ISDA 08
Data Warehouses

Prof. Dr. Volker Markl

Folienmaterial von Prof. Dr. Felix Naumann

Fachgebiet Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

http://www.dima.tu-berlin.de/

Rückblick

- Transaktionen
- Isolationsebenen
- Serialisierbarkeit
- Konfliktserialisierbarkeit
- Sperrprotokolle
- Sperren

Überblick

- **Einsatzgebiete**
- OLAP versus OLTP
- Multidimensionale Modellierung
- OLAP Operationen
- Relationale Implementierung

Kapitel 10.6 und 10.7 im Lehrbuch

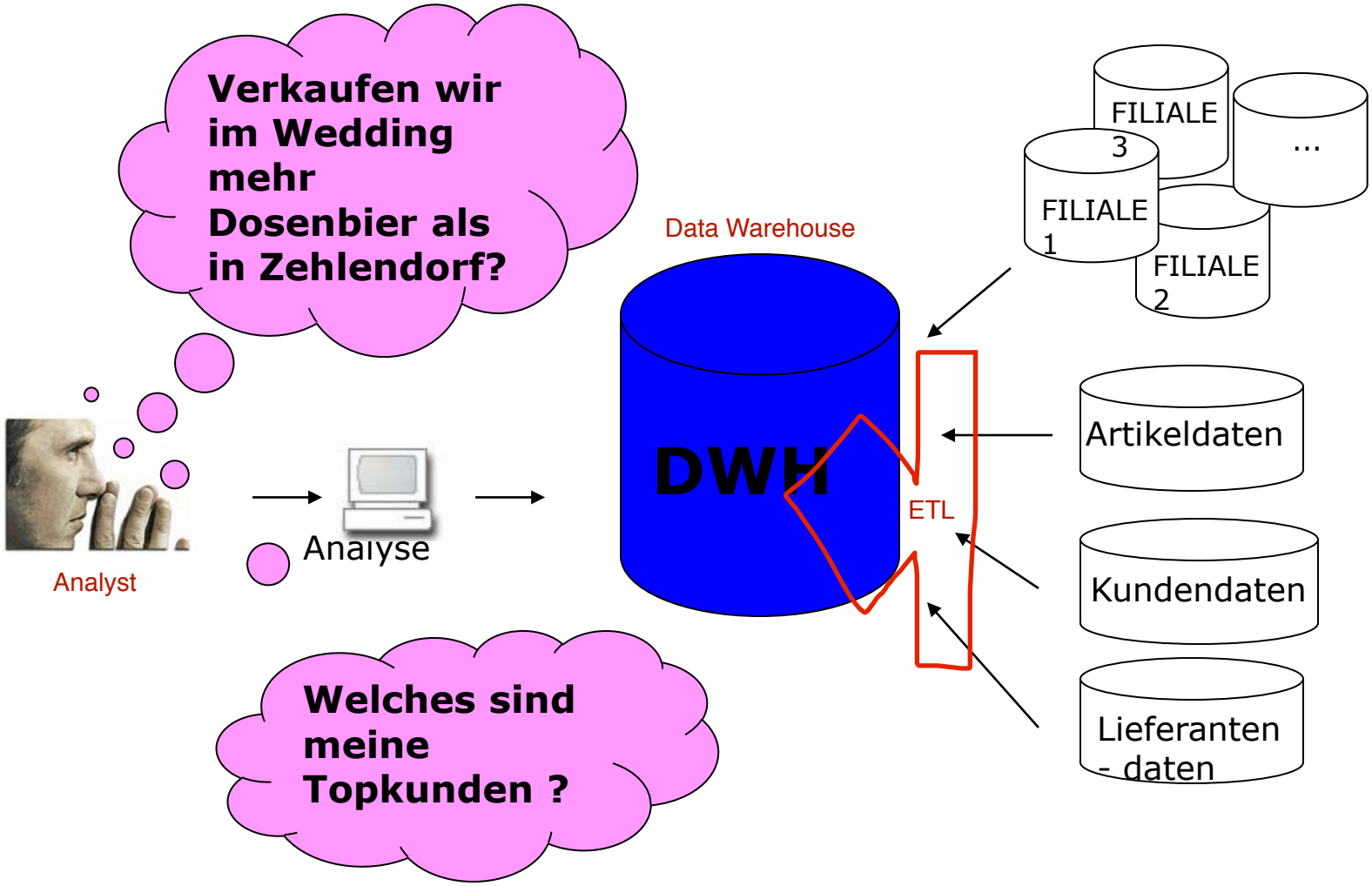
Beispielszenario

OLAP-Beispiel

- Ein beliebiges Handelshaus: Spar, Kaufland, ...
- Physikalische Datenverteilung
  - Viele Niederlassungen (bis zu mehrere tausend)
  - Noch mehr Registerkassen
- Aber: Zentrale Planung, Beschaffung, Verteilung
  - Was wird wo und wie oft verkauft?
  - Was muss wann wohin geliefert werden?
    - Bedenke: Verderbliche Waren
- ... nur möglich, wenn
  - Zentrale Übersicht über Umsätze
  - Integration mit Lieferanten / Produktdaten

weiteres Beispiel: TCP-H Benchmark aus der letzten Hausaufgabe (Test für Datenbanksysteme: Wie schnell sind Anfragen, getestet anhand eines Online-Händlers)
-> analytische Anfragen an Daten

Handels - DWH


Verkaufen wir im Wedding mehr Dosenbier als in Zehlendorf?
Analyst
Analyse
Data Warehouse
DWH
ETL
FILIALE 3
FILIALE 1
FILIALE 2
...
Artikeldaten
Kundendaten
Lieferanten - daten
Welches sind meine Topkunden ?

DWH Datenquellen

- Lieferantendatenbanken
  - Produktinformationen: Packungsgrößen, Farben, ...
  - Lieferbedingungen, Rabatte, Lieferzeiten, ...
- Personaldatenbank
  - Zuordnung Kassenbuchung auf Mitarbeiter
  - Stundenabrechnung, Prämien
- Kundendatenbank
  - Kundenklassen: Premium, normal, soziale Brennpunkte, ...
  - Persönliche Vorlieben & Historie
    - Kundenkarten (Safeway, …)
- Weitere Vertriebswege
  - Internet, Katalogbestellung, Verkaufsclubs, ...

Definition DWH

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

Daten aus mehreren verschiedenen Systemen

Daten werden nicht verändert

ETL
ETL Prozess vor der Integration neuer Daten ins Data Warehouse:
E - Extraktion: Daten aus Quellsystem exportieren
T - Transformation: Daten ins Format des Data Warehouse umwandeln
L - Laden: transformierte Daten werden ins Data Warehouse importiert

Transformationen von Daten:
Selektion relevanter Spalten, Anpassung von Format & Datentypen, Normalisierung, De-Duplikation, Vor-Aggregation, Datenvalidierung…

Erfolg von DWH

- Top-Thema seit Mitte der 90er Jahre
  - Industrie schneller als Forschung

Große Datenbestände in der Industrie vorhanden
-> Interesse an Analyse dieser Daten

- Voraussetzungen

wachsender Markt bis heute

  - Extreme Verbilligung von Plattenspeicherplatz
  - Relationale Modellierung: Anwendungsneutral
  - Graphische Benutzeroberflächen und Terminals
  - IT in allen Unternehmensbereichen (SAP R/3)
  - Vernetzung und DB Standardisierung (SQL)
- Aber
  - Vision der vollständigen Integration scheitert (immer wieder aufs neue)
  - Soziale versus technische Aspekte

Akteure: IBM, Oracle, SAP + kleinere

Überblick

- Einsatzgebiete
- **OLAP versus OLTP**
- Multidimensionale Modellierung
- OLAP Operationen
- Relationale Implementierung

OLTP Beispiel

Login

- `SELECT pw FROM kunde WHERE login=„...“`
- `UPDATE kunde SET last_acc=date, tries=0 WHERE`

COMMIT

Willkommen

- `SELECT k_id, name FROM kunde WHERE login=„...“`
- `SELECT last_pur FROM purchase WHERE k_id=...`

COMMIT

Bestellung

- `SELECT av_qty FROM stock WHERE p_id=...`
- `UPDATE stock SET av_qty=av_qty-1 where ...`
- `INSERT INTO shop_cart VALUES( o_id, k_id, ...`

COMMIT

Best. löschen

- `DELETE FROM shop_cart WHERE o_id=...`
- `UPDATE stock SET av_qty=av_qty+1 where ...`

COMMIT

OLAP Beispiele

- Welche Produkte hatten im letzten Jahr im Bereich Potsdam 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 Durchschnitt?
  - Wie hoch waren die Bestellungen im Vergleich zu den durchschnittlichen Bestellungen des jeweiligen Kunden in einem vergleichbaren Zeitraum?
  - Lohnen sich Mailing-Aktionen?
- Haben solche Zweigstellen einen höheren Umsatz, die gemeinsam gekaufte Produkte nebeneinander platzieren?
  - Welche Produkte werden überhaupt zusammen gekauft – und wo?

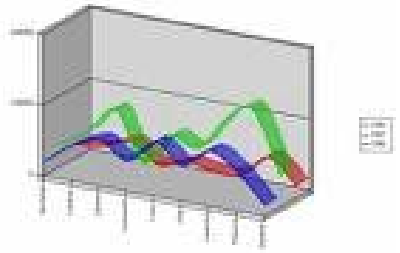
OLAP versus OLTP

transaktionales System

	OLTP	OLAP
Typische Operationen	Insert, Update, Delete, Select	Select, Bulk-Inserts
Transaktionen	viele, kurze	Lange Lesetransaktionen
Typische Anfragen	Einfache Anfragen, Primärschlüsselzugriff, Schnelle Abfolgen von Selects/inserts/updates/deletes	Komplexe Anfragen: 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 Sek.	Minuten
Modellierung	Anwendungsorientiert	Themenorientiert
Typische Benutzer	Sachbearbeiter, Kunde	Management

Systeme für große Datenmengen
-> gut geeignet für Durchsuchen der Daten

DWH Architektur & Komponenten



Monitoring

Analysewerkzeuge

Metadaten

Quelle 1
RDBMS

Quelle 2
IMS

Staging
Area

Staging
Area

Cube

Mart 2

Mart 1

Datenquellen

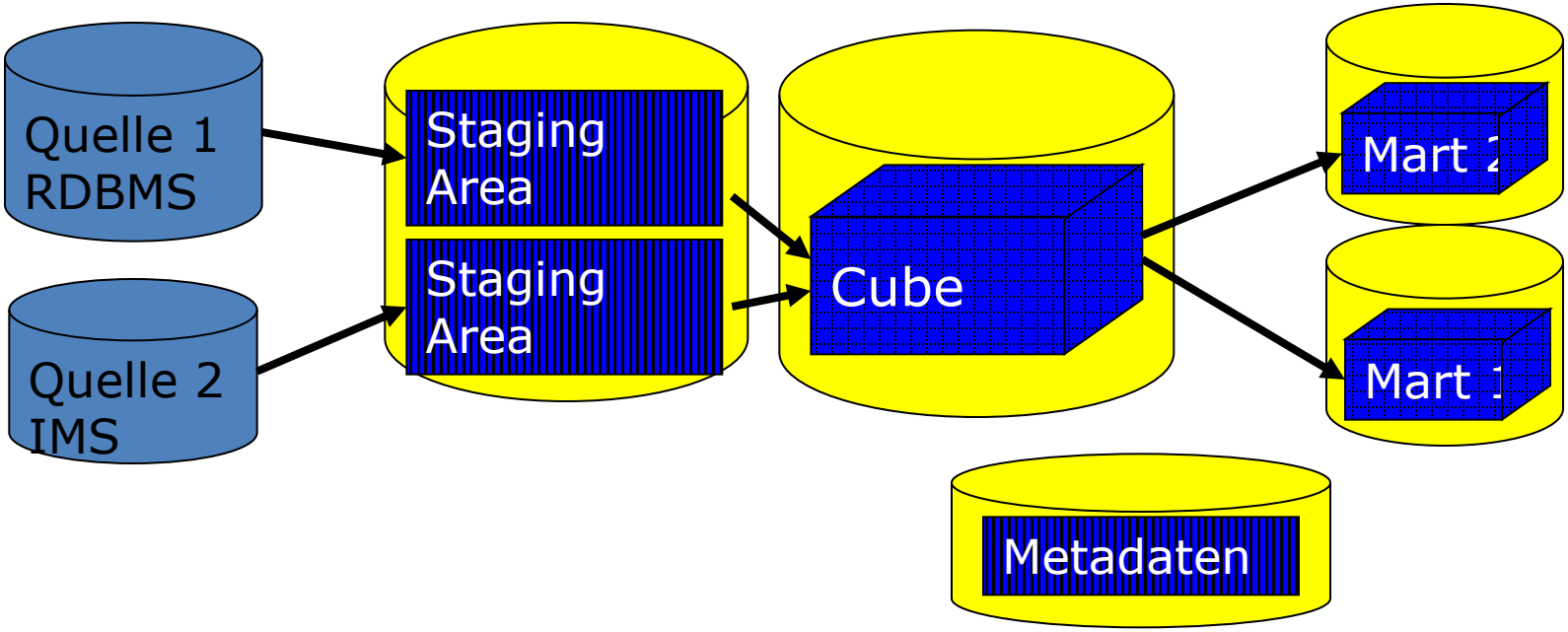
Basisdaten

Abgeleitete
Sichten

Arbeitsbereich

Alternativen

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



Arbeitsbereich

- Staging Area
  - Temporärer Speicher
  - Quellnahes Schema
- Motivation
  - ETL Arbeitsschritte effizienter implementierbar
    - Mengenoperationen, SQL
  - Zugriff auf Basisdatenbank möglich (Lookups)
  - Vergleich zwischen Datenquellen möglich
  - Filterfunktion: Nur einwandfreie Daten in Basisdatenbank übernehmen

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

Überblick

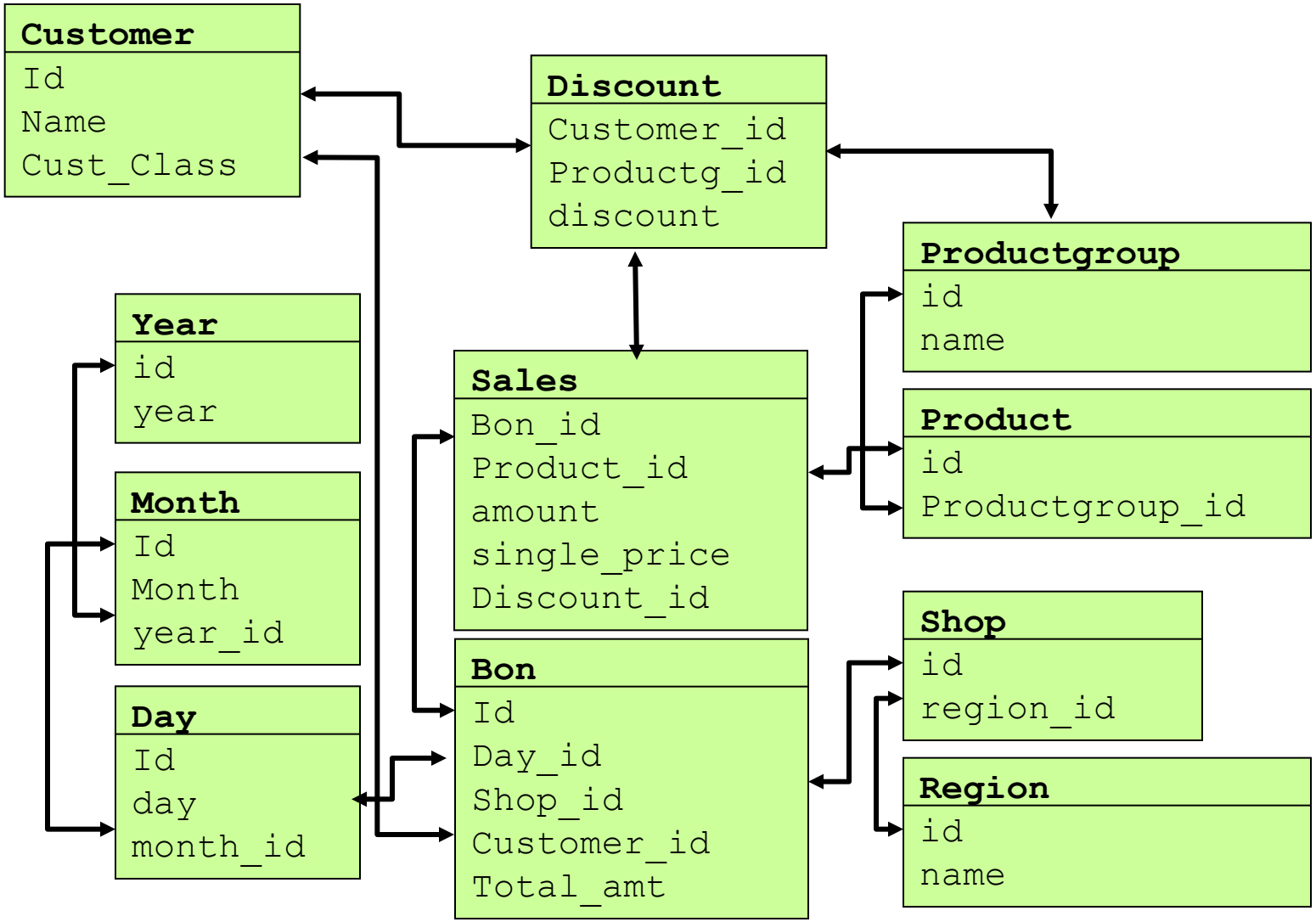
- Einsatzgebiete
- OLAP versus OLTP
- **Multidimensionale Modellierung**
- OLAP Operationen
- Relationale Implementierung

Datenmodell für analytische Anfragen (nicht für OLTP geeignet!)
-> Fokus auf schneller Aggregation & Analyse von Daten, weniger Normalisierung, verhindern von Redundanz…

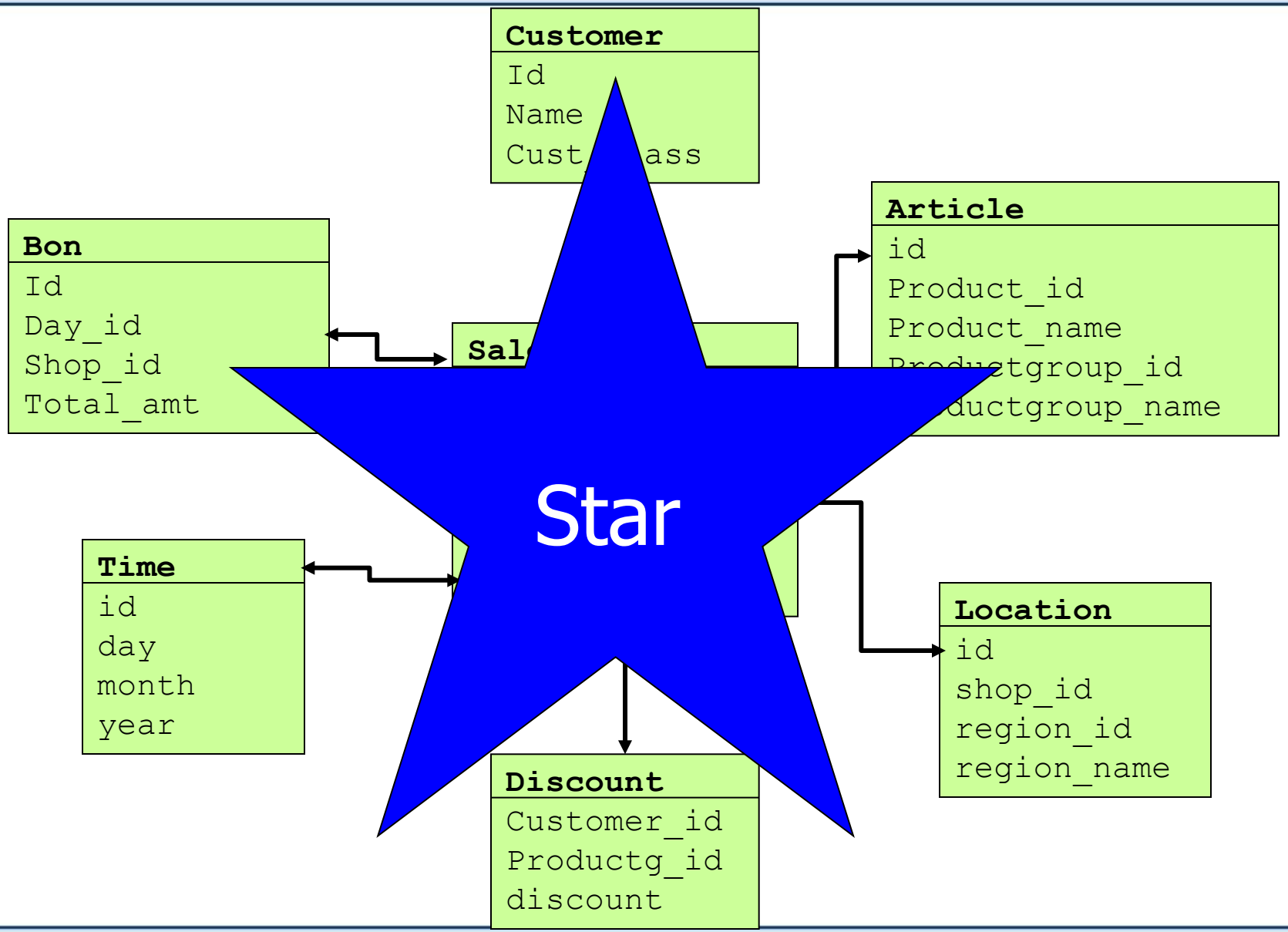
Daten sind unterteilt in:
- Fakten (messbare Daten, z.B. Verkaufspreis)
- Dimensionen (beschreiben die Fakten)
-> idR hierarchisch geordnet
z.B. Zeit: Sekunde < Stunde < Tag ….

—> Typische Anfrage: Aggregiere Fakten nach Dimensionen

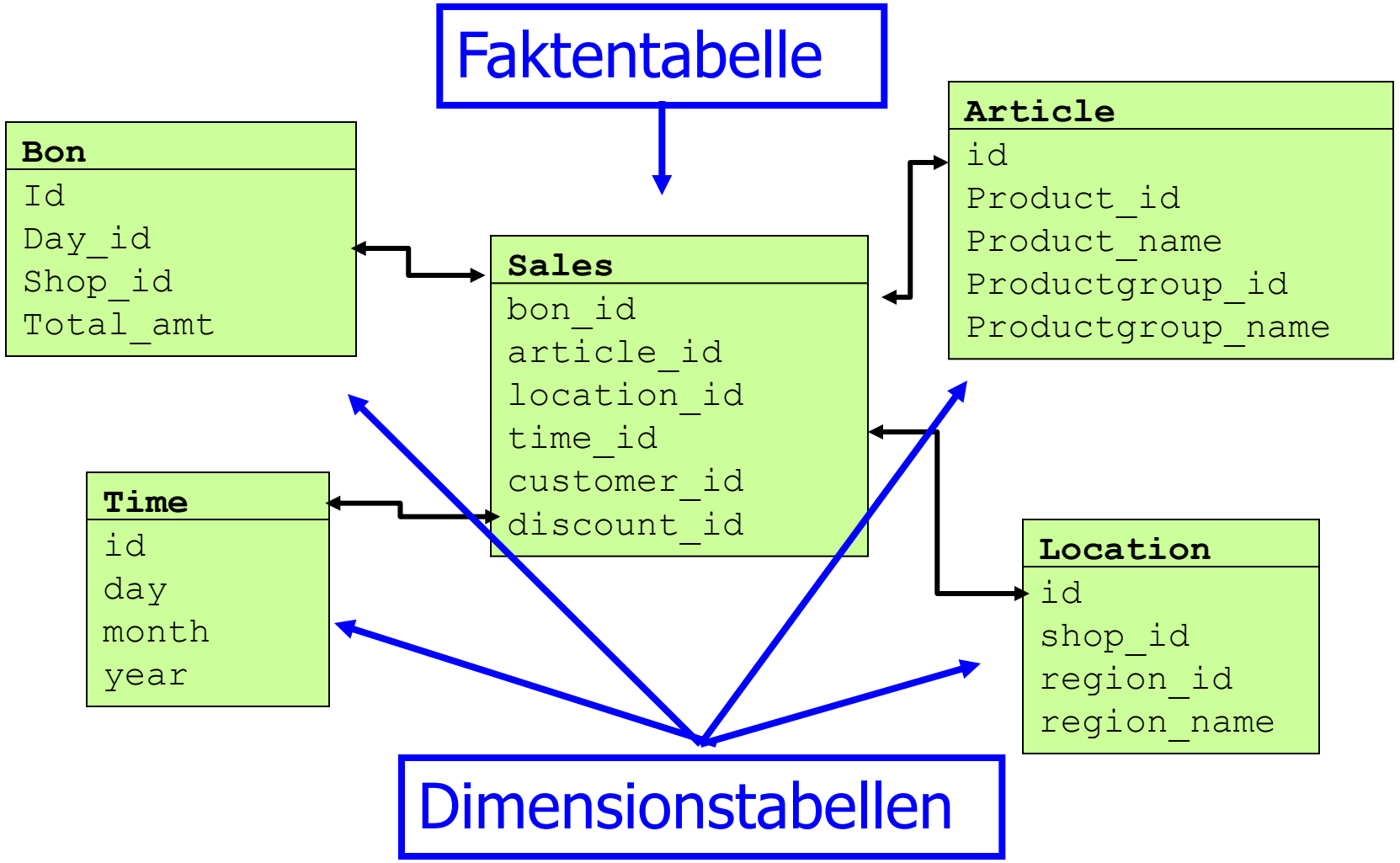
Beispiel: Normalisiertes Schema



Multidimensionales Schema



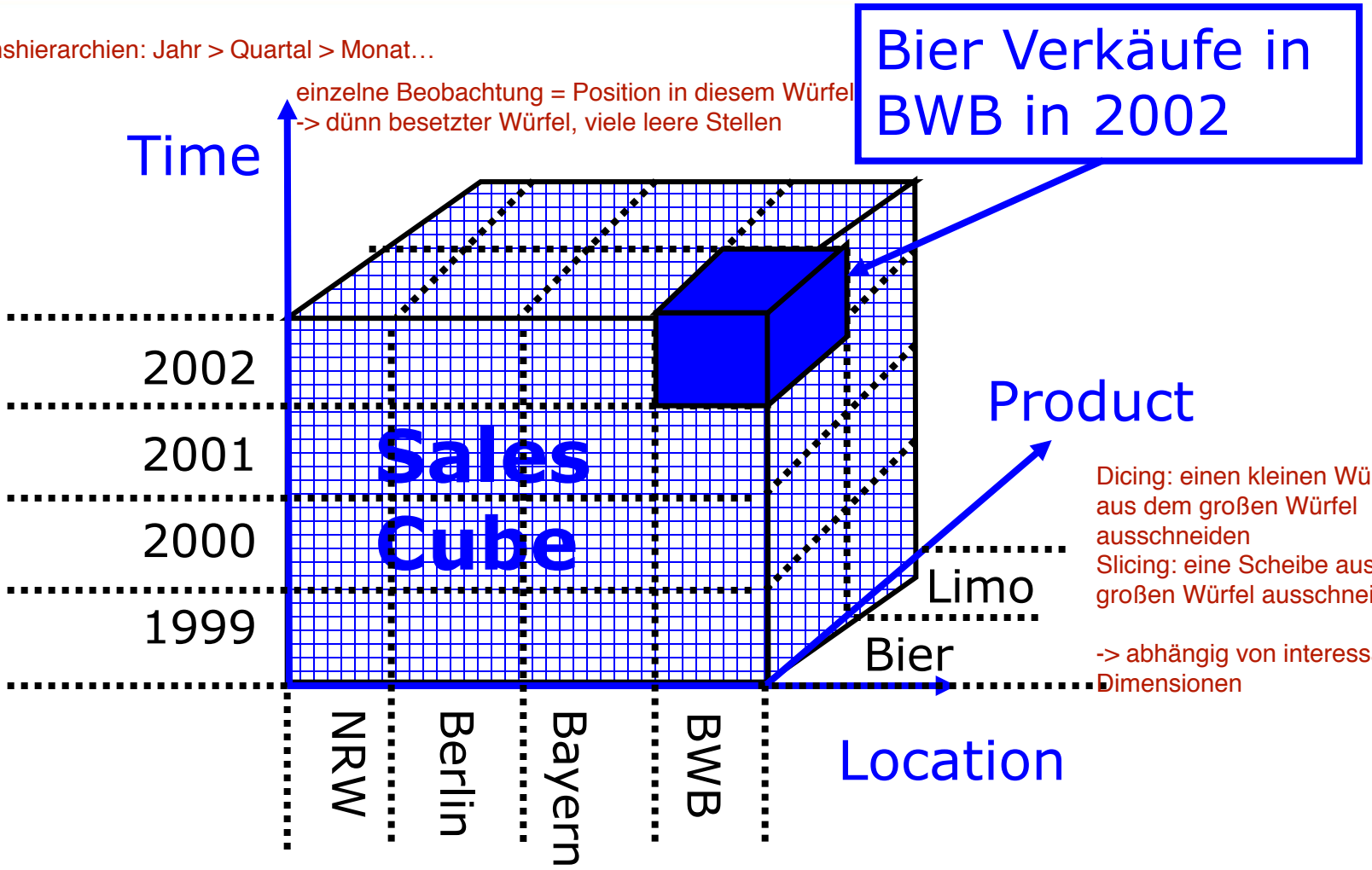
Begriffe


Faktentabelle
Bon
Id
Day_id
Shop_id
Total_amt
Sales
bon_id
article_id
location_id
time_id
customer_id
discount_id
Article
id
Product_id
Product_name
Productgroup_id
Productgroup_name
Time
id
day
month
year
Location
id
shop_id
region_id
region_name
Dimensionstabellen

Cube

Dimensionshierarchien: Jahr > Quartal > Monat…

einzelne Beobachtung = Position in diesem Würfel
-> dünn besetzter Würfel, viele leere Stellen



Dicing: einen kleinen Würfel aus dem großen Würfel ausschneiden
Slicing: eine Scheibe aus dem großen Würfel ausschneiden

-> abhängig von interessanten Dimensionen

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

Dimensionen

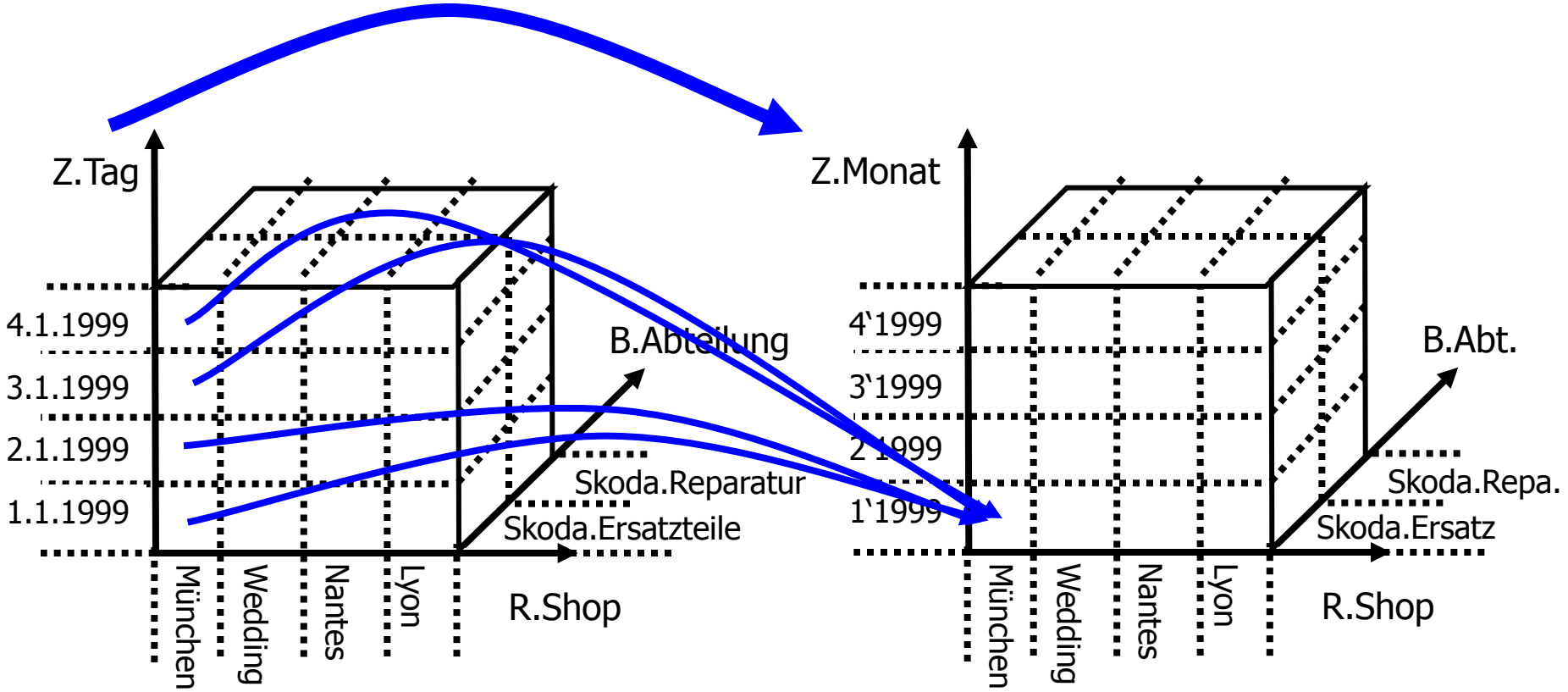
- Eindeutige Strukturierung des Datenraums
- Jede Dimension hat ein Schema
  - Tag, Woche, Jahr
  - Landkreis, Land, Staat
  - Produkt, Produktgruppe, Produktklasse, Produktfamilie
- ... und Wertebereiche
  - (1, 2, 3, ..., 31), (1, ... 52), (1900, ..., 2003)
  - (...), (Berlin, NRW, Department-1, ...), (BRD, F, ...)

Überblick

- Einsatzgebiete
- OLAP versus OLTP
- Multidimensionale Modellierung
- **OLAP Operationen**
- Relationale Implementierung

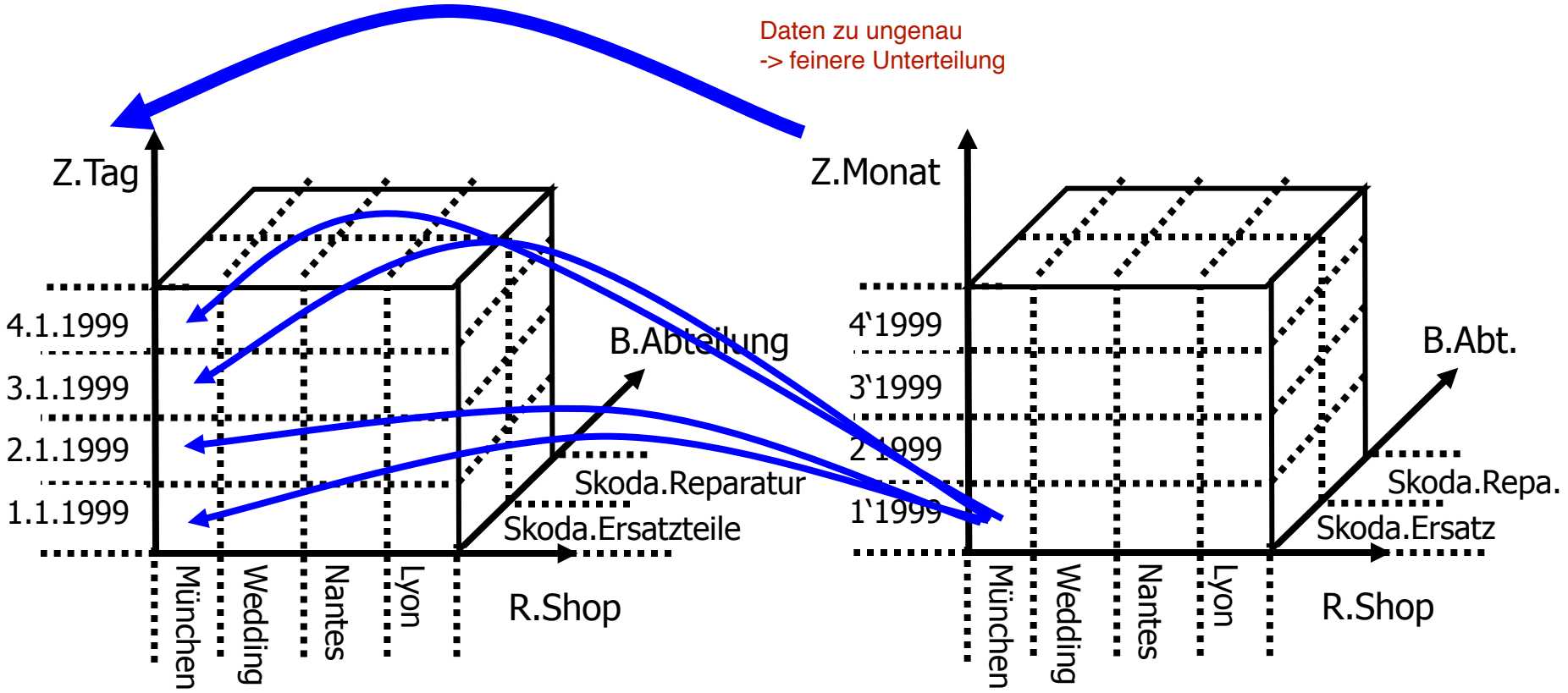
Beispiel: Aggregation (Roll-Up)

Daten zu fein unterteilt:
-> Zusammenfassen nach Dimension



Hier: Zusammenfassen der Daten einzelner Tage zu Monaten

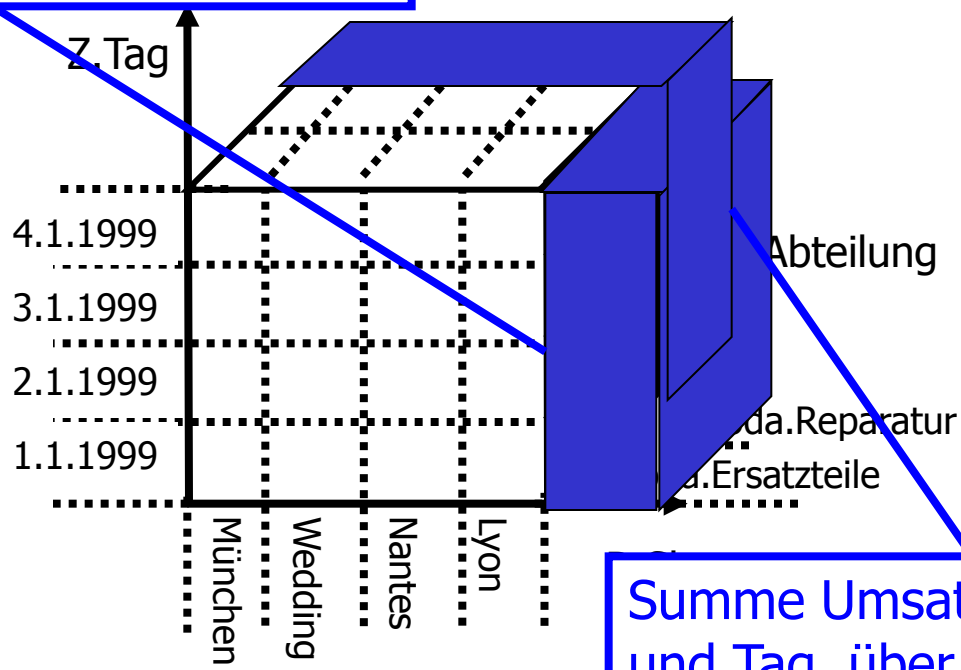
Beispiel: Verfeinerung (Drill-Down)



Hier: Statt Daten pro Monat, Daten pro Tag im betrachteten Monat

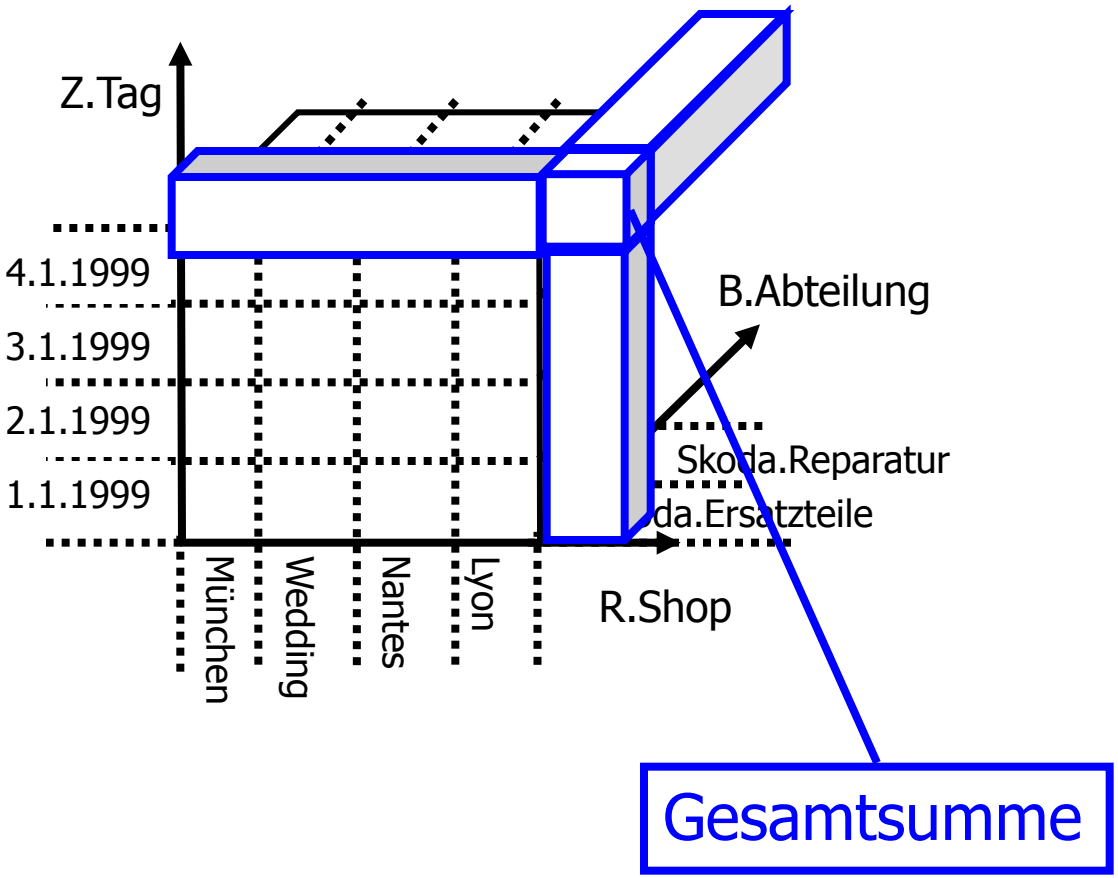
Aggregation bis TOP

Summe Umsatz pro Tag und Abteilungen über alle Shops



Summe Umsatz pro Shop und Tag, über alle Abteilungen

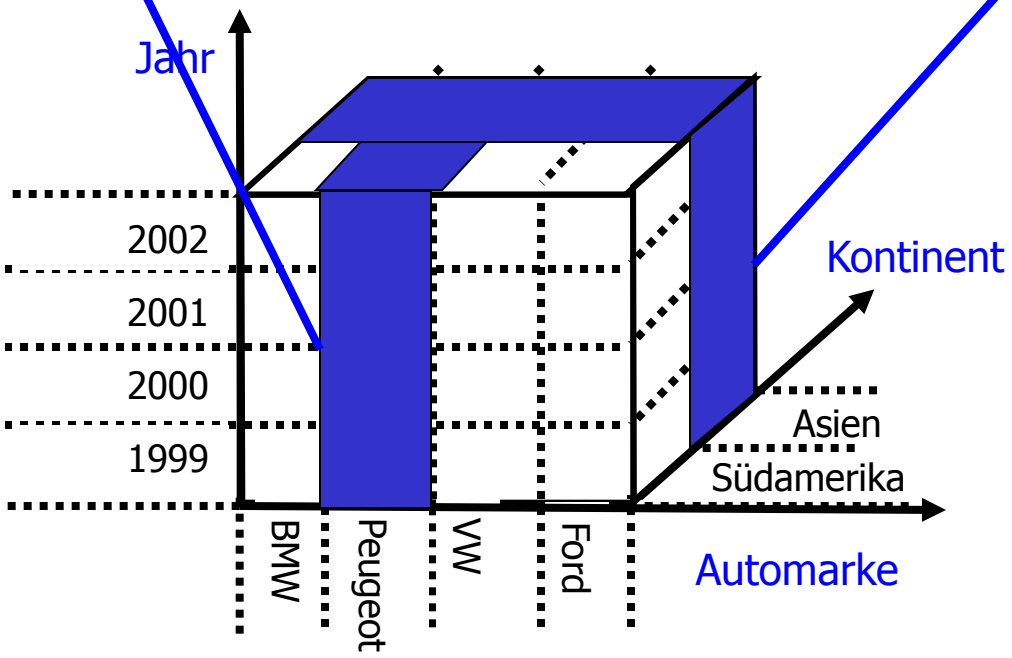
... in mehreren Dimensionen



Selektion einer Scheibe (*Slicing*)

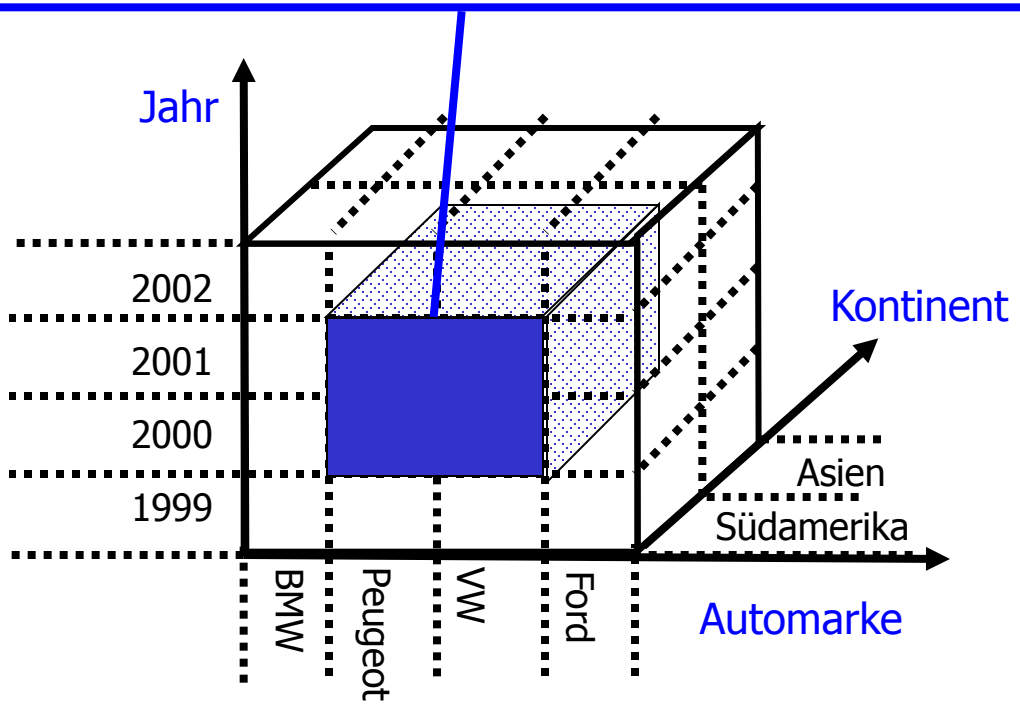
Verkäufe von Peugeot pro Jahr und Kontinent

Verkäufe in Asien pro Jahr und Marke



Auswahl von Unterwürfeln (*Dicing*)

Verkäufe von (Peugeot, VW) in (2000, 2001) pro Kontinent



Create your Own Exam: Data Warehousing

- Bitte erstellen Sie eine Multiple Choice Aufgabe zum Thema SQL
  - Formulieren Sie eine Frage und 3 Antworten (A, B, C)
  - Davon sollte mindestens eine Antwort richtig und mindestens eine Antwort falsch sein

- Geben Sie die Aufgabe an Ihren rechten Nachbarn. Diskutieren Sie gemeinsam und markieren Sie die richtigen Lösungen

- Geben Sie am Ende der Vorlesung Ihre Aufgabe bei mir ab

5 min

Überblick

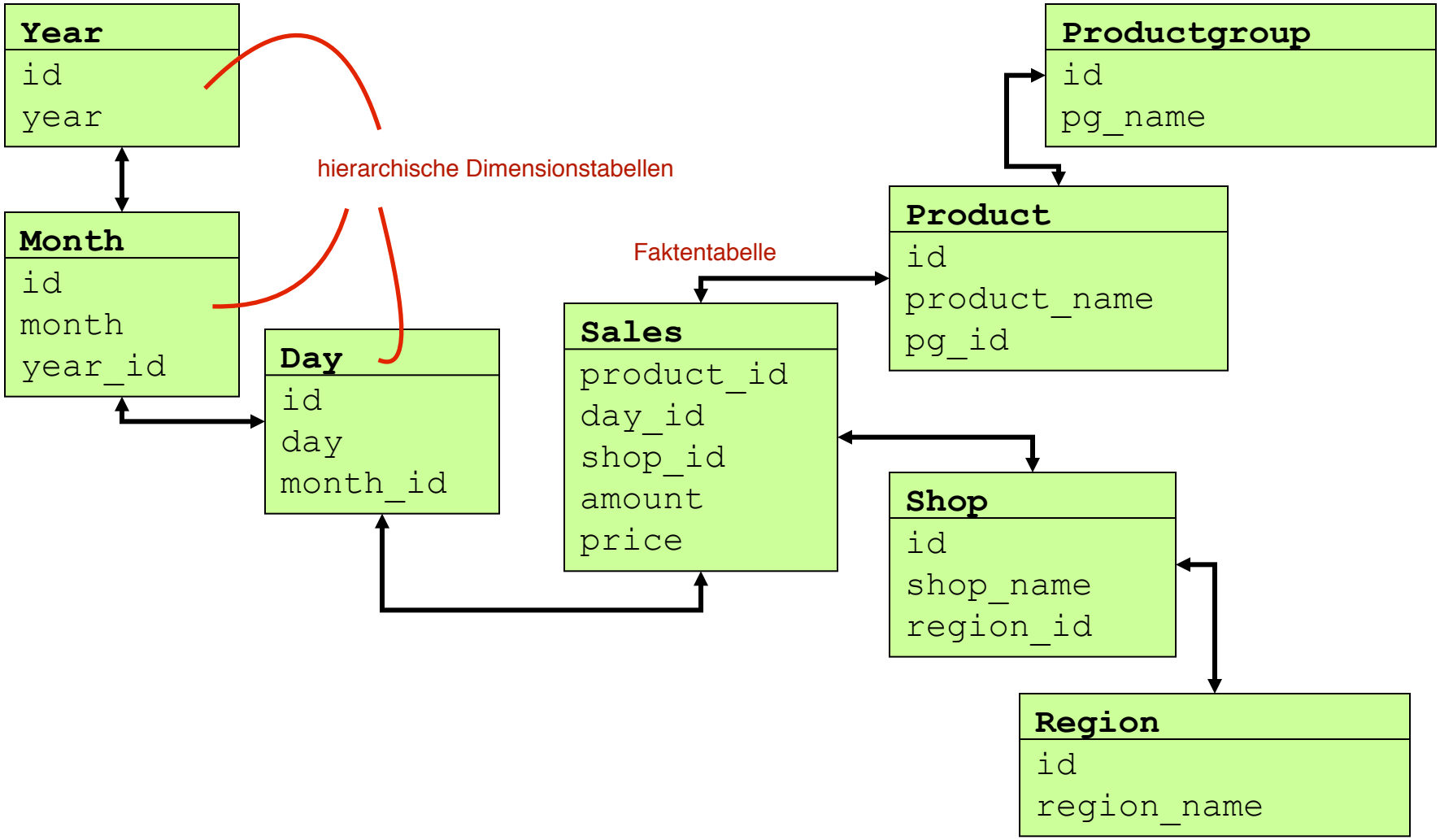
- Einsatzgebiete
- OLAP versus OLTP
- Multidimensionale Modellierung
- OLAP Operationen
- **Relationale Implementierung**

Problem:
Abbildung des OLAP-Würfels auf relationale Tabellen
Weil: die meisten kommerziellen OLAP-Systeme bauen intern auf relationalen Datenbanken auf (ROLAP - relational OLAP)

Variante 1: ER-Schema
-> Ungünstig weil ER für OLTP-Anwendungen optimiert, weniger für typische OLAP-Anfragen
stattdessen alternative Schemata (siehe nächste Folien)

Variante 1 - Snowflake

Bessere Darstellung von Dimensions-Hierarchien (in der Struktur)
-> dafür etwas langsamer als Stern-Schema (mehr Joins nötig, da mehr Tabellen)

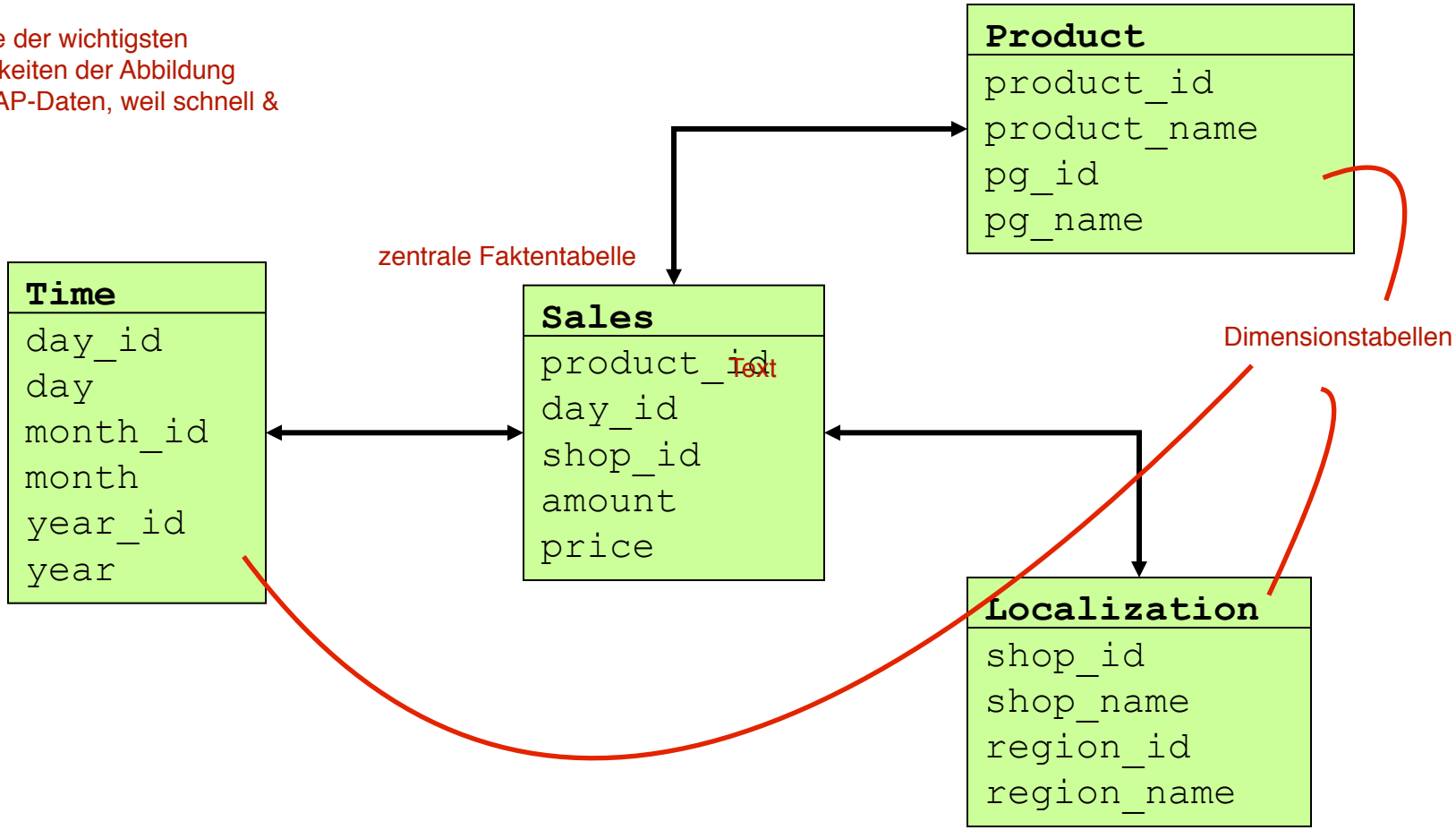


Variante 2: Star Schema

Immer der gleiche Aufbau:
zentrale Faktentabelle plus Dimensionstabellen
-> wenige Joins notwendig

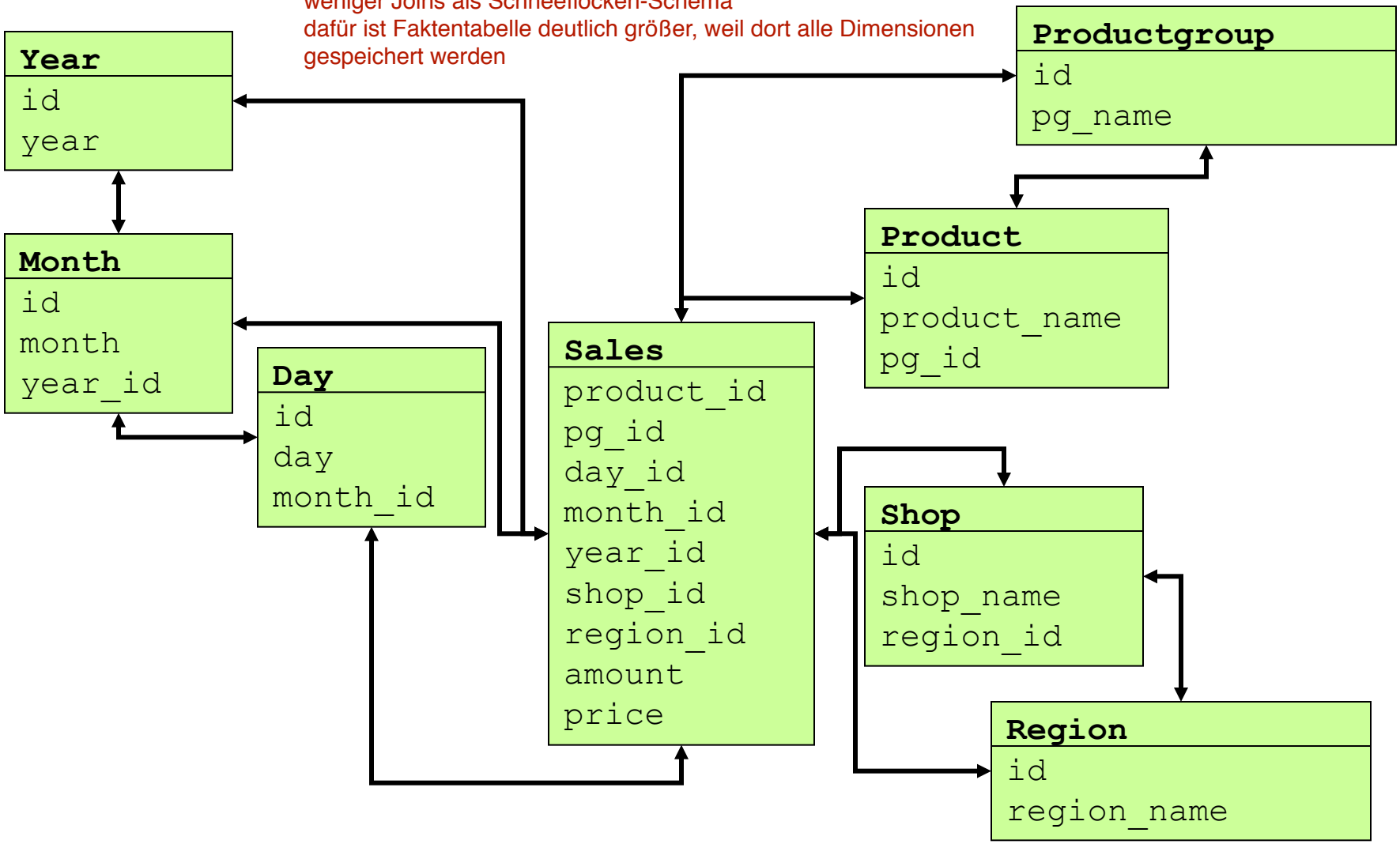
—> eine der wichtigsten Möglichkeiten der Abbildung von OLAP-Daten, weil schnell & einfach

Problem:
Hierarchien der Dimensionen sind nicht gut abgebildet (z.B. bei Time nur als Attribute
-> Verfeinerung zu Schneeflockenschema



Variante 3: Fullfact

Idee: Mischung aus Stern & Schneeflocken
(denormalisierte Dimensionen -> können sich ändern)
weniger Joins als Schneeflocken-Schema
dafür ist Faktentabelle deutlich größer, weil dort alle Dimensionen gespeichert werden



Fazit – Speicher und Query

- Speicherverbrauch Snowflake / Star praktisch identisch
  - Wenn Bedarf für Dimensionen vernachlässigbar
- Fullfact mit deutlich höherem Speicherverbrauch
  - Faktentabelle wird breiter
  - Dafür minimale Anzahl Joins
- Anzahl Joins: FullFact < Star < Snowflake
- Laufzeitverhalten hängt von mehr Faktoren als dem Schema ab
  - Bereichs- oder Punktanfrage
  - Indexierung
  - Selektivität der Bedingungen
  - Gruppierung und Aggregation
  - ...
- … aber Joins sind tendenziell teuer

Roll-Up Operationen: Hierarchische Aggregation

- Wunsch: Verkaufsumsatz der Produktgruppe „Wein“ nach Tagen, Monaten und Jahren
- ```
  SELECT T.year_id, T.month_id, T.day_id,
       sum(s.amount)
  FROM   Sales S, Product P, Time T
  WHERE  P.pg_name=„Wein“
  AND    P.product_id = S.product_id
  AND    T.day_id = S.day_id
  GROUP BY T.year_id, T.month_id, T.day_id
  ```
- Summe nur für Tage (unterteilt nach Monaten/Jahren)
- Keine Summen pro Monat / pro Jahr
- Wunsch nicht in einer Anfrage formulierbar

1997	1	1	150
1997	1	2	130
1997	1	3	145
1997	1	4	122
...	...	...	...
1997	1	31	145
1997	2	1	133
1997	2	2	122
...	...	...	...
1997	3	10	180
1997	12	31	480
1998	1	1	240
...	...	...	...
2003	6	18	345
  ```

Hierarchische Aggregation –2-

- Alle Verkäufe der Produktgruppe „Wein“ nach Tagen, Monaten und Jahren
- Benötigt UNION und eine Anfrage pro Klassifikationsstufe

```
SELECT T.day_id, sum(amount*price)
FROM   Sales S, Product P
WHERE  P.pg_name=„Wein“ and
       P
GROUP BY
```

```
SELECT T.month_id, sum(amount*price)
FROM   Sales S, Product P, Time T
WHERE  P.pg_name=„Wein“ and
       P.pro
       T.day_
GROUP BY T.m
```

```
SELECT T.year_id, sum(amount*price)
FROM   Sales S, Product P, Time T
WHERE  P.pg_name=„Wein“ and
       P.product_id = S.product_id and
       T.day_id = S.day_id
GROUP BY T.year_id
```

ROLLUP Operator

- Herkömmliches SQL
  - Dimension mit k Stufen – Union von k Queries
  - ➢ k Scans der Faktentabelle
    - Keine Optimierung wg. fehlender Multiple-Query Optimierung in kommerziellen RDBMS
  - ➢ Schlechte Ergebnisreihenfolge

- ROLLUP Operator
  - Hierarchische Aggregation mit Zwischensummen
  - Summen werden durch „ALL“ als Wert repräsentiert

ROLLUP Beispiel

```
SELECT T.year_id, T.month_id, T.day_id, sum(...)
FROM   Sales S, Time T
WHERE  T.day_id = S.day_id
GROUP BY ROLLUP(T.year_id, T.month_id, T.day_id)
```

1997	Jan	1	200
1997	Jan	...	
1997	Jan	31	300
1997	Jan	ALL	31.000
1997	Feb	...	
1997	March	ALL	450
1997	...	...	
1997	ALL	ALL	1.456.400
1998	Jan	1	100
1998	...	...	
1998	ALL	ALL	45.000
...	...	...	
ALL	ALL	ALL	12.445.750

Multidimensionale Aggregation

	1998	1999	2000	Gesamt
Weine	15	17	13	45
Biere	10	15	11	36
Gesamt	25	32	24	81

- `sum() ... GROUP BY pg_id, year_id`
- `sum() ... GROUP BY pg_id`
- `sum() ... GROUP BY year_id`
- `sum()`

Cube Operator

- *d* Dimensionen, jeweils eine Klassifikationsstufe
  - Jede Dimension kann in Gruppierung enthalten sein oder nicht
  - 2^d Gruppierungsmöglichkeiten
- Herkömmliches SQL
  - Viel Schreibarbeit
  - 2^d Scans der Faktentabelle (wieder keine Optimierung möglich)
- CUBE Operator
  - Berechnung der Summen von sämtlichen Kombinationen der Argumente (Klassifikationsstufen)
  - Summen werden durch „ALL“ repräsentiert
  - Keine Beachtung von Hierarchien
    - Durch Schachtelung mit ROLLUP erreichbar

Einfacher SQL Ansatz

```
SELECT Marke, Farbe, SUM(Verkäufe)
FROM AutoTab
GROUP BY (Marke, Farbe)

UNION

SELECT Marke, ALL, SUM(Verkäufe)
FROM AutoTab
GROUP BY (Marke)

UNION

SELECT ALL, Farbe, SUM(Verkäufe)
FROM AutoTab
GROUP BY (Farbe)

UNION

SELECT ALL, ALL, SUM(Verkäufe)
FROM AutoTab;
```

Marke	Farbe	Verkäufe
VW	Blau	32
VW	Weiß	17
VW	Rot	5
Opel	Blau	24
Opel	Weiß	19
Opel	Rot	12
VW	ALL	54
Opel	ALL	55
ALL	Blau	56
ALL	Weiß	36
ALL	Rot	17
ALL	ALL	109

Folie: Mark Liebetrau (HPI)

Ansatz mit CUBE-Operator

- **Neuer Ansatz**
- SELECT Marke, Farbe,
  SUM(Verkäufe)
  FROM AutoTab
  GROUP BY CUBE(Marke, Farbe);

- Unterschiede in der Syntax:
  - keine UNIONs mehr notwendig
    ⇒ einfachere Anfrage
- Unterschiede in der Semantik:
  - Keine

Bisheriger Ansatz

```
SELECT Marke, Farbe, SUM(Verkäufe)
FROM AutoTab
GROUP BY (Marke, Farbe)

UNION

SELECT Marke, ALL, SUM(Verkäufe)
FROM AutoTab
GROUP BY (Marke)

UNION

SELECT ALL, Farbe, SUM(Verkäufe)
FROM AutoTab
GROUP BY (Farbe)

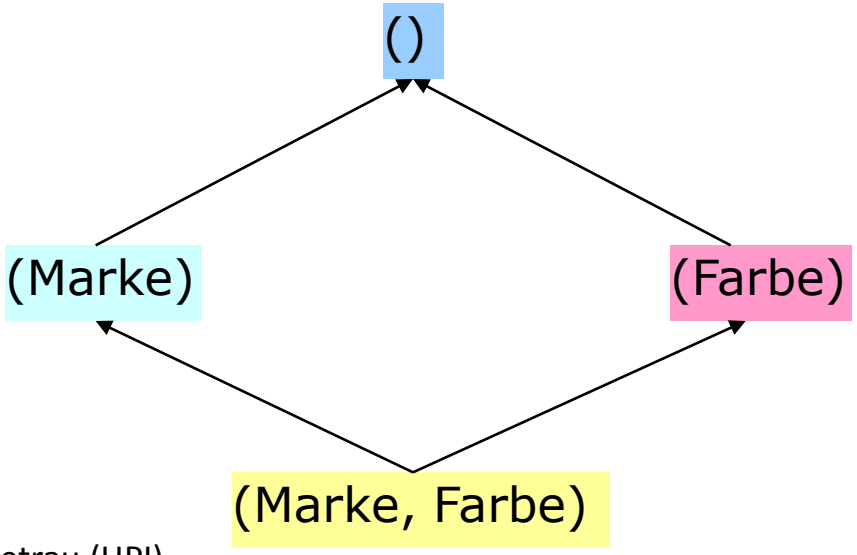
UNION

SELECT ALL, ALL, SUM(Verkäufe)
FROM AutoTab;
```

Folie: Mark Liebetrau (HPI)

Ansatz mit CUBE-Operator

- **Ableitbarkeit der Gruppen**
  - Beziehung lässt sich mithilfe eines Aggregationsgitters darstellen
  - (X,Y) ≙ Gruppierung über X und Y



Folie: Mark Liebetrau (HPI)

Marke	Farbe	Verkäufe
VW	Blau	32
VW	Weiß	17
VW	Rot	5
Opel	Blau	24
Opel	Weiß	19
Opel	Rot	12
VW	ALL	54
Opel	ALL	55
ALL	Blau	56
ALL	Weiß	36
ALL	Rot	17
ALL	ALL	109

Rückblick: Data Warehousing

- Einsatzgebiete
- OLAP versus OLTP
- Multidimensionale Modellierung
- OLAP Operationen
- Relationale Implementierung

In der nächsten Veranstaltung:
Anfrageverarbeitung
(Kapitel 16 des Lehrbuchs)

Dazu IV “Data Warehousing and Business Intelligence”
immer im WS