

**Seminar: Business Intelligence –  
Teil I: OLAP & Data Warehousing**

# Data-Warehouse-Design

Jörg Ramser

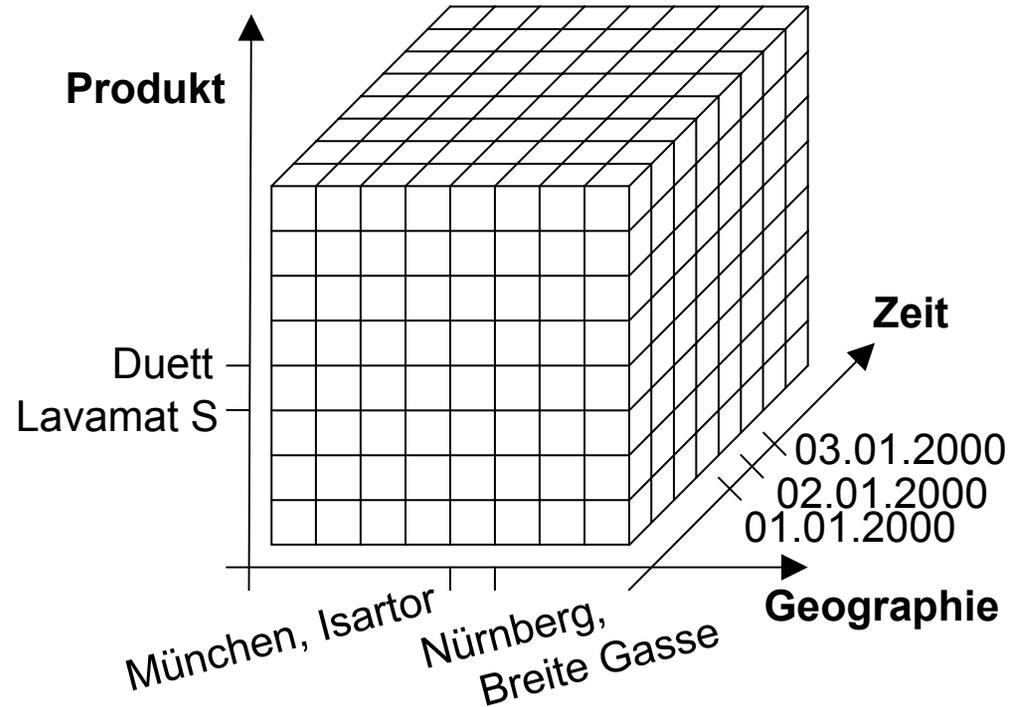
18.07.2003

# Übersicht

- Umsetzung des multidimensionalen Datenmodells
  - Relationale Speicherung (ROLAP)
  - Multidimensionale Speicherung (MOLAP)
  - Hybride Lösung (HOLAP)
  
- Metadaten
  - Rolle der Metadaten
  - Anforderungen an Repositorien
  - Standards
  
- Zusammenfassung

# ROLAP: Faktentabelle

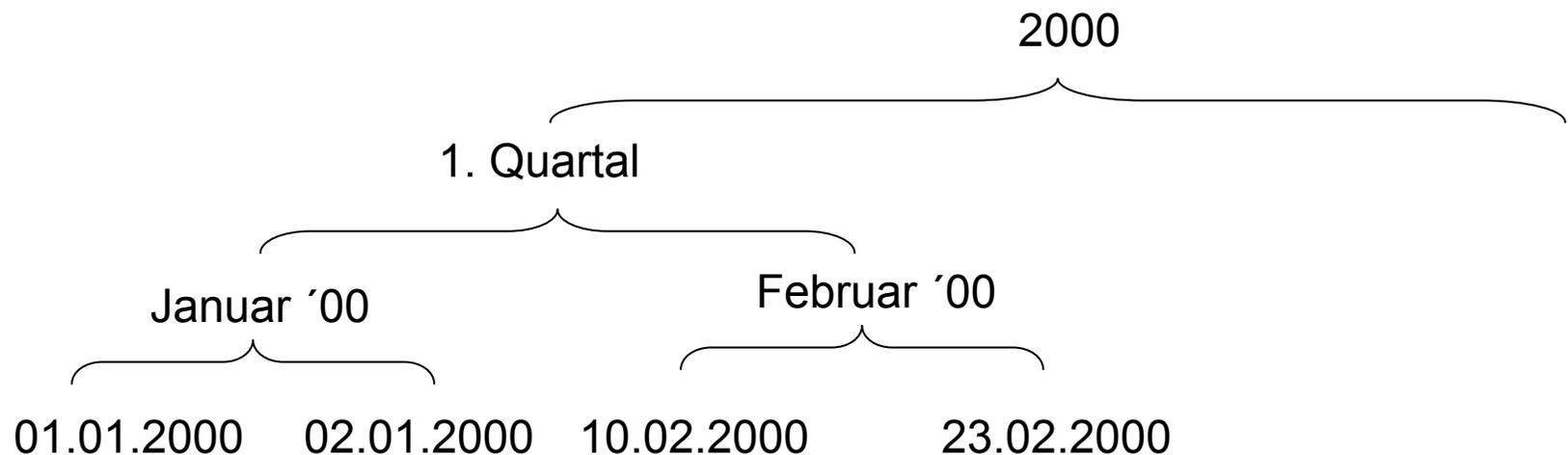
- Ohne Klassifikationshierarchien
- Interpretation der Spalten als
  - Dimension bzw.
  - Kenngröße



Produkt (Dimension)	Geographie (Dimension)	Zeit (Dimension)	Verkäufe (Kenngröße)	Preis (Kenngröße)
Duett	Nürnberg, Breite Gasse	03.01.2000	2	800
Duett	München, Isartor	02.01.2000	3	1200
Lavamat S	München, Isartor	01.01.2000	2	1500

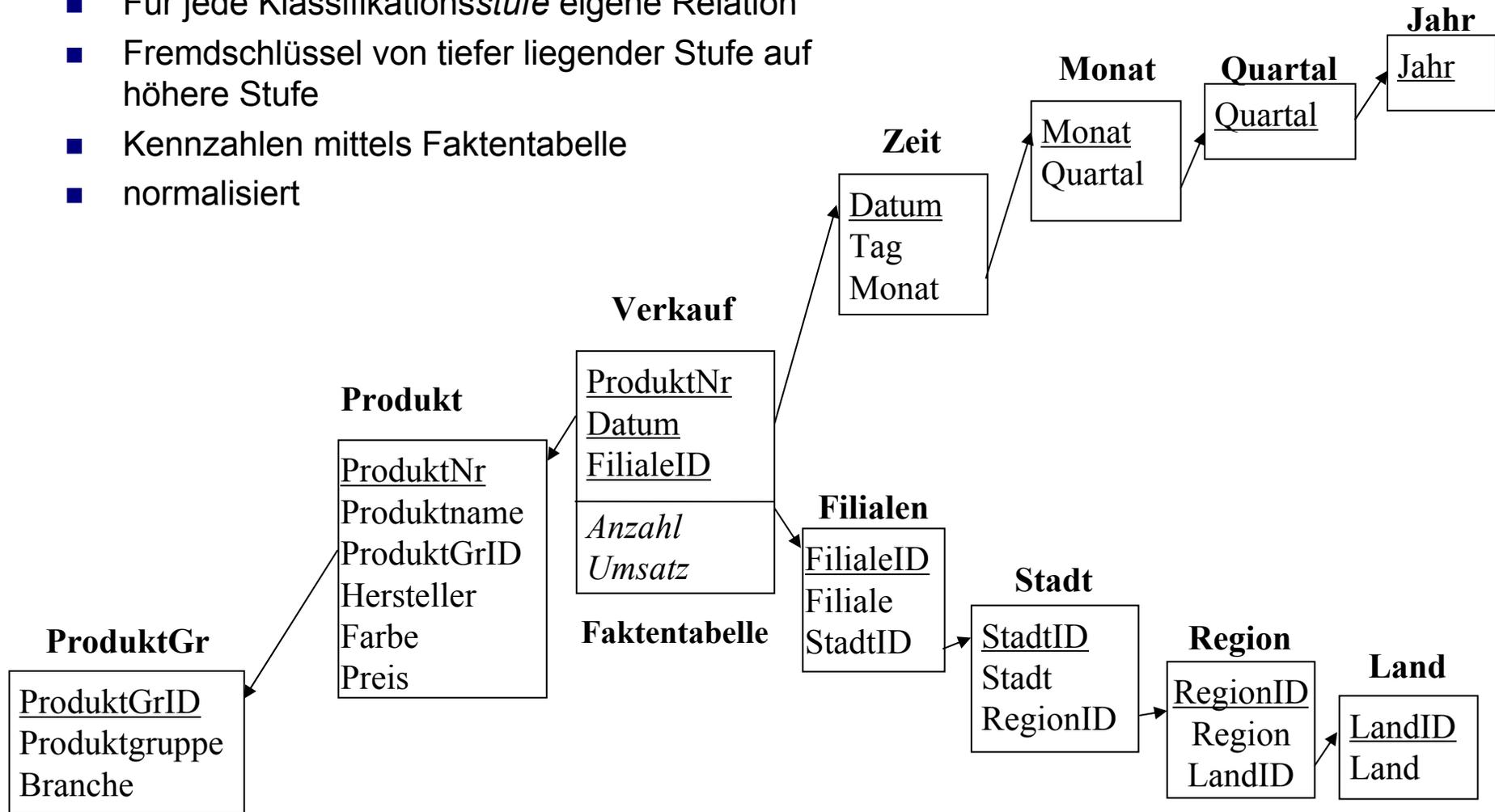
# Klassifikationshierarchien

- Snowflake-Schema
- Star-Schema
- Mischformen von Star- und Snowflake-Schema
- Galaxie



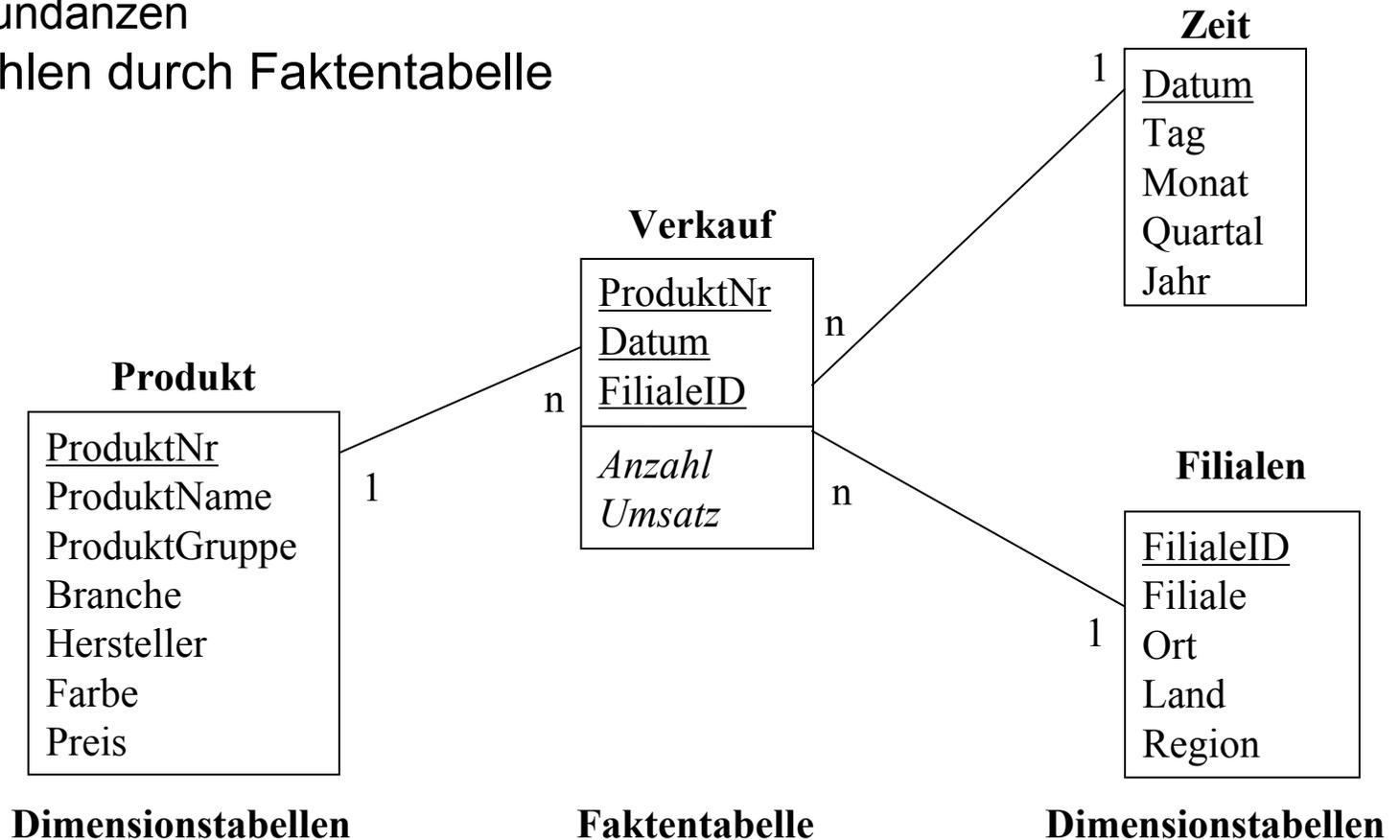
# Snowflake-Schema

- Für jede Klassifikationsstufe eigene Relation
- Fremdschlüssel von tiefer liegender Stufe auf höhere Stufe
- Kennzahlen mittels Faktentabelle
- normalisiert



# Star-Schema

- Denormalisierung von Snowflake-Schema
  - Pro Dimension nur noch eine Tabelle
  - Redundanzen
- Kennzahlen durch Faktentabelle



# Star-Schema

## ■ Vorteile

- Weniger teure Verbundoperationen
- Star-Schema besitzt einfachere Struktur

## ■ Nachteile

- Redundanzen → Änderungsanomalien

## ■ Trotz Redundanzen häufig Star-Schema, da

- Redundanzen nur in Dimensionstabellen
- Änderungen an Dimensionstabellen selten

# Varianten

- Mischformen von Star- und Snowflake-Schema
  - Vereinigung der Vorteile beider Schemata
  - Snowflake-Schema, falls
    - Frequenz der Änderungen *hoch*
    - *Große* Anzahl von Dimensionselementen auf niedrigster Stufe
    - Anzahl der Stufen innerhalb einer Dimension *hoch*
    - Innerhalb einer Dimension *viele* Aggregate materialisiert
  
- Galaxien bzw. Multi-Faktentabelle

# Versionisierungs-/Evolutionaspekte

- Klassifikationshierarchieänderungen
  - „Update in place“
  - Tupelversionierung
  - Zeitattribute
- Schemaänderungen
  - Schemaevolution
  - Schemaversionierung

# Tupelversionierung – Beispiel

## Produkt

ANR_VNR	Artikel	Produktgruppe	Produktfamilie
1235-001	Quickphone 150	Singleband	Mobiltelefon
1235-002	Quickphone 150	Dualband	Mobiltelefon
1237-001	Quickphone 100	Singleband	Mobiltelefon
1239-002	Quickphone 200	Dualband	Mobiltelefon
...	...	...	...

## Faktentabelle

ANR_VNR	Filiale_ID	Datum	Verkäufe	Preis
1235-001	50015	02.03.2003	60	299,00
1237-001	50015	02.03.2003	31	599,00
1235-002	50015	05.03.2003	50	199,00
1235-002	50015	06.03.2003	53	199,00
1239-002	50015	07.03.2003	35	99,00
...	...	...	...	...

# Zeitbehaftete Schemabeschreibung

## Beispiel

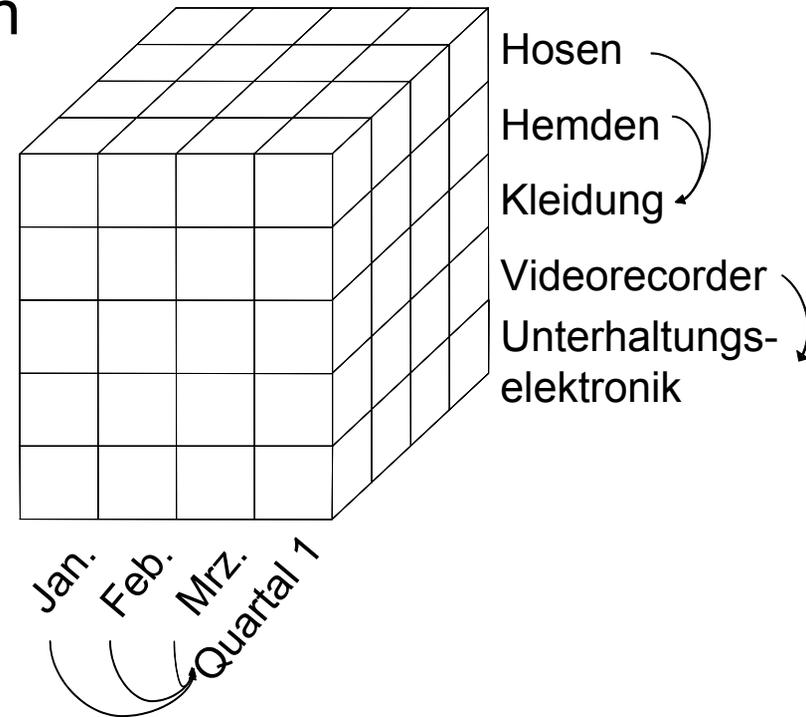
Attribut(AttributDefName, RelationDefName,	$G_A,$	$G_E)$
„Artikel“ „Produkt“	1999-07-01,	$\infty$
„Marke“ „Produkt“	1999-07-01,	1999-12-31
„Produktgruppe“ „Produkt“	1999-08-01,	$\infty$
„Produktfamilie“ „Produkt“	1999-08-01,	$\infty$
„Artikel_ID“ „Produkt“	1999-08-01,	$\infty$

Relation (RelationDefName,	$G_A,$	$G_E)$
„Produkt“	1999-07-01,	$\infty$
„Verkäufe“	1999-07-01,	$\infty$

# Multidimensionale Speicherung

## ■ Speicherung der Datenstrukturen

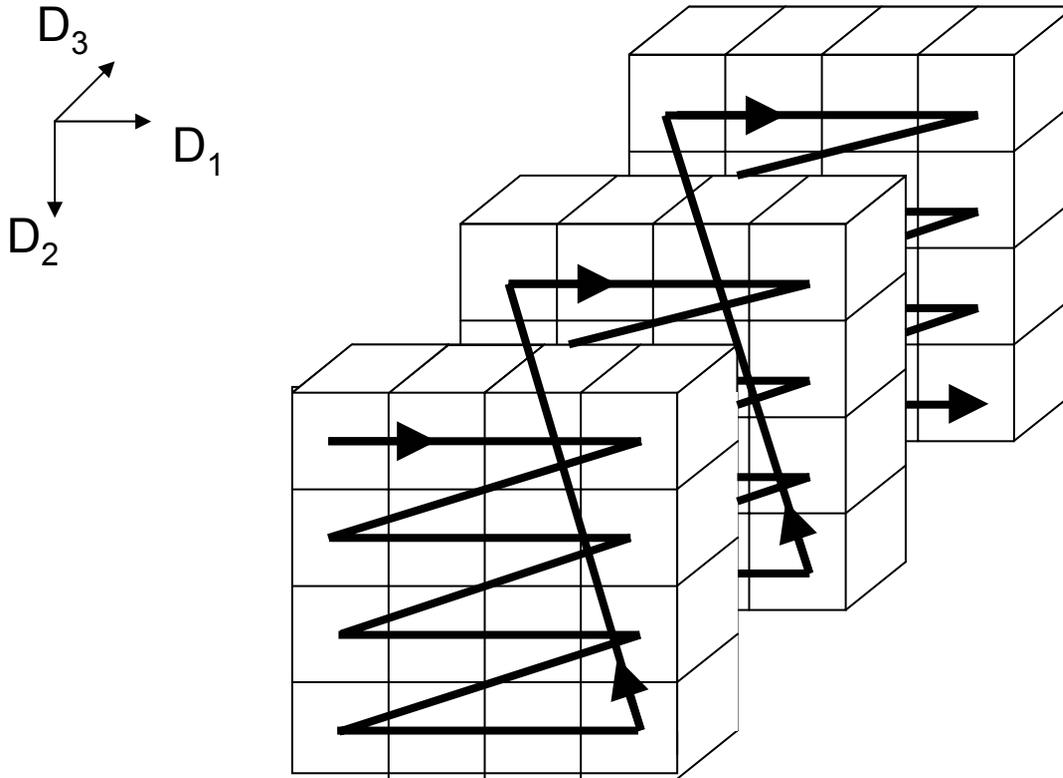
- Dimension (inklusive Hierarchien)
- Würfel



⇒ Speicherung in multidimensionaler Matrix

⇒ Problem: Speicherung nur linear möglich

# Linearisierungsreihenfolge



$$\text{Index}(z) = x_1 + (x_2 - 1) \cdot |D_1| + (x_3 - 1) \cdot |D_1| \cdot |D_2| + \dots + (x_n - 1) \cdot |D_1| \cdot \dots \cdot |D_{n-1}|$$

$$= 1 + \sum_{i=1}^n (x_i - 1) \cdot \prod_{j=1}^{i-1} |D_j| \quad , \text{ wobei } z = (x_1, x_2, \dots, x_n)$$

# relational vs multidimensional

	Relational	Multidimensional
<b>Vorteile</b>	<ul style="list-style-type: none"><li>■ Bewährte Technologie</li><li>■ Standard SQL</li><li>■ Datenimport einfach</li><li>■ Vorhandene Sicherheitsmechanismen</li><li>■ Verarbeitung großer Datenmengen</li><li>■ Leichte Skalierbarkeit</li></ul>	<ul style="list-style-type: none"><li>■ Hohe Anfragegeschwindigkeit</li><li>■ Effiziente multidimensionale Speicherungsstrukturen</li><li>■ Multidimensionale Anfragesprache</li></ul>
<b>Nachteile</b>	<ul style="list-style-type: none"><li>■ Standard SQL nur bedingt ausreichend</li><li>■ Semantikverluste</li><li>■ Performanzverluste bei Abbildung</li></ul>	<ul style="list-style-type: none"><li>■ Dünnbesetzte Würfel</li><li>■ Kein Anfragesprachenstandard</li><li>■ Skalierbarkeit eingeschränkt</li></ul>

# Hybride Speicherung

- Ziel: Vereinigung der Vorteile beider Welten



- Detaildaten relational speichern
- Aggregierte Daten multidimensional speichern
- Forderung: Zugriffstransparenz für Benutzer

# Übersicht

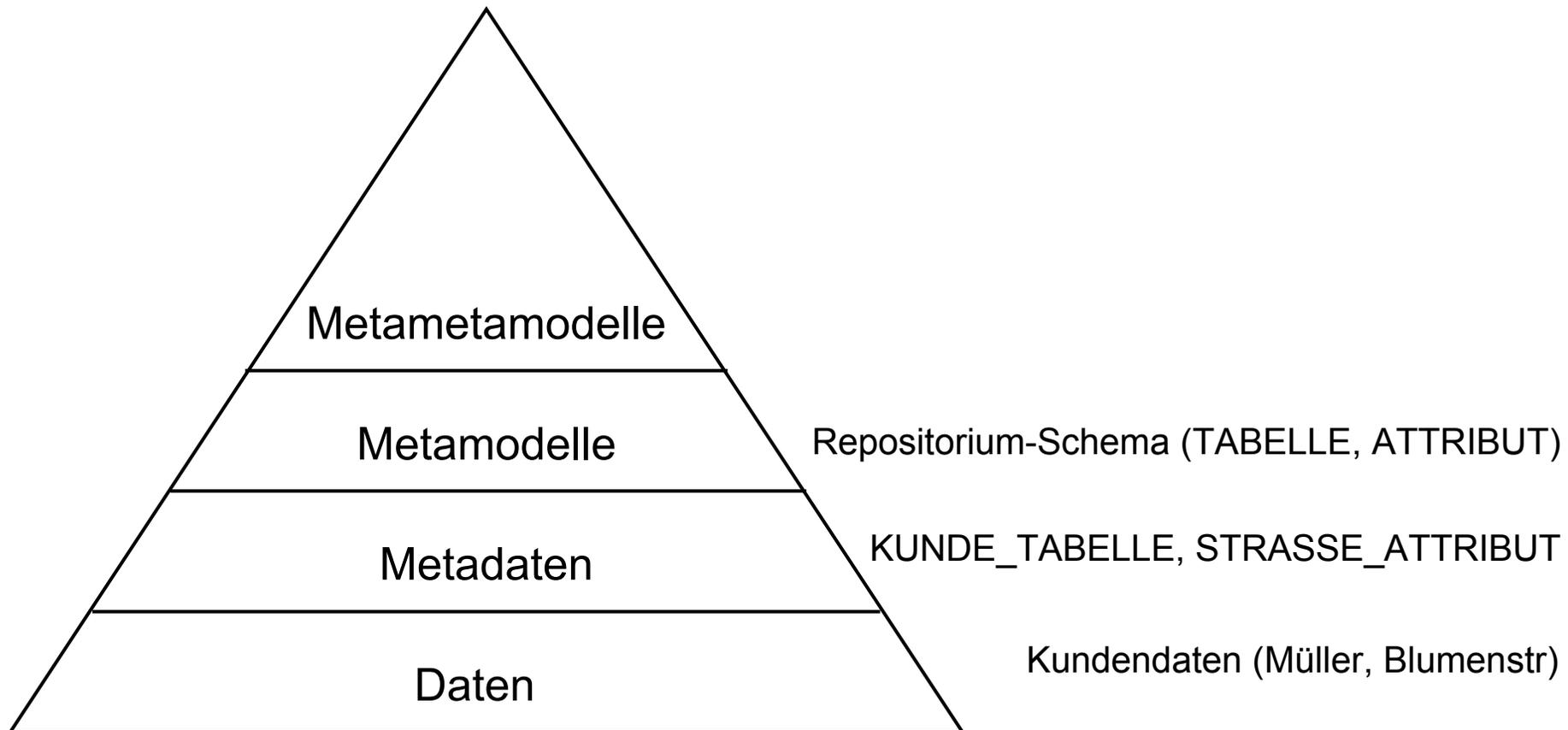
- Umsetzung des multidimensionalen Datenmodells
  - Relationale Speicherung
  - Multidimensionale Speicherung
  - Hybride Lösung
- Metadaten
  - Rolle der Metadaten
  - Anforderungen an Repositorien
  - Standards
- Zusammenfassung

# Rolle der Metadaten

- Haltung der Metadaten im Metadaten-Warehouse bzw. Repositoryum
  
- Zweck der Metadaten
  - Effektive Beschaffung von Informationen
    - Datenqualität
    - Terminologie
    - Datenanalyse
  
  - Unterstützung des laufenden Betriebs
    - Automatisierung der Administrationsprozesse
    - Systemintegration
    - Schutz und Sicherheitsaspekte
    - Flexibler Softwareentwurf

# Modellierung Metadaten

- Mindestens 4 Ebenen zur Modellierung komplexer Informationssysteme



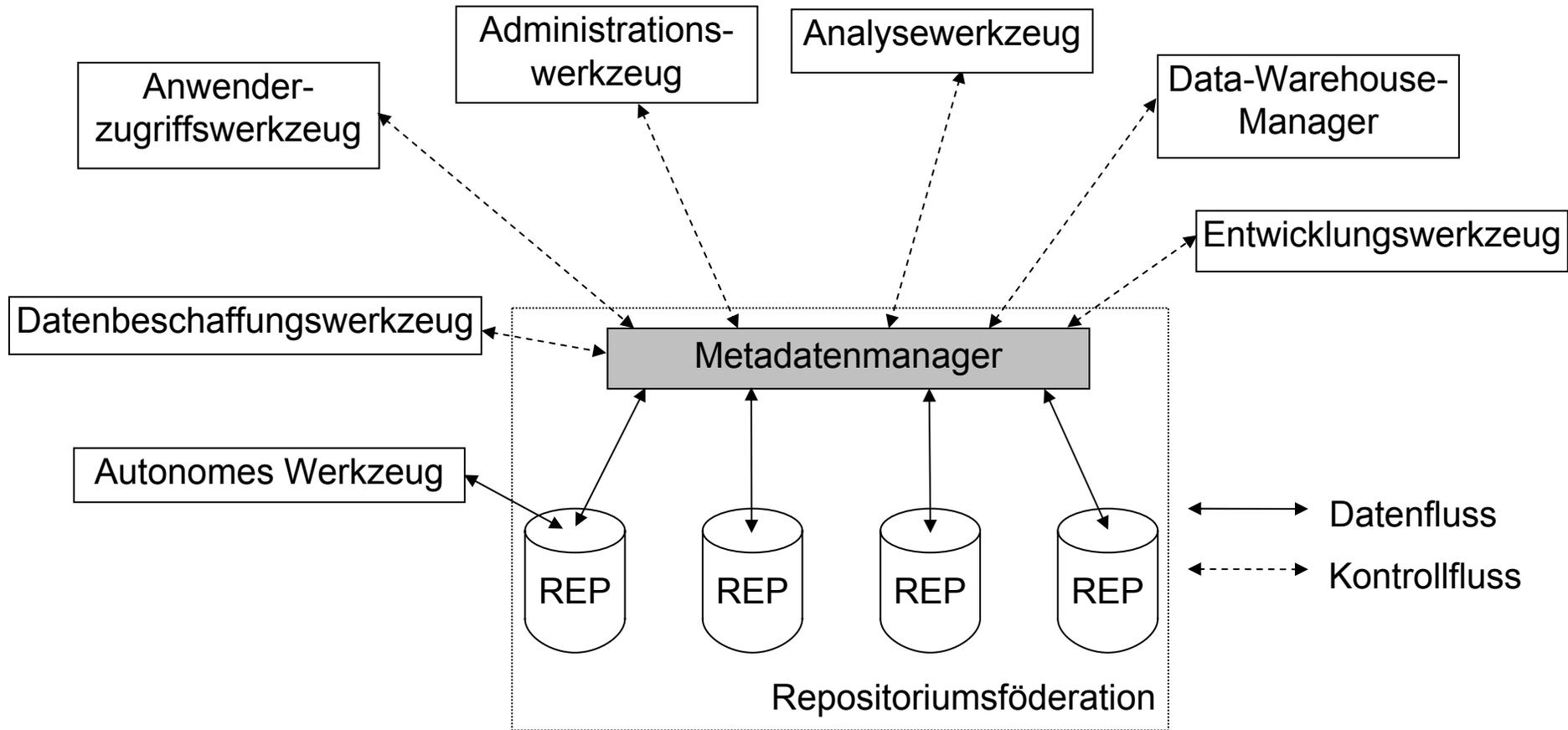
# Funktionalitäten der Repositorien

- Anwenderzugriff
  - Navigation
  - Selektion
  - Filterung
  - Manuelle Aktualisierung
  
- Interoperabilität und Werkzeugunterstützung
  - Austauschformat
  - Programmierschnittstelle (API)
  - Erweiterbares Metamodell
  
- Changemanagement
  - Versions- und
  - Konfigurationsverwaltung
  - Notifikationsdienst
  - Auswirkungsanalyse

# Realisierung Metadatenverwaltung

- **Zentrale Metadatenverwaltung**
  - Zentral und konsistent verwaltet
- **Dezentrale Metadatenverwaltung**
  - Repositorien komplett unabhängig
  - Zusammenarbeit durch Austausch
- **Föderierte Metadatenverwaltung**
  - Mischung aus zentraler und dezentraler Verwaltung
  - Globale, konzeptionelle Sicht auf Metadaten

# Föderierte Metadatenverwaltung



# Standards

## ■ Repositoriumstandards

- Information Resource Dictionary System (IRDS)
- Portable Common Tool Environment (PCTE)

## ■ Austauschstandards

- XML-basiert
- Case Data Interchange Format (CDIF)

## ■ Standard Metamodelle

- CWM (Common Warehouse Metamodel)
- OIM (Open Information Metamodel)
- Zachman Framework

# CWM

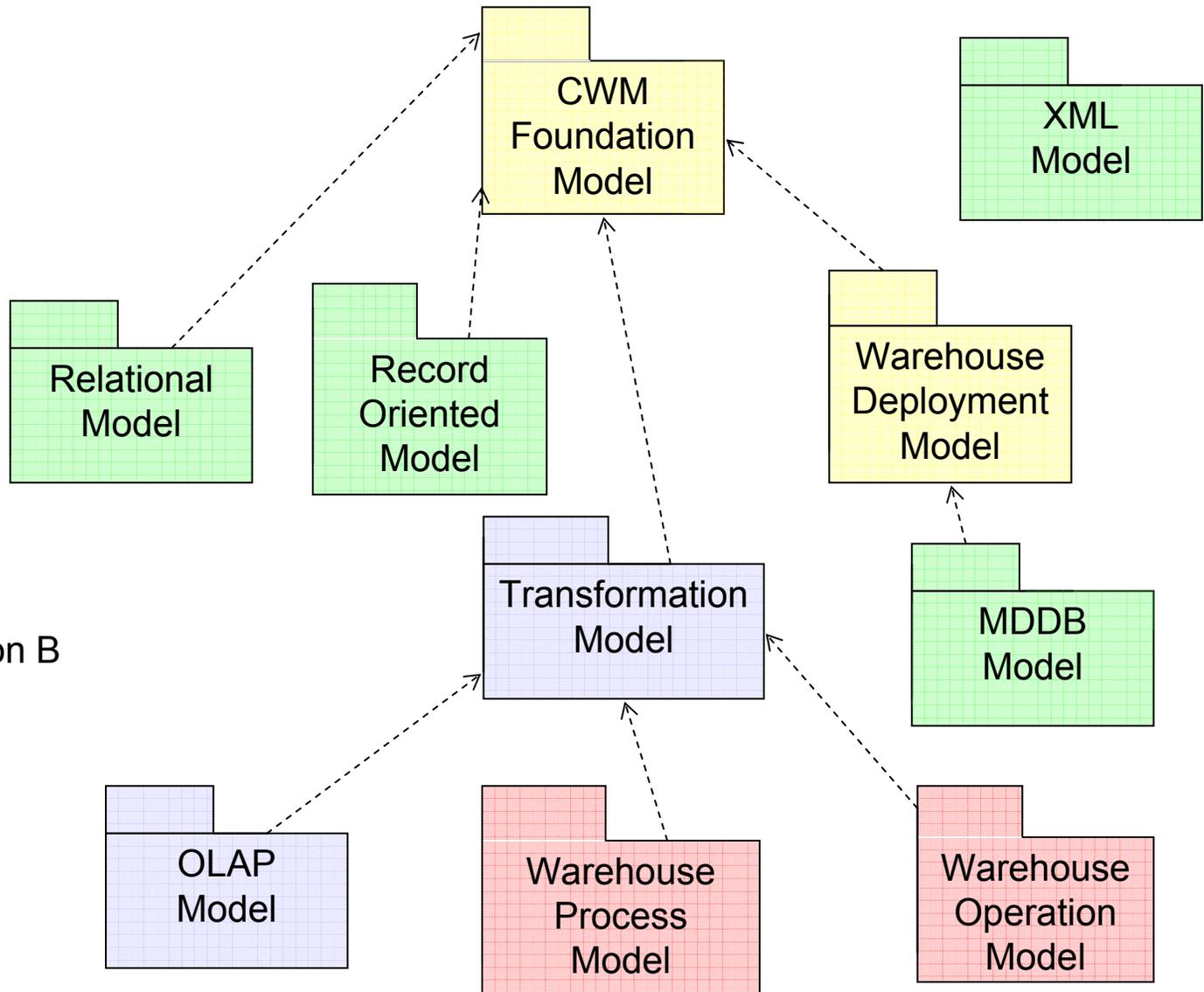
Foundation

Resource

Analysis

Management

A----->B  
A ist abhängig von B



# Zusammenfassung

- Umsetzung des multidimensionalen Datenmodells
  - Relationale Speicherung
  - Multidimensionale Speicherung
  - Hybride Lösung
  
- Metadaten
  - Rolle der Metadaten
  - Anforderungen an Repositorien
  - Standards