Data Warehousing

Operationen im multidimensionalen Datenmodell

Graphische MDDM

Summierbarkeit

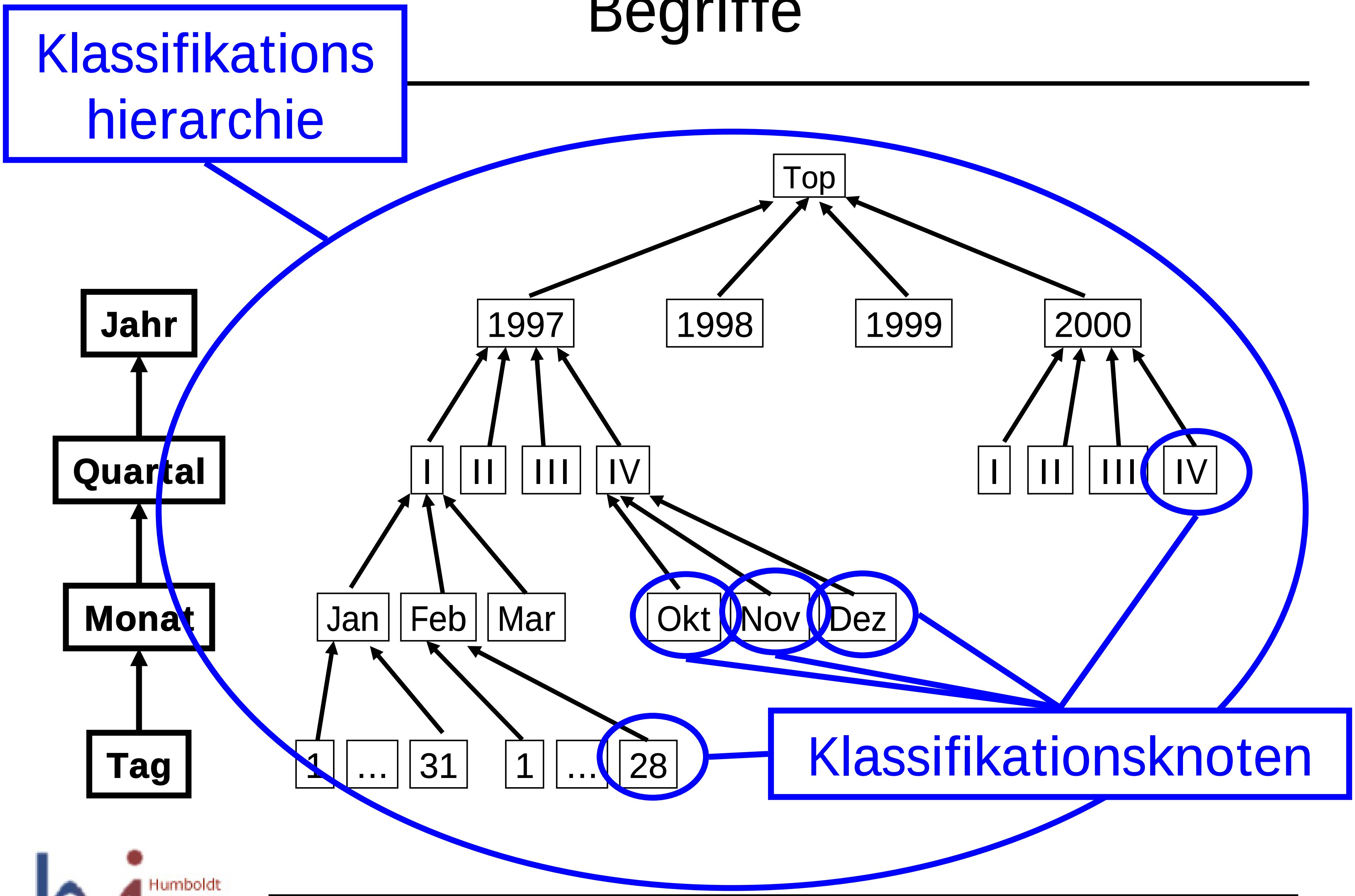
Ulf Leser

Wissensmanagement in der Bioinformatik

Klassifikationsschema

- *Definition*
  Ein Klassifikationsschema K (einer Dimension D) ist ein Quadrupel (K_k, $\rightarrow_k$, $K_{s,}$ $\rightarrow_s$) mit
  - *K_s ist die Menge von Klassifikationsstufen $\{k_0, \dots k_n\}$*
  - *„$\rightarrow_s$“ ist eine Halbordnung auf K_s mit größtem Element top(K_s)*
    - *D.h.: $\forall k \in K_s$: $k \rightarrow_s$ top(K_s)*
  - *K_k ist die Menge von Klassifikationsknoten $\{n_0, \dots n_m\}$*
  - *Jeder Klassifikationsknoten n ist genau einer Klassifikationsstufe k zugeordnet*
    - *stufe(n) = k*
    - *knoten(k) = $\{n \mid n \in K_k \wedge \text{stufe}(n)=k\}$*
  - *„$\rightarrow_k$“ ist die Halbordnung auf K_s übertragen auf K_k*
    - *$k,l \in K_s$, $k \rightarrow_s l \Rightarrow \forall n \in \text{knoten}(k), m \in \text{knoten}(l): n \rightarrow_k m$*
- Bemerkung
  - Eine Klassifikationsstufe hat mehrere Klassifikationsknoten, aber jeder Klassifikationsknoten ist genau einer Klassifikationsstufe zugeordnet
  - Wir benutzen i.d.R. einfach $\rightarrow$ für $\rightarrow_k$ oder $\rightarrow_s$

Begriffe


Klassifikations hierarchie
Top
1997
1998
1999
2000
Jahr
Quartal
Monat
Tag
I
II
III
IV
I
II
III
IV
Jan
Feb
Mar
Okt
Nov
Dez
1
...
31
1
...
28
Klassifikationsknoten

Klassifikationspfade

- *Definition*
  Ein Klassifikationspfad P in einem Klassifikationsschema K mit Klassifikationsstufen K_s ist eine Menge $\{p_0, ..., p_m\}$ mit
  - $\{p_0, ...,p_m\} \subseteq K_s$
  - $p_m = top(K_s)$
  - $\forall p_i$, $1\leq i\leq m$: $p_{i-1} \rightarrow p_i$ und $\nexists q$: $p_{i-1} \rightarrow q \rightarrow p_i$
  - *Die Länge des Pfades P ist |P|=m+1*
  - *Der Klassifikationslevel von p_i in P ist i*
- Bedeutung
  - Ein Pfad ist damit eine voll geordnete Teilmenge von K_s
  - Jeder Pfad beinhaltet das größte Element TOP
  - Verdichtung wird nur entlang von Klassifikationspfaden definiert
    - Und damit entlang funktionaler Abhängigkeiten

Dimension

- *Definition*
  Eine Dimension $D=(K, \{P_1,\ldots,P_j\})$ besteht aus
  - *Einem Klassifikationsschema K*
  - *Einer Menge von Pfaden P_i in K*
- Bemerkungen
  - D muss nicht alle Pfade enthalten, die es in K gibt
    - Designentscheidung
  - Nicht alle Klassifikationsstufen von K müssen in einem Pfad enthalten sein
    - Aber man wird seine Pfade so wählen, dass doch
- Schreibweise
  - D.k bezeichnet eine Klassifikationsstufe k aus D
  - Ein D.k kann in mehreren Pfaden vorkommen

Granularität

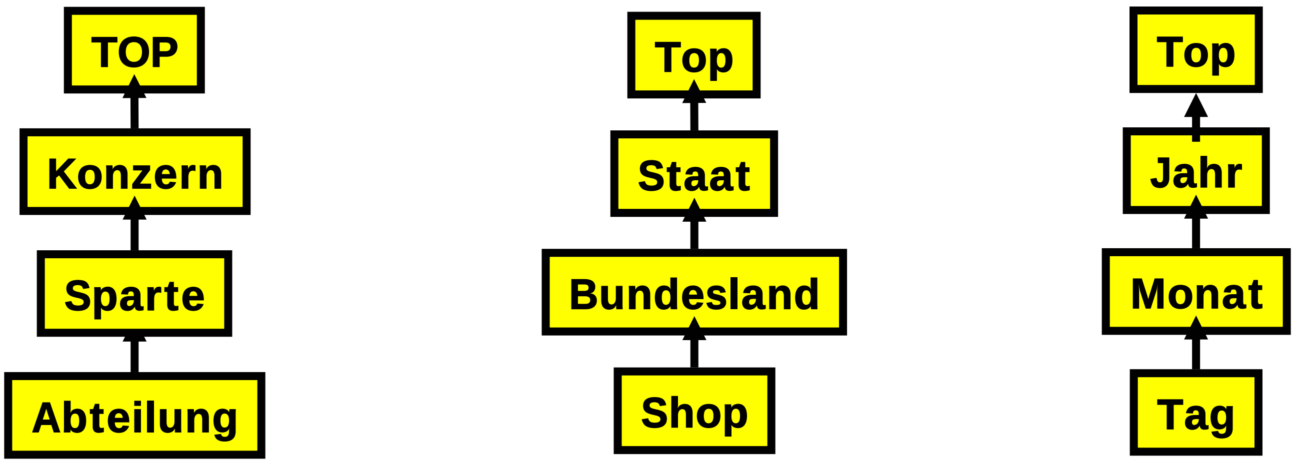
- *Definition*
  Gegeben eine Menge U von n Dimensionen $D_1, \ldots, D_n$. Eine Granularität G über U ist eine Menge $\{D_1.k_1, \ldots, D_n.k_n\}$ für die gilt
  - *k_i ist eine Klassifikationsstufe in D_i*
  - *Es gibt keine funktionalen Abhängigkeiten zwischen den Klassifikationsstufen $D_1.k_1, \ldots, D_n.k_n$*
- Bemerkungen
  - Abkürzung: Lässt man in U eine Dimension D_i weg, meint dies implizit D_i.TOP
  - Zweite Bedingung ist immer erfüllt, wenn keine funktionalen Abhängigkeiten zwischen Dimensionen bestehen
    - Beispiel: Nicht gleichzeitig Dimensionen Zeit und „Fiskalisches Jahr“ in einer Granularität betrachten
  - Mit einer Granularität legt man fest, in welcher Detailstufe Fakten in den einzelnen Dimensionen beschrieben werden
    - Eine Granularität ist ein Hyperwürfel in einer bestimmten Auflösung
    - Operationen navigieren zwischen Granularitäten

Halbordnung auf Granularitäten

- *Definition*
  Auf der Menge aller Granularitäten zu einer Menge U von Dimensionen ist eine Halbordnung „≤" wie folgt definiert
  - *Sei* $G_1=\{D_1^1.k_1^1, \dots, D_n^1.k_n^1\}$ *und* $G_2=\{D_1^2.k_1^2, \dots, D_n^2.k_n^2\}$
  - *Ordne die Dimensionen in* G_1*‘ und* G_2*‘ beliebig, aber gleich*
  - *Es gilt* $G_1 \leq G_2$ *genau dann wenn*
    - $\forall i$: $D_i^1.k_i^1 \rightarrow D_i^2.k_i^2$ *oder* $D_i^1.k_i^1 = D_i^2.k_i^2$
- Benutzung
  - Beschreibung der Transformation von Granularitäten
  - Anfrageoptimierung: Wiederverwendung von Aggregaten

Beispiel

Bereich **Region** **Zeit**




```
(B.Sparte, R.Shop, Z.Tag)
≤ (B.Sparte, R.Shop, Z.Monat)
≤  (B.Sparte, R.Top, Z.Monat)
≤  (B.Top, R.Top, Z.Top)
```

```
(B.Sparte, R.Staat, Z.Tag)
? (B.Konzern, R.Shop, Z.Tag)
```

Inhalt dieser Vorlesung

- Operationen im multidimensionalen Datenmodell
- ME/R: Graphische multidimensionale Datenmodellierung
- Summierbarkeit

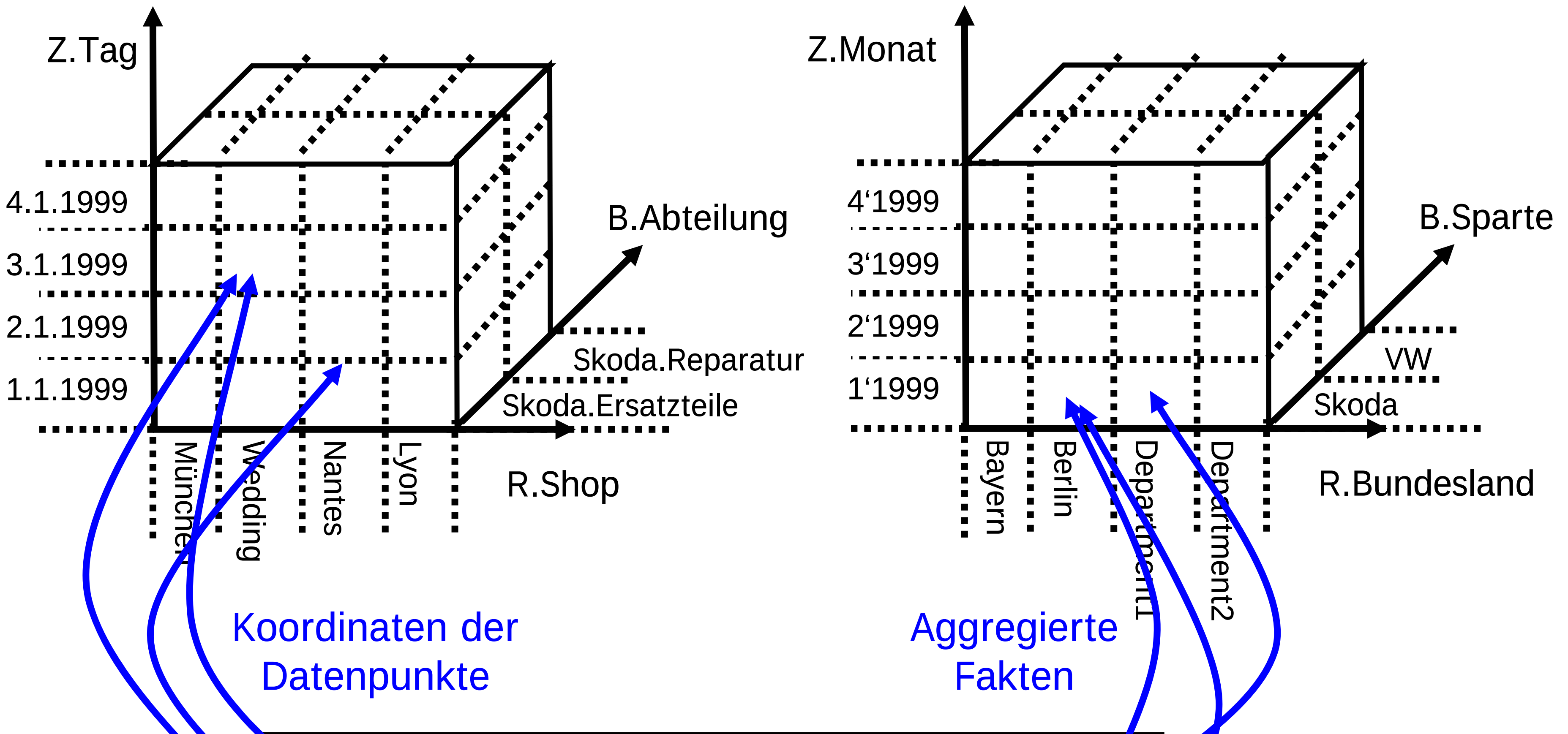
1. OLAP Operationen

- MDDM spricht die Sprache des Analysten
  - Operationen auf dem MDDM sollten die Analysevorgänge abbilden bzw. unterstützen
  - Ziel: Schnelle und intuitive Manipulation der Daten
- Notwendig
  - Definition der Operationen
  - (Unterstützung durch graphische Werkzeuge)
    - Heute: ME/R
  - Umsetzung in ausführbare Operationen
    - Ab nächster Stunde

Vorgehen

- Die wichtigsten Operationen im MDDM sind
  - Aggregation (Roll-Up)
  - Verfeinerung (Drill-Down)
- Weiteres Vorgehen
  - Datenpunkte und Würfelkoordinaten
  - Aggregation in Klassifikationshierarchien
  - Inhalt eines Würfel einer bestimmten Granularität
  - OLAP Operationen
  - Visualisierung

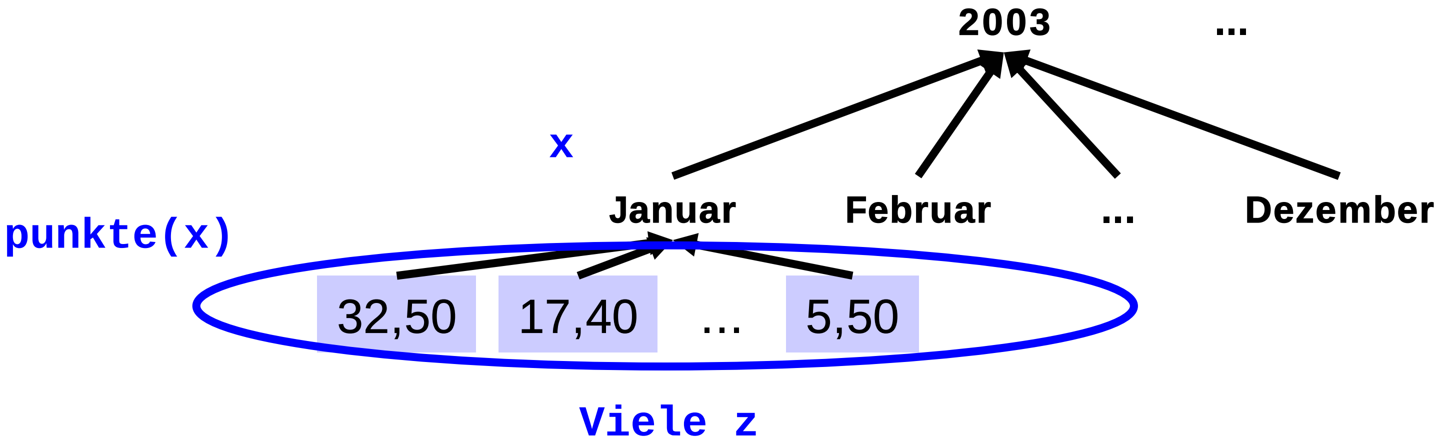
Übersicht




```
 (1.101,3.1.1999, Wedding, Ersatzteile)
(12.400, 1.1.1999, Nantes, Reparatur)
 (1.540, 3.1.1999, Wedding, Ersatzteile)
```

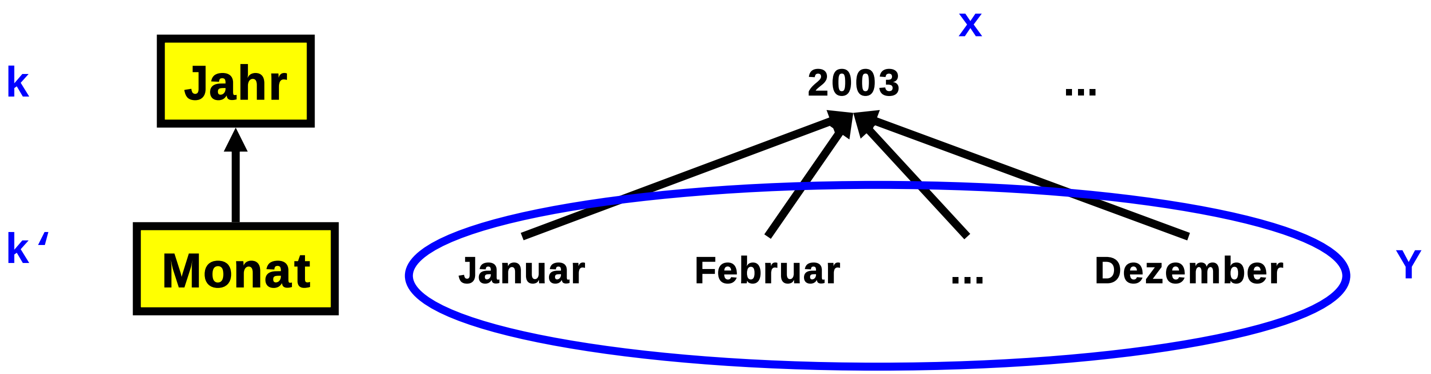
Datenpunkte und Würfelkoordinaten

- *Definition.*
  - *Gegeben Würfel W, Dimension D und ein Klassifikationsknoten x aus D*
  - *Sei z ein Datenpunkt (Fakt) und D(z) seine Koordinate bzgl. D*
  - *D(z) bzw. z liegt in x gdw. entweder D(z)=x oder D(z)$\in$nachfahren(x)*
  - *punkte(x) sei die Menge aller Datenpunkte z mit D(z) in x*
- Bemerkung
  - Erweiterung auf mehrere Koordinaten durch Schnittmengenbildung
    - punkte(x,y,z) = punkte(x) $\cap$ punkte(y) $\cap$ punkte(z) mit x in D_1, y in D_2, z in D_3



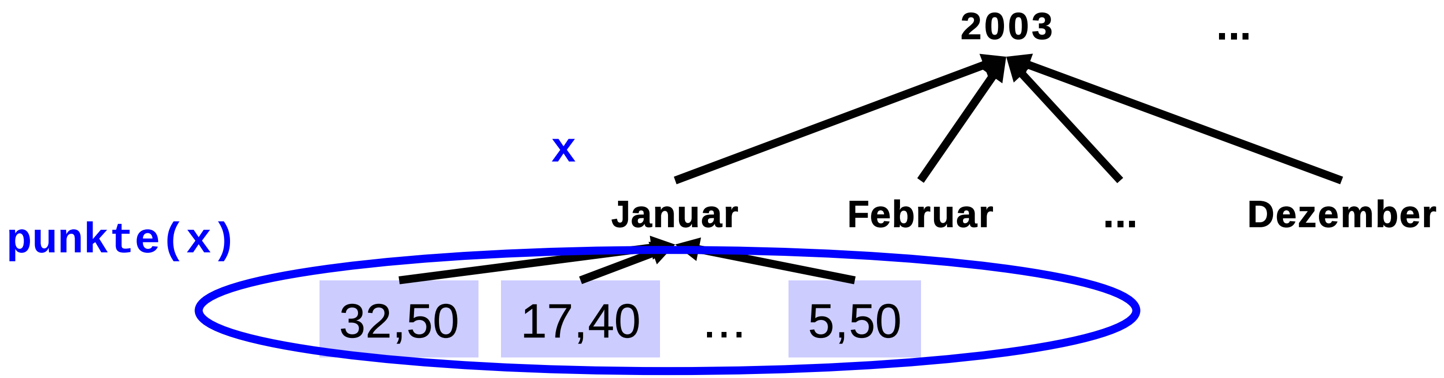
Aggregation in Hierarchien

- *Definition.*
  - *Gegeben*
    - *Dimension D, Pfad $P=\{k_0 \rightarrow \ldots \rightarrow k' \rightarrow k \rightarrow \ldots \rightarrow TOP\}$ in D*
    - *$x \in$ knoten(k) ein Klassifikationsknoten der Klassifikationsstufe $k \in P$*
    - *Aggregatfunktion f , Measure F*
  - *Sei Y=kinder(x) (die Menge $\{y_1,\ldots,y_n\} \subseteq$ knoten(k') von Klassifikationsknoten von k', von denen x funktional abhängt)*
  - *Die Aggregation von Y nach x bzgl. f und F bezeichnet die Berechnung des aggregierten Faktes $F(x)=f(F(y_1),\ldots., F(y_n))$*
  - *Die Aggregation von k' nach k bzgl. f und F bezeichnet die Berechnung von F(x) für alle $x \in$ knoten(k)*
    - Das schreiben wir kurz als F(k)



Startpunkt

- Sei $P=\{k_0 \rightarrow \ldots \rightarrow k_n \rightarrow TOP\}$
- Man berechnet F(TOP) aus $F(k_n)$, $F(k_n)$ aus $F(k_{n-1})$, etc.
- Wie berechnet man aber $F(k_0)$?
  - Aus den einzelnen Datenpunkten
  - Bzw. das sind die einzelnen Datenpunke
- *Definition.*
  - *Gegeben*
    - *Dimension D, Pfad* $P=\{k_0 \rightarrow \ldots \rightarrow TOP\}$ *in D*
    - *Ein Klassifikationsknoten* $x \in knoten(k_0)$
    - *Aggregatfunktion f , Measure F*
  - *Sei* $punkte(x)=\{z_1,\ldots,z_n\}$ *die Menge aller Datenpunkte mit* $D(z_i)$ *in x*
  - *Dann berechnet sich das aggregierte Fakt F(x) als* $F(x)=f(F(z_1),\ldots,F(z_n))$
    - *Mit F(z) als dem Wert des Measures F von Datenpunkt z*



Berechnung

- Die letzte Definition ist prinzipiell abwendbar für alle Klassifikationsknoten bzw. –stufen eines Pfades
  - Bedingt durch „ ...D-Koordinate in x...“
  - Also können wir F(x) für x aus beliebiger Klassifikationsknoten aus den Datenpunkten berechnen
- Aber
  - Es schneller, die Aggregation aus der nächstkleineren Stufe zu berechnen
    - Weniger Werte - weniger Arbeit
    - Das setzt voraus, dass man die auch hat – Präaggregation
  - OLAP Operationen wandern immer nur eine Stufe in einem Pfad herauf oder herab – deshalb definieren wir Aggregation von Ebene zu Ebene

Wo sind wir?

- Wir können entlang eines Pfades in einer Dimension für ein Measure mit einer Aggregatfunktion die aggregierten Fakten für höherstufige Knoten berechnen
- Zwei Möglichkeiten
  – Aggregierte Measures von Knoten einer Stufe aus aggregierten Measures der nächstkleineren Stufe berechnen
    - Verlangt Präaggregation
  – Aggregierte Measures immer direkt aus den Werten aller Datenpunke mit den entsprechenden Koordinaten berechnen
    - Ist eine Anfrage über dem Cube

Würfelinhalt

- Ein Würfel $W=(G,F)$
  - Granularität $G=(D_1.k_1,\ldots,D_n.k_n)$
  - Menge Measures $F=\{F_1,\ldots,F_m\}$ mit Aggregatfunktionen $f_1,\ldots,f_m$
- Schreibweise für Zellen eines Würfels
  - $W(x_1, x_2,\ldots, x_n) = (f_1,\ldots,f_m)$
  - x_i sind die Koordinaten im Würfel: $x_i \in knoten(k_i)$
  - Pro Punkt in W gibt es m (evt. aggregierte) Measures $f_1,\ldots, f_m$
- Betrachten wir den einfachen Fall: $n=m=1$
  - Dimension D mit Knoten x, ein Measure F mit Aggregatfunktion f
  - Dann: $W(x) = F(x) = f(\ \{F(y) | y \in kinder(x)\}\ )$
- Bemerkung
  - Auf unterster Ebene ist kinder(x)=punkte(x)

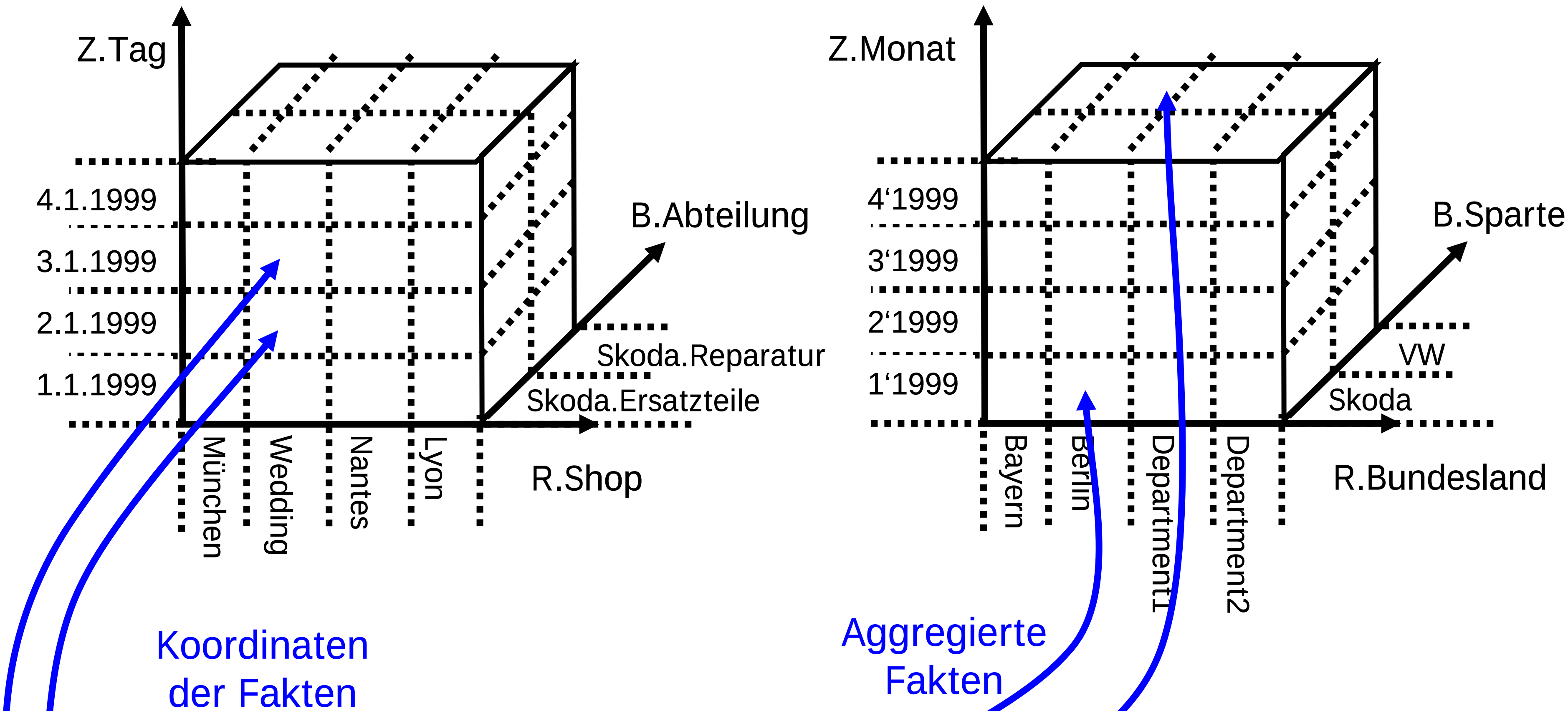
Würfelinhalt, allgemeiner Fall

- Ein Würfel W=(G,F)
  - Granularität $G=(D_1.k_1,\ldots,D_n.k_n)$
  - Menge Measures $F=\{F_1,\ldots,F_m\}$ mit Aggregatfunktionen $f_1,\ldots,f_m$
- Allgemeiner Fall
  - $W(x_1,\ldots,x_n) = (f_1,\ldots,f_m) =$

$$
\begin{aligned}
(&F_1(punkte(x_1) \cap punkte(x_2) \cap \ldots \cap punkte(x_n)),\\
&F_2(punkte(x_1) \cap punkte(x_2) \cap \ldots \cap punkte(x_n)),\\
&\ldots\\
&F_m(punkte(x_1) \cap punkte(x_2) \cap \ldots \cap punkte(x_n)))
\end{aligned}
$$

  - mit Koordinaten $x_i \in knoten(k_i)$
- Bemerkung
  - Die Schnittmengenbildung berechnet die Menge von Punkten in der Würfelzelle mit Koordinaten $x_1,\ldots,x_n$
  - Jedes Measure wird gesondert aggregiert

Beispiel



(3.1.1999, Wedding, Ersatzt.) = (...)	(1‘1999, Berlin, Skoda) = (...)
(2.1.1999, Wedding, Ersatzt.) = (...)	(4‘1999, Bayern, VW) = (...)

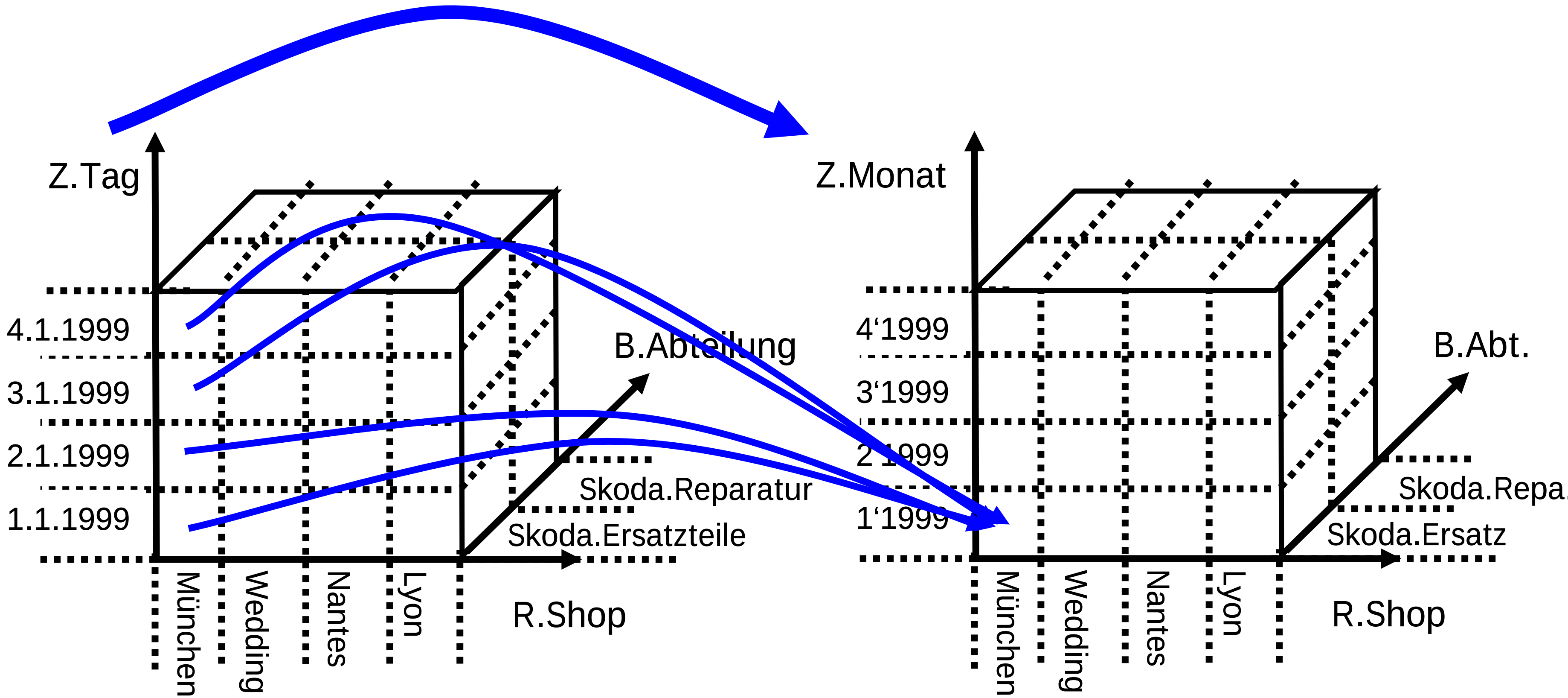
Operationen auf Würfeln

- OLAP Operationen überführen einen Würfel W=(G,F) in einen Würfel W'=(G',F')
- Dabei gilt
  - Aggregation: $G < G'$
  - Verfeinerung $G > G'$
- Eine einfache Operation verändert nur die Klassifikationsstufe einer Dimension in G
  - Komplexe Operationen können auf natürliche Weise aus einfachen Operationen durch Verkettung zusammengesetzt werden
  - Wir betrachten deshalb im folgenden nur einfache Operationen

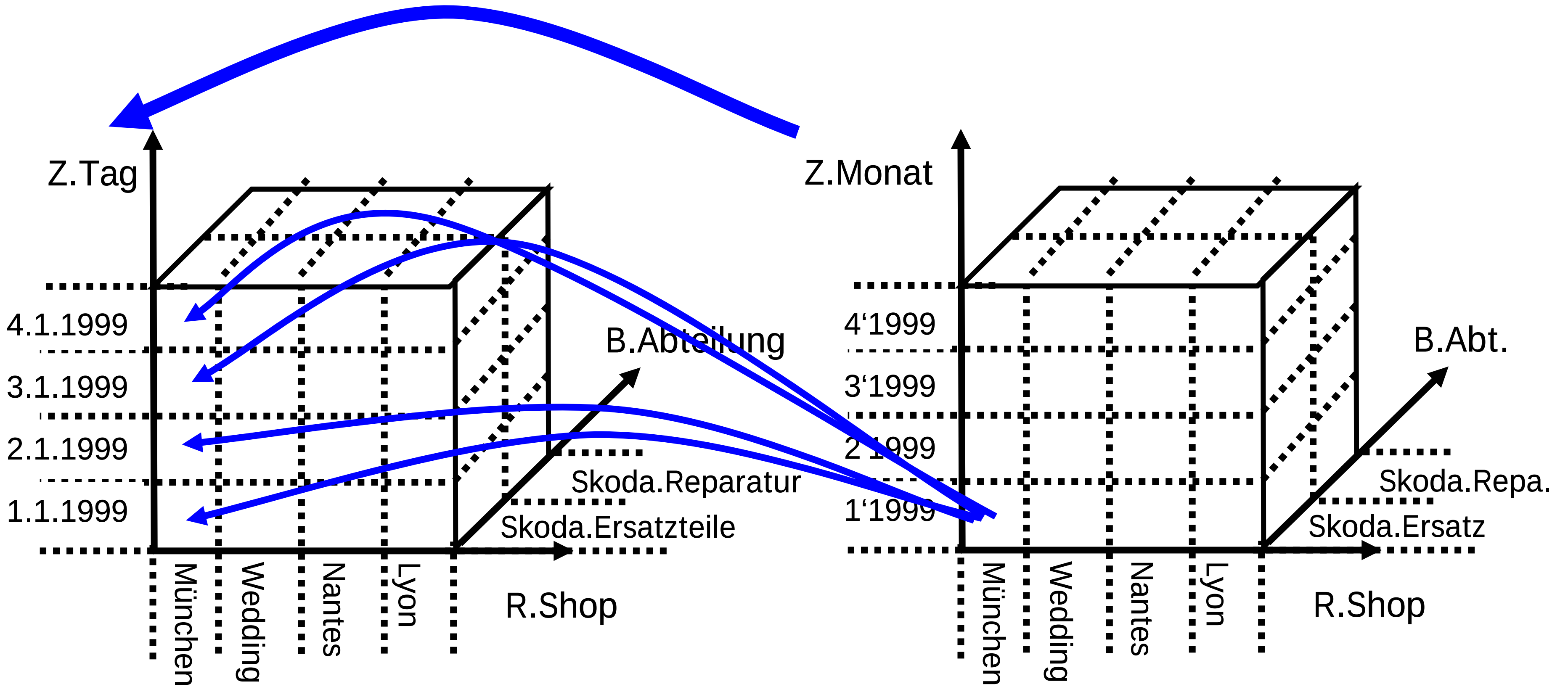
OLAP Operationen

- *Definition.*
  - *Geg.: Würfel $W=(G,F)$ mit $G=(D_1.k_1,\ldots,D_n.k_n)$ und $F=(F_1,..,F_m)$*
  - *Sei $P=\{k_0 \rightarrow \ldots \rightarrow k_{i''} \rightarrow k_i \rightarrow k_{i'} \rightarrow \ldots \rightarrow TOP\}$ ein Pfad in D_i*
  - *Die einfache Aggregation in W entlang P mit Aggregatfunktion f überführt W in*
    - *$W' = ((D_1.k_1,\ldots, D_i.k_{i'},\ldots, D_n.k_n), (F_{1'}, \ldots, F_{m'}))$*
    - *$(F_{1'}, \ldots, F_{m'}) = (F_1(\text{punkte}(k_1) \cap \ldots \text{punkte}(k_{i'}) \cap \ldots \text{punkte}(k_n)),$*
      $\ldots,$
      $F_m(\text{punkte}(k_1) \cap \ldots \text{punkte}(k_{i'}) \cap \ldots \text{punkte}(k_n)))$
  - *Die einfache Verfeinerung in W entlang P mit Aggregatfunktion f überführt W in*
    - *$W' = ((D_1.k_1,\ldots, D_i.k_{i''},\ldots, D_n.k_n), (F_{1''}, \ldots, F_{m''}))$*
    - *Mit $(F_{1''}, \ldots, F_{m''}) = \ldots$*
- Bemerkung
  - Bei Aggregation berechnet man die $F_{i'}$ per Aggregation
  - Bei Verfeinerung ist ein Zugriff auf die Datenbank (Datenpunkte oder Präaggregate) notwendig

Beispiel: Aggregation (Roll-Up)



Beispiel: Verfeinerung (Drill-Down)

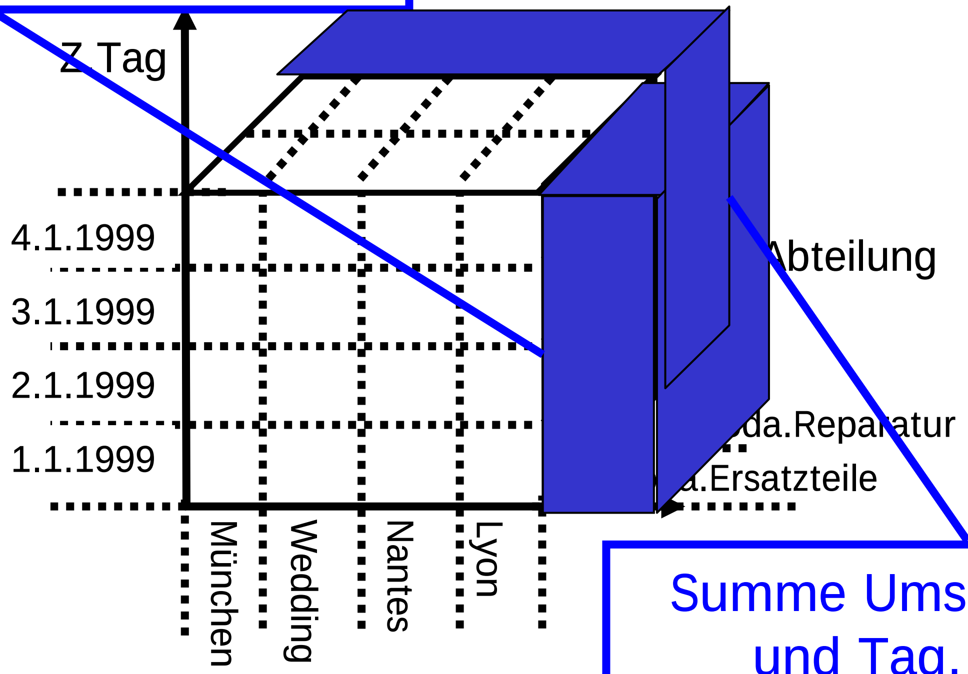


Wo sind wir?

- Das MDDM gibt Klassifikationsstufen, Knoten, Pfade und Dimensionen vor
- Die Daten bestehen aus verschiedenen Measures, jeweils mit Koordinaten in den jeweiligen Dimensionen
- Koordinaten (Dimensionen) sind hierarchisch gegliedert (entlang der Pfade); jeder Datenpunkt hat eine Koordinate in jeder Klassifikationsstufe
  – Die Koordinaten in einem Pfad sind voneinander abhängig
- Würfel stellen aggregierte Daten gemäß der Würfel-granularität dar
- OLAP Operationen (Aggregation, Verfeinerung) sind entlang der Pfade und gemäß der Granularität möglich
  – Sie verändern die Granularität und die aggregierten Fakten

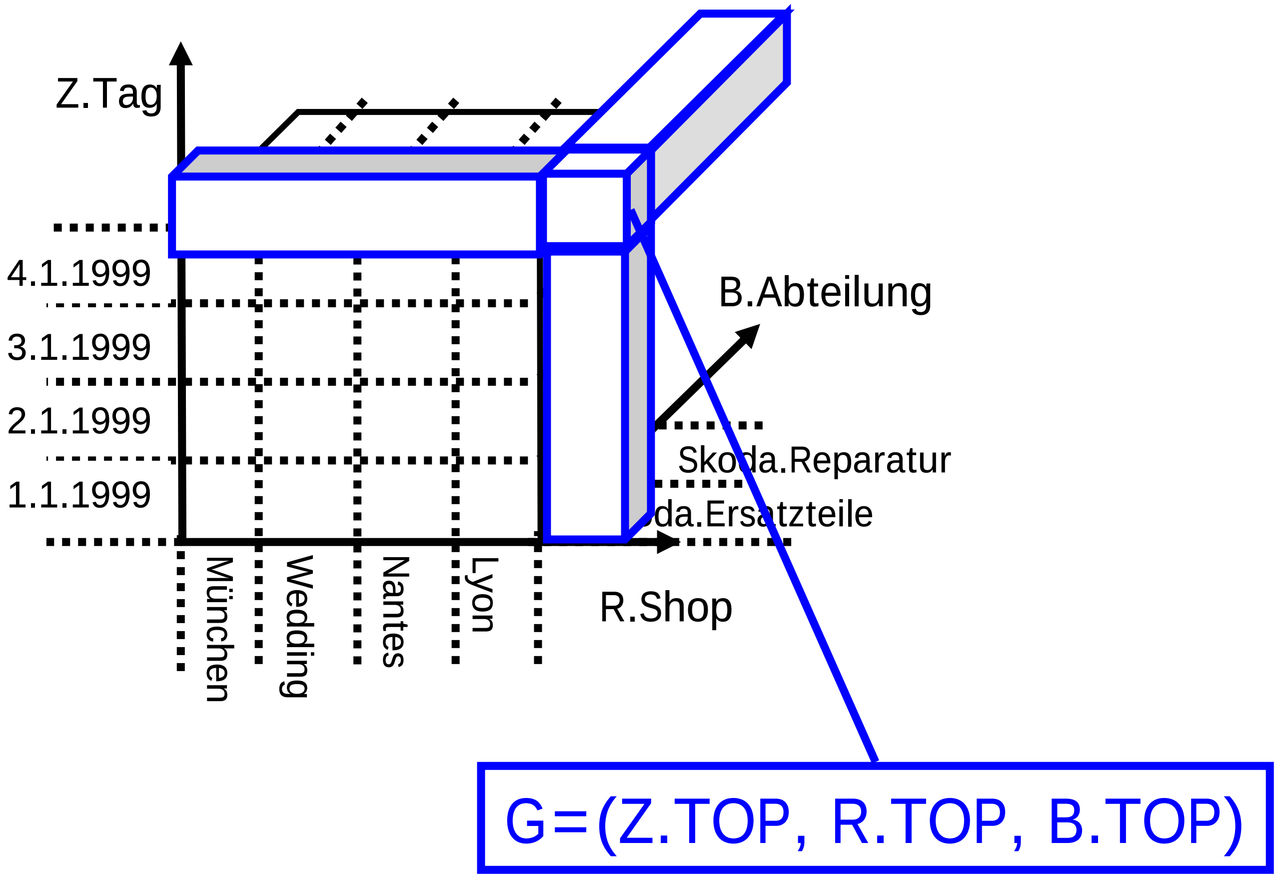
Aggregation bis TOP

Summe Umsatz pro Tag und Abteilungen über alle Shops



Summe Umsatz pro Shop und Tag, über alle Abteilungen

... in mehreren Dimensionen

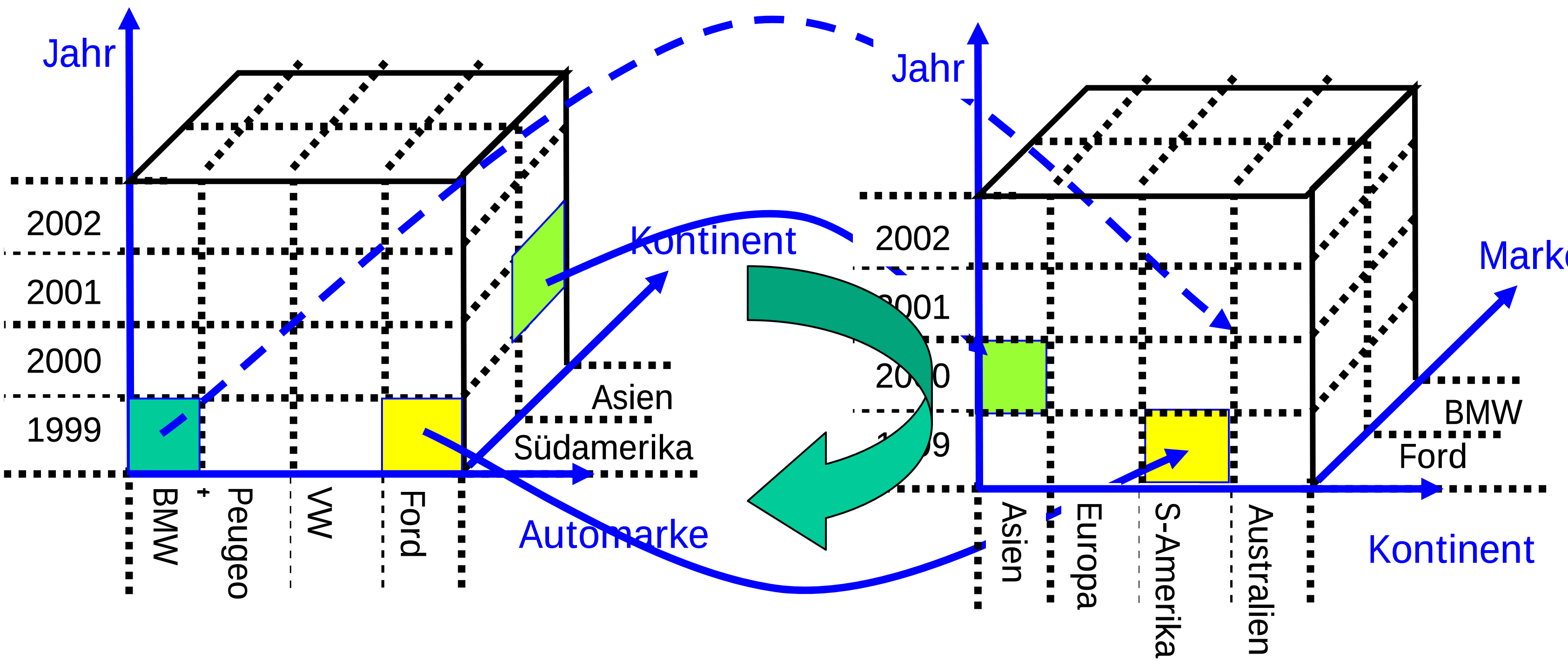


Weitere Operationen

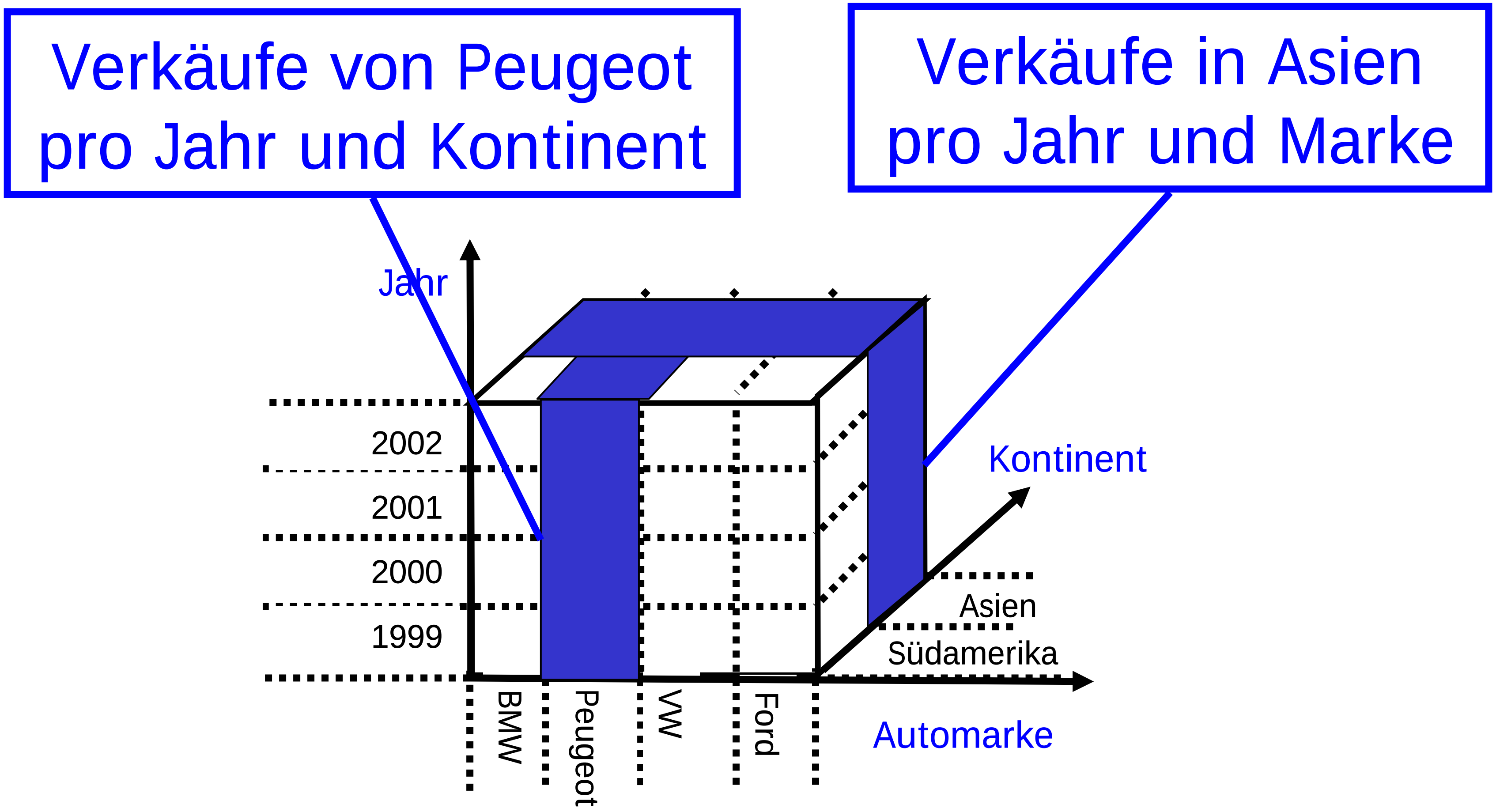
- Aggregation ist die spannendste OLAP Operation
- Die restlichen führen keine Berechnungen durch, sondern selektieren bzw. ändern nur Ausschnitte des Würfels
  - In der Literatur werden teilweise noch weitere Operationen angeführt (Drill-Across, Dicing, ...)

Rotation

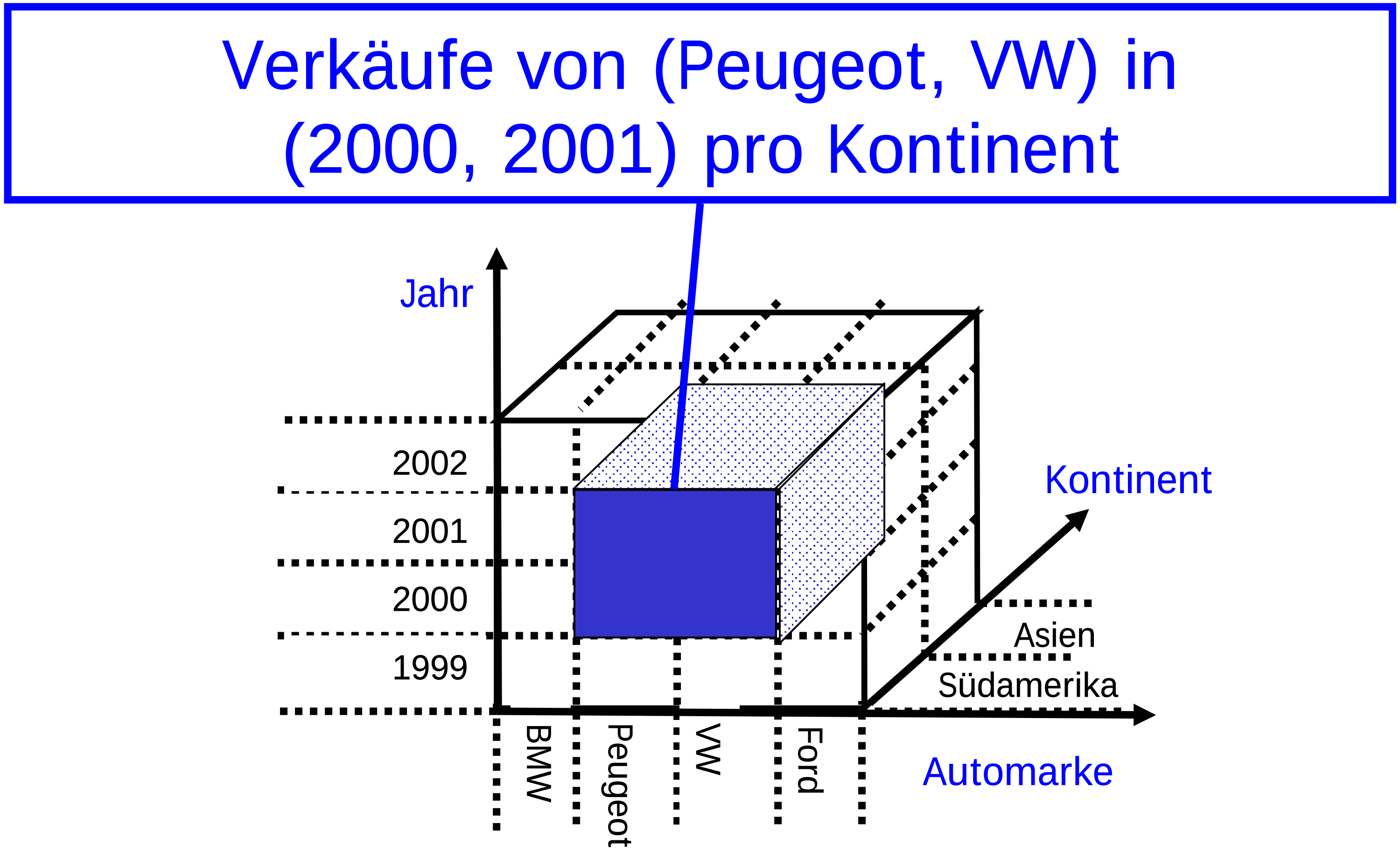
Unterschiedliche Sichtweisen auf einen Datenbestand



Selektion einer Scheibe (Slicing)



Auswahl von Unterwürfeln



2. Grafische Modellierung

- Grafische Modellierung multidimensionaler Daten

Datenbankentwurf

Konzeptionelles Modell → Logisches Modell → Implementierung

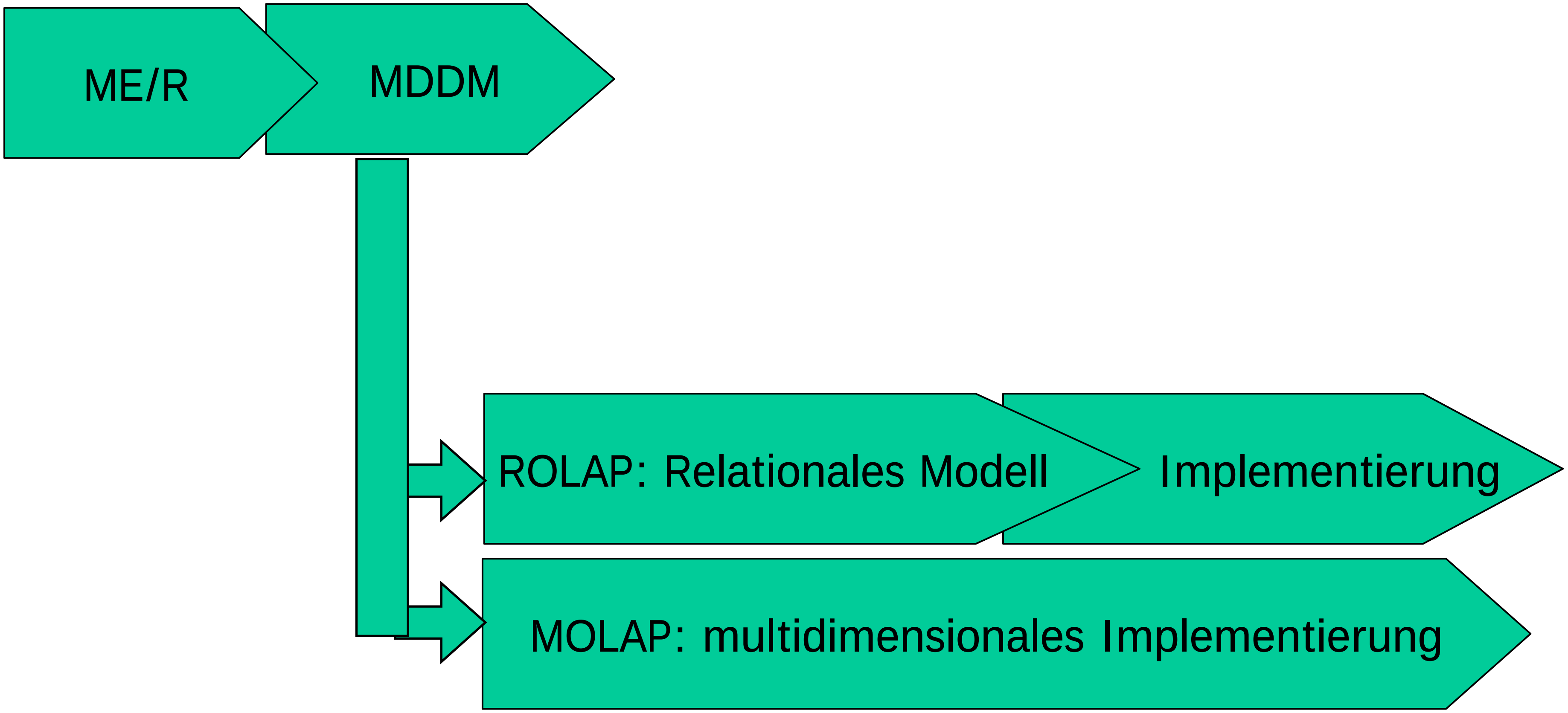
Konzeptionelles Modell	Logisches Modell	Implementierung
Designer, Requirements Engineering	Entwickler, Administrator	Administrator Produktsupport
E/R, UML: Klassen, Attribute, Assoziationen	Relationen, Attribute, Keys, SQL	Indexe, Buffermanagement, Sortierung

ME/R

- Literatur
  - Sapia at al. „Extending the E/R Model for the Multidimensional Paradigm“, Workshop DWDM, 1998
- Logisches MDDM
  - Fakten, Klassifikationsstufen, Dimensionen, ...
- Konzeptionelles/Graphisches Modell für MDDM?
- E/R nicht ausreichend
  - Keine Repräsentation von MDDM Konzepten

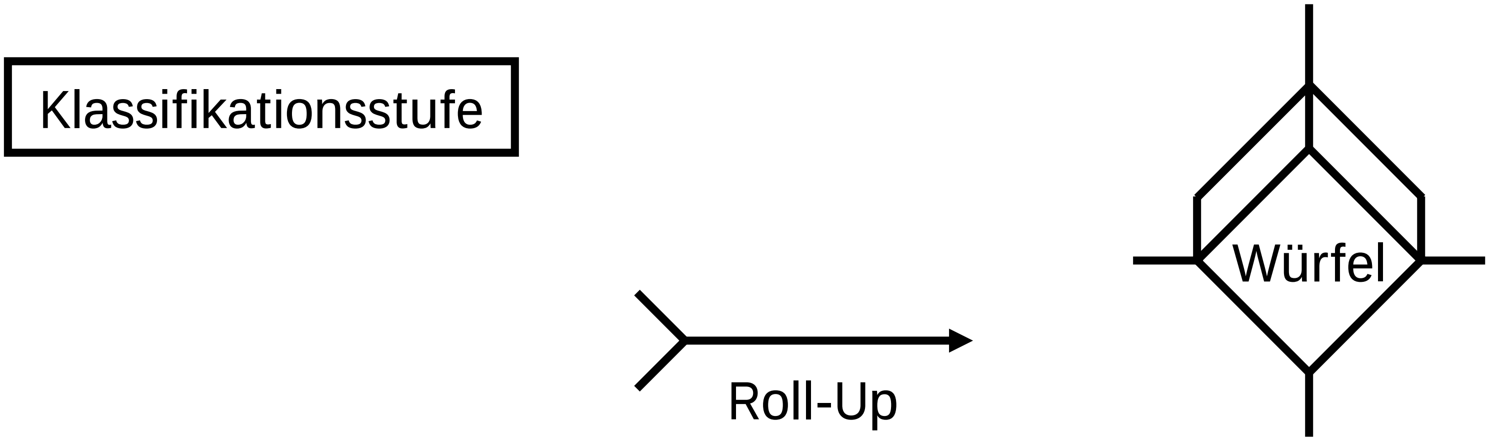
➢ ME/R: Erweitertes E/R Modell

Gesamtbild

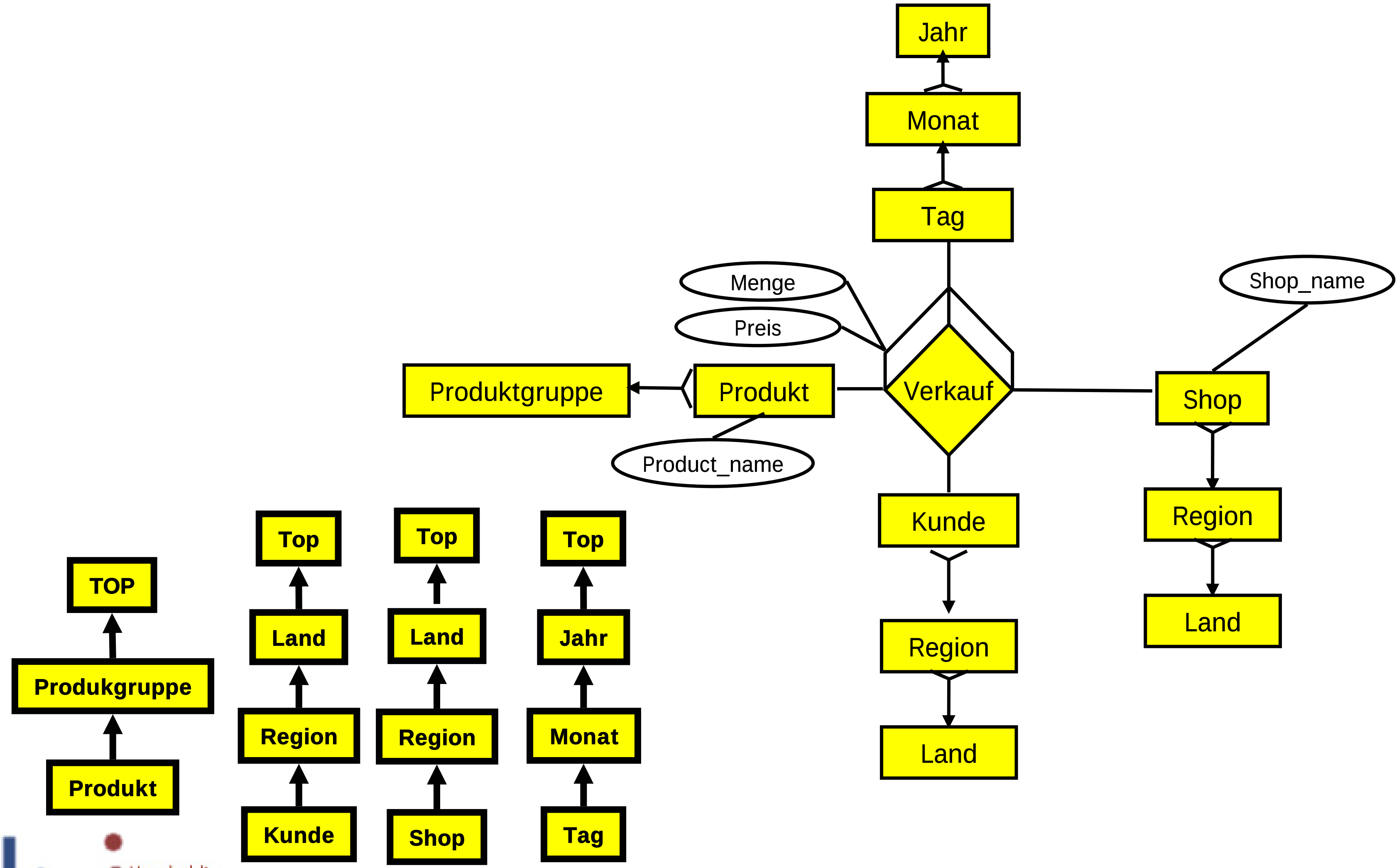

ME/R
MDDM
ROLAP: Relationales Modell
Implementierung
MOLAP: multidimensionales Implementierung

ME/R: Elemente und Notation

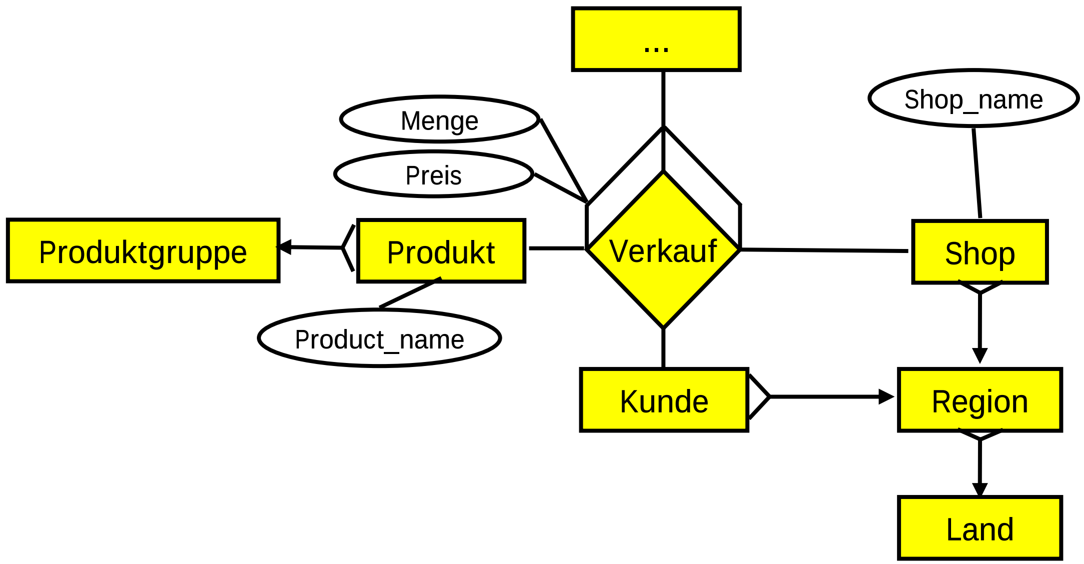
- Neue Elemente
  - Klassifikationsstufe (Dimension Level)
  - Würfel (Fact)
  - Halbordnung zwischen Klassifikationsstufen (Roll-Ups)
  - Constraint: Keine Zyklen in den ROLL-UP Beziehungen



Beispiel


Jahr
Monat
Tag
Menge
Preis
Shop_name
Produktgruppe
Produkt
Verkauf
Shop
Product_name
Kunde
Region
Region
Land
Land
TOP
Produkgruppe
Produkt
Top
Land
Region
Kunde
Top
Land
Region
Shop
Top
Jahr
Monat
Tag

„Wiederverwendung“ von Stufen



- Möglich
  - Fremdschlüssel in Kunde und Shop verweisen auf PK von Region
- ... aber nicht unproblematisch
  - Wenn Kundenregionen (nach Vertrieb) anders organisiert sind als Shopregionen (nach Logistik) – IC schwierig zu überwachen
  - Wenn Kunden in mehr Regionen wohnen dürfen als es Shops gibt – verborgenes Integritätsconstraint
    - Aggregation über Shops bis auf Region gibt viele leere Regionen
  - Drill-Down von Region ist nicht eindeutig

ME/R Bewertung

- Minimale, konservative Erweiterung
- Mehrere Würfel sind möglich
- Flexible Zuordnung Dimensionen – Würfel
- Klare Semantik durch Metamodell in Extended E/R
- Greift eher kurz
  - Keine Eigenschaften von Fakten (Summierbarkeit)
  - Keine Besonderheiten von Klassifikationspfaden
    - Übersprungene Stufen, unendliche Hierarchien
- Keine Übersetzungsmethode definiert
- Toolunterstützung?

Weitere Ansätze

- mUML
  - UML-Erweiterung basierend auf UML Metamodell (Constraints, Stereotypen) -> Werkzeugunterstützung vorhanden
  - Klassifikationsstufe: <<Dimensional class>>
  - Fakten: <<Fakt class>>
  - Würfel: <<Dimension>> (Assoziationstyp)
  - Hierarchie: <<Roll-Up>> (Assoziationstyp)
  - ...
- Multi-dimensional Modelling Language (MML)
- ...
- Keine der Methoden hat sich (bisher) durchgesetzt

3. Summierbarkeit

- Ziel des MDDM
  - Verdichtung von Daten entlang der Klassifikationspfade
- Aber das geht nicht immer gut
  - Numerische versus kategorische Fakten
    - Numerisch: Umsatz, Verkäufe, Messwerte, …
    - Kategorisch: Geschlecht, Kundensegment,…
      - Sollte man als Dimension definieren, muss man aber nicht
  - Nicht alle Aggregatsfunktionen sind hierarchisch anwendbar
    - Summe der Umsätze aller Tage → Umsatz des Monats
      Summe der Umsätze aller Monate → Umsatz des Jahres
    - Durchschnitt der Umsätze aller Tage → Durchschnitt des Monats?
    - Durchschnitt der Umsätze aller Monate → Durchschnitt des Jahres?

Beispiel

Level 0	Level 1	Level 2
1	Summe: 2 Avg: 1	Summe: 12 Avg: 11/2 ≠ 12/3
1		
10	Summe: 10 Avg: 10	

Charakterisierung von Aggregatsfunktionen

- Literatur
  - Lenz, Shoshani: „Summarizability in OLAP and Statistical Databases", SSDBM, 1997
- Geg.: Set X, Partitionierung $(X_1, X_2, \ldots, X_n)$
  - X entspricht einem Klassifikationsknoten
  - $(X_1, X_2, \ldots, X_n)$ sind seine Kinder
- *Definition.*
  - *Eine Aggregatfunktion f ist distributiv gdw*
    - $\exists g: f(X) = f(\, g(X_1), g(X_2), \ldots\, g(X_n))$
  - *Eine Aggregatfunktion f ist algebraisch gdw*
    - *f(X) berechenbar aus fester Menge von g's*
  - *Eine Aggregatfunktion f ist holistisch gdw*
    - *f(X) kann nur aus den Grundelementen von X berechnet werden*
    - D.h., dass die Menge von g's nur durch die Größe von X begrenzt ist
    - Man muss immer auf die Datenpunkte zurück

Beispiele

Distributiv	Summe, Count, Max, Min, ...
Algebraisch	AVG (mit G_1=SUM und G_2=CNT) STDDEV, MaxN, ...
Holistisch	MEDIAN, RANK, PERCENTILE Highest Frequency, ...

- Highest Frequency
  - Merke Werte und jeweilige Frequenz: ((v_1,f_1), (v_2,f_2), ...)
  - Merge zweier Sets möglich
  - Aber: Keine feste Grenze für Platzbedarf, da Anzahl unterschiedlicher Werte nicht fest

Summierbarkeit / Aggregierbarkeit

- Bisher: Wann darf man Werte hierarchisch summieren ?
  – Natürlich nur numerische Fakten
  – Art der Aggregatfunktion beachten
- Aber
  – Summe der Lagerbestände pro Produkt über Jahre?
  – Gesamtsumme Studenten als Summe über Studenten pro Studiengang?
- Weitere notwendige Kriterien für Aggregierbarkeit entlang eines Klassifikationspfades
  – Überlappungsfreiheit der Zuordnung von Klassifikationsknoten
  – Vollständigkeit der Zerlegung pro Klassifikationslevel
  – Typverträglich von Fakt und Aggregatfunktion

Beispiel

Studenten pro Jahr und Studiengang

	1994	1995	1996	Gesamt
Informatik	15	17	13	28
BWL	10	15	11	21
Gesamt	25	31	23	49

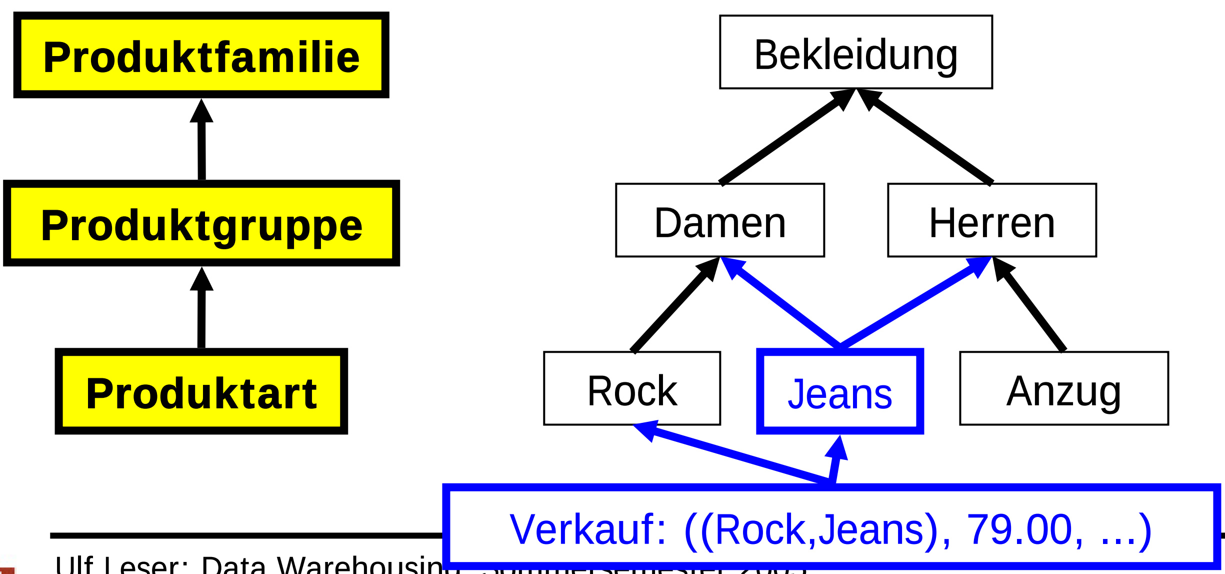
Wie kann das stimmen?

- Wie lange dauern Studiengänge?
- Studenten nur in einem Studiengang eingeschrieben?

Summen lassen sich nicht aus den Einzelwerten berechnen

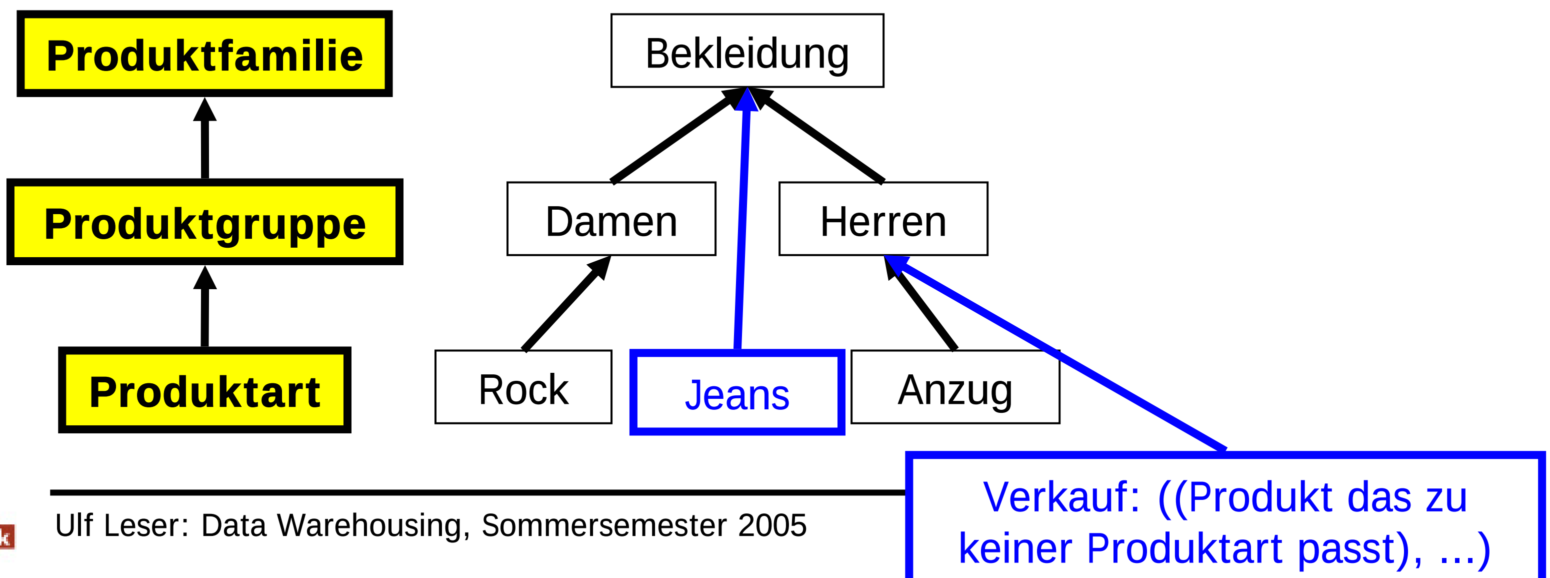
Überlappungsfreiheit

- *Definition.*
  Eine Klassifikationshierarchie ist *überlappungsfrei* *gdw*
  - *Jeder Klassifikationsknoten mit Level i ist höchstens einem Klassifikationsknoten in Level i+1 zugeordnet*
  - *Jeder Datenpunkt ist höchstens einem Klassifikationsknoten mit Level 0 zugeordnet*



Vollständigkeit

- *Definition.*
  Eine Klassifikationshierarchie ist vollständig gdw
  - *Jeder Klassifikationsknoten mit Level i ist mindestens einem Klassifikationsknoten in Level i+1 zugeordnet*
  - *Jedes Fakt ist mindestens einem Klassifikationsknoten mit Level 0 zugeordnet*



Was tun?

- Wir sind bisher immer von vollständigen und überlappungsfreien Klassifikationsschemas ausgegangen
  – Summierbarkeit ist ein wichtiger Grund dafür
- Das entspricht nicht immer der betrieblichen Realität
- Was tun wenn nicht?
  – Neue Klassifikationsknoten
    - Artifizielle Klassifikationsknoten: „Others“, „Rest“, „Nicht zugewiesen“
    - Knotenaufspaltung: „Herrenjeans“, „Damenjeans“
  – Gewichtete Zuordnung
    - Türkei ist 10% Europa, 90% Asien

Typen von Fakten

- Flow (Ereignis zu Zeitpunkt T)
  - Verkäufe, Umsatz, Lieferungen, diplomierte Studenten, …
- Stock (Zustand zu Zeitpunkt T)
  - Lagerbestand, eingeschriebene Studenten, Einwohnerzahlen, …
- Value-per-Unit (Eigenschaft zu Zeitpunkt T)
  - Preis, Herstellungskurs, Währungskurs, …

Typverträglichkeit

	Stock	Flow	Value-per-Unit
MIN/MAX	√	√	√
SUM	Zeit: nein Sonst: √	√	Nie
AVG	√	√	√

- Erkennen von Problemen
  - Nicht automatisch möglich
  - Metadaten – Beschreibung der Measures notwendig

Zusammenfassung

- OLAP Operationen
  - MDDM Modell
    - Modellierung der Daten in der Sprache des Analysten
    - Einschränkung auf sinnvolle Operationen
  - Roll-Up und Drill-Down
  - Selektion und Rotation
- Grafische Notationen sind vorhanden, aber noch nicht verbreitet
- Vorsicht vor Modellierungsfallen
  - Summierung über kategorische Fakten
  - Rekursive Durchschnittsbildung
  - Typunverträgliche Summierung
  - „Löchrige“ Klassifikationshierarchien