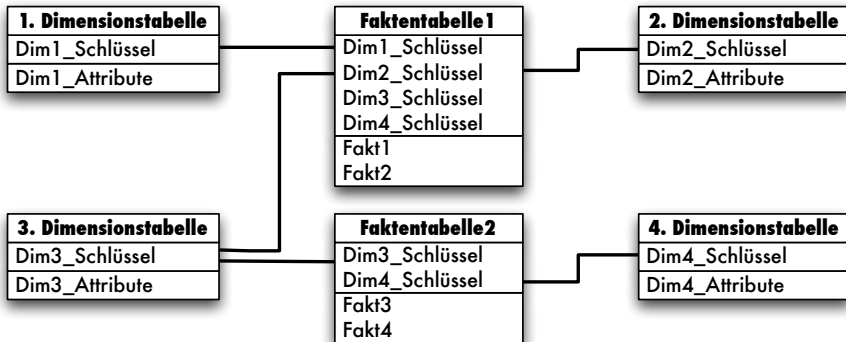
Mischformen

- Abbildung einzelner Dimensionen analog Snowflake-Schema oder Star-Schema
- Entscheidungskriterien:
 - ▶ Änderungshäufigkeit der Dimensionen:
 - ★ Reduzierung des Pflegeaufwandes durch Normalisierung (Snowflake)
 - ▶ Anzahl der Klassifikationsstufen einer Dimension:
 - ★ Mehr Klassifikationsstufen → größere Redundanz im Star-Schema
 - ▶ Anzahl der Dimensionselemente:
 - ★ Einsparung durch Normalisierung bei vielen Elementen einer Dimension auf niedrigster Klassifikationsstufe
 - ▶ Materialisierung von Aggregaten:
 - ★ Performance-Verbesserung durch Normalisierung bei materialisierten Aggregaten für eine Klassifikationsstufe

Galaxie-Schema

- Star- und Snowflakeschema
 - ▶ Eine Faktentabelle
 - ▶ Mehrere Kennzahlen nur möglich bei gleichen Dimensionen
- Galaxie-Schema
 - ▶ Mehrere Faktentabellen
 - ▶ Teilweise mit gleichen Dimensionstabellen verknüpft
 - ▶ Auch: Multi-Faktentabellen-Schema, Multi-Cube, Hyper-Cube

Galaxie-Schema: Muster



Fact Constellation

- Speicherung vorberechneter Aggregate in Faktentabelle
 - ▶ Beispiel: Umsatz für Region
 - ▶ Unterscheidung in Dimensionstabelle über spezielle Attribute (Bsp.: „Stufe“)
- Alternative: Auslagerung in eigene Faktentabelle
 - ▶ Fact-Constellation-Schema (Spezialfall eines Galaxie-Schemas)

Darstellung von Klassifikationshierarchien

- Horizontal: Modellierung der Stufen der Klassifikationshierarchie als Spalten der denormalisierten Dimensionstabelle
 - ▶ Vorteil:
 - ★ Einschränkungen auf höherer Granularität ohne Join
 - ▶ Nachteile:
 - ★ Duplikateliminierung beim Anfragen bestimmter Stufen (Bsp.: Produktgruppe innerhalb einer Kategorie)
 - ★ Schemaänderung beim Hinzufügen neuer Stufen

| Produkt_ID | Artikel | Produktgruppe | Produktkategorie |
|------------|-------------|---------------|------------------|
| 1234 | Immer Ultra | Hygiene | Kosmetik |
| 1235 | Putzich | Hygiene | Kosmetik |
| 2345 | Rohrfrei | Reiniger | Haushalt |

Darstellung von Klassifikationshierarchien

- Vertikal (rekursiv): normalisierte Dimensionstabelle mit Attributen
 - ▶ Dimensions_ID: Schlüssel für Faktentabelle
 - ▶ Eltern_ID: Attributwert der Dimensions-ID der nächsthöheren Stufe
- Vorteile:
 - ▶ Einfache Änderung am Klassifikationsschema
 - ▶ Einfache Behandlung vorberechneter Aggregate
- Nachteil:
 - ▶ Self-Join für Anfragen einzelner Stufen (Bsp.: Produktgruppe innerhalb einer Kategorie)

| Dimensions_ID | Eltern_ID |
|---------------|-----------|
| Immer Ultra | Hygiene |
| Hygiene | Kosmetik |
| Putzich | Hygiene |

Darstellung von Klassifikationshierarchien

- Kombiniert: Verbindung beider Strategien
 - ▶ Repräsentation der Klassifikationsstufen als Spalten (jedoch generische Bezeichnung)
 - ▶ Speicherung der Knoten aller höheren Stufen als Tupel
 - ▶ Zusätzliches Attribut „Stufe“ → Angabe der bezeichneten Klassifikationsstufe

| Dimensions_ID | Stufe1_ID | Stufe2_ID | Stufe |
|---------------|-----------|-----------|-------|
| Immer Ultra | Hygiene | Kosmetik | 0 |
| Putzich | Hygiene | Kosmetik | 0 |
| Hygiene | Kosmetik | NULL | 1 |
| Kosmetik | NULL | NULL | 2 |

Vermeidung von Semantikverlusten

- Semantikverlust bei relationaler Abbildung:
 - ▶ Unterscheidung zwischen Kennzahl und Dimension (Attribute der Faktentabelle)
 - ▶ Attribute von Dimensionstabellen (beschreibend, Aufbau der Hierarchie)
 - ▶ Aufbau der Dimensionen (Drill-Pfade)
- Ausweg:
 - ▶ Erweiterung des Systemkatalogs um Metadaten für multidimensionale Anwendungen
 - ▶ Beispiel: CREATE DIMENSION, HIERARCHY in Oracle

Probleme der relationalen Umsetzung

- Transformation multidimensionaler Anfragen in relationale Repräsentation notwendig → komplexe Anfragen
- Einsatz komplexer Anfragewerkzeuge notwendig (OLAP-Werkzeuge)
- Semantikverlust
- Daher: direkte multidimensionale Speicherung → siehe Teil VI