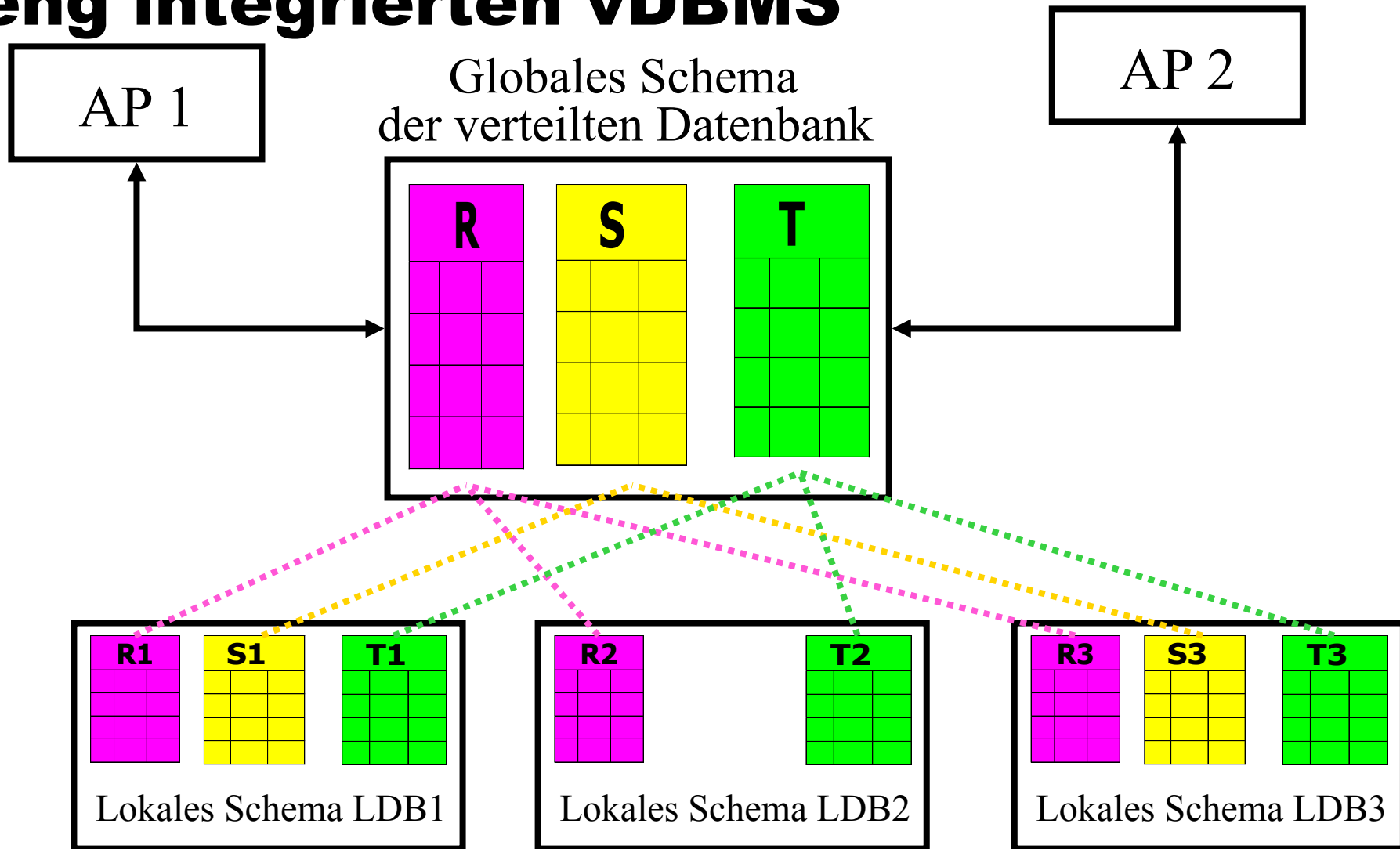




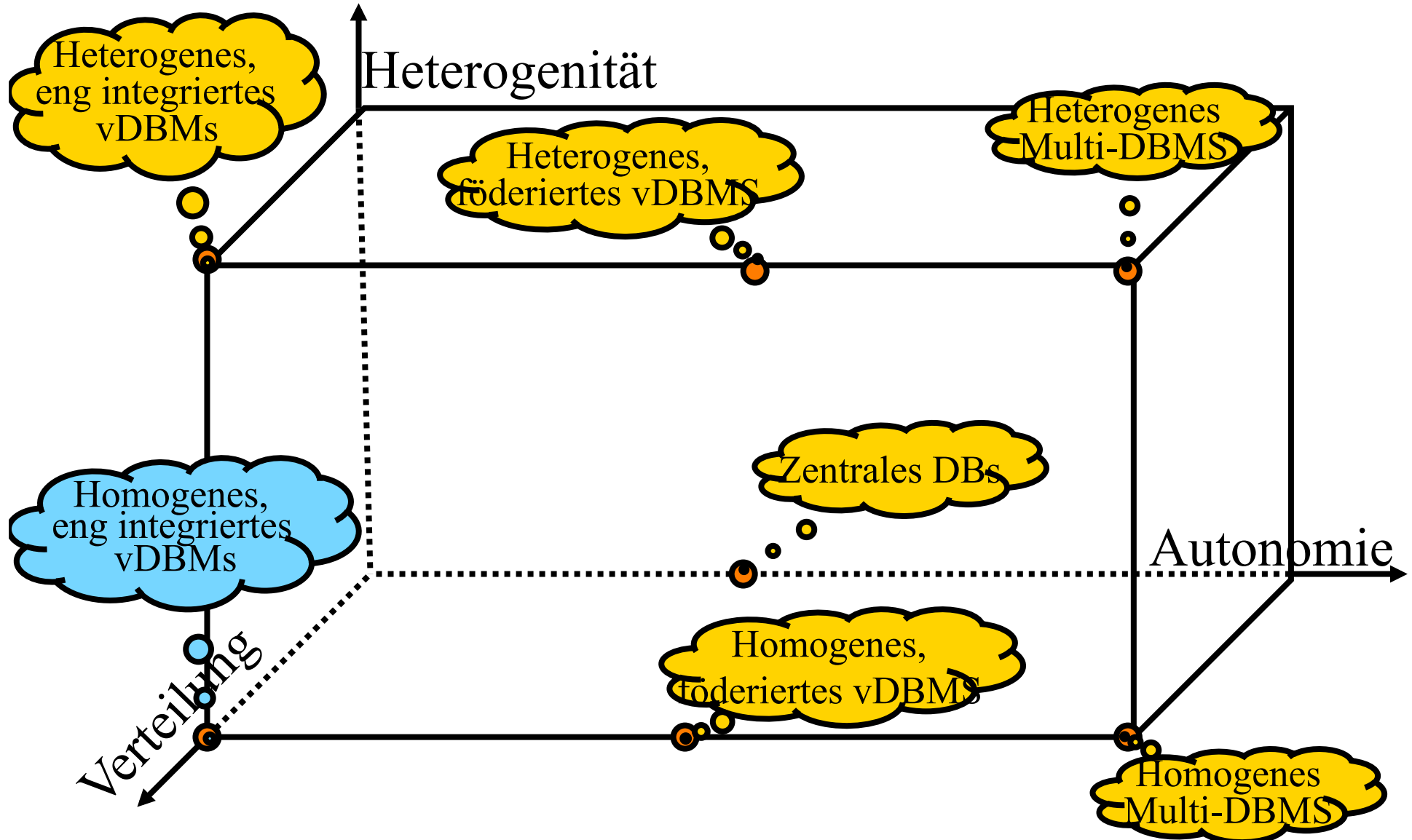
Speicherung globaler Relationen: Fragmentierung und Allokation

**À Priori Schema-Entwurf
für integrierte, homogene
verteilte Datenbanken**

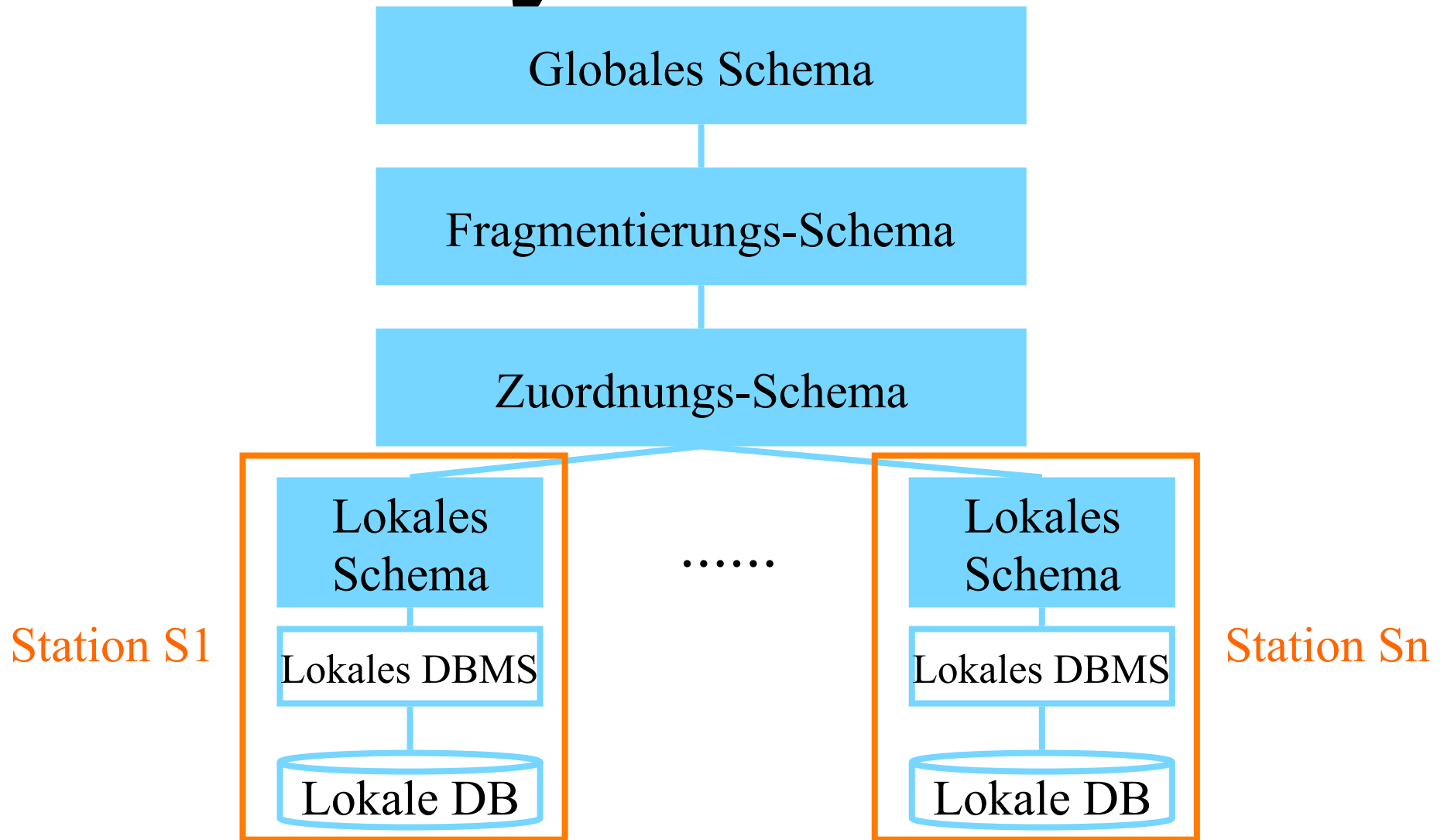
Schema-Integration bei homogenen, eng integrierten vDBMS



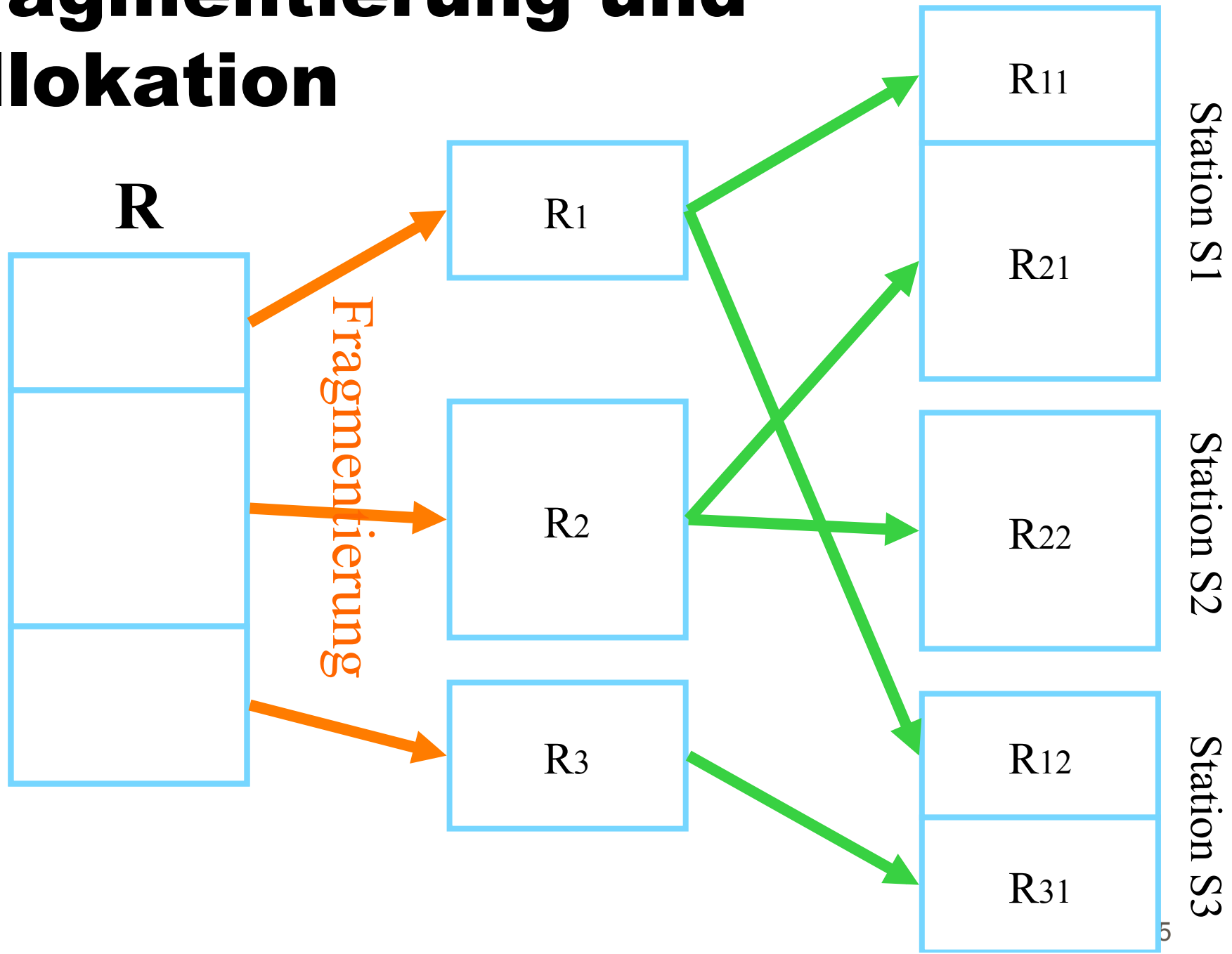
Klassifikation verteilter DBMS

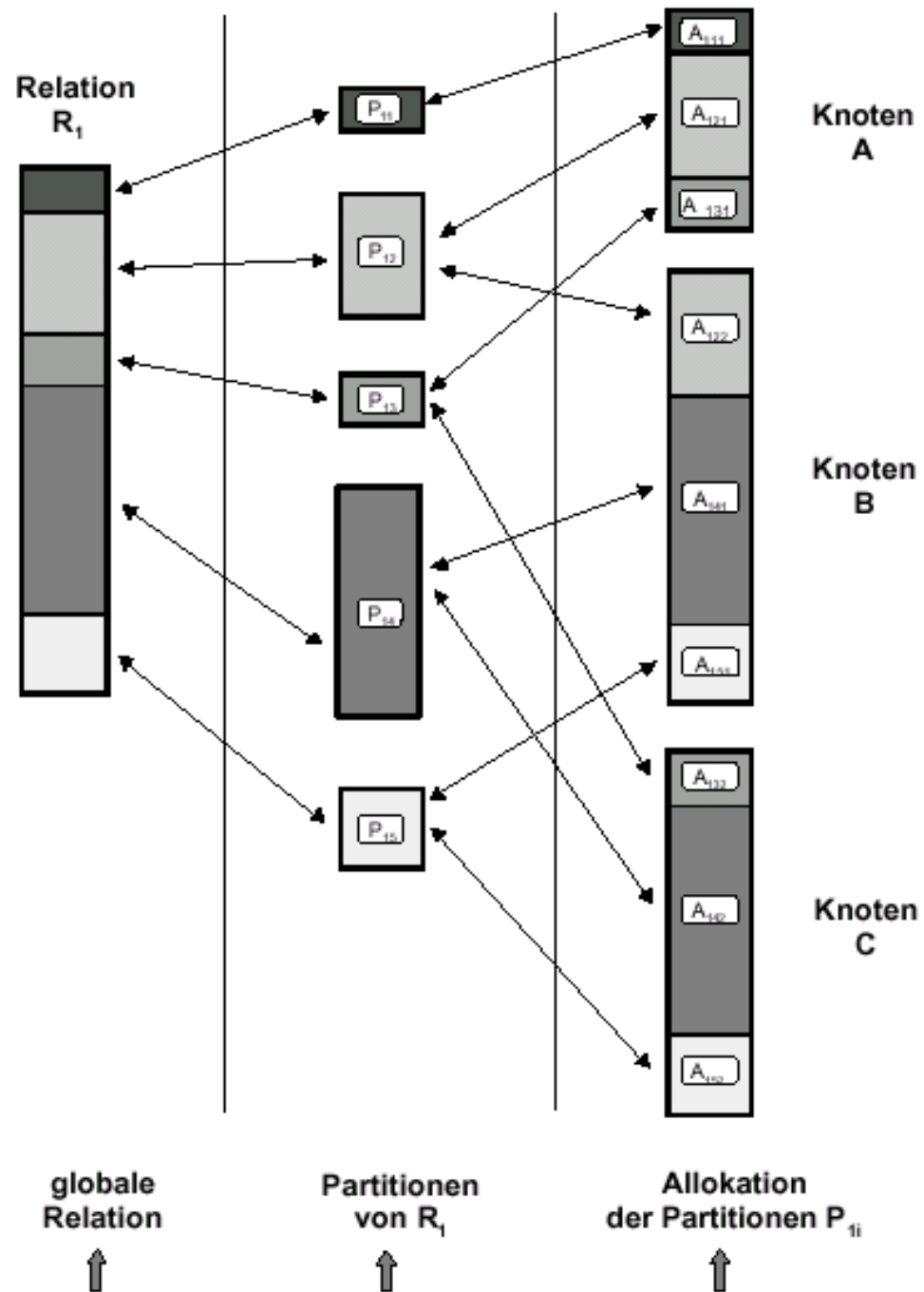


Entwurf eines verteilten Datenbanksystems



Fragmentierung und Allokation





Horizontale Partitionierung

Teile		
TeileNr	TeileBez	...
35181	Schraube M8x13	...
36799	Mutter M8	...
37244	Mutter M7	...
42351	Bolzen B33	...
43749	Schraube M4x6	...
44812	Bolzen B27	...
45438	Schraube M3,5x5	...
...

Horizontale Partitionierung

Teile		
TeileNr	TeileBez	...
35181	Schraube M8x13	...
36799	Mutter M8	...
37244	Mutter M7	...
42351	Bolzen B33	...
43749	Schraube M4x6	...
44812	Bolzen B27	...
45438	Schraube M3,5x5	...
...

Horizontale Partitionierung

$Teile1 := \sigma_{TeileBez \approx "Schraube"} Teile$

$Teile2 := \sigma_{TeileBez \approx "Mutter"} Teile$

$Teile3 := \sigma_{TeileBez \approx "Bolzen"} Teile$

$Teile3 := \sigma_{alles_andere} Teile$

$Teile = \bigcup_i Teile_i$

Teile1		
TeileNr	TeileBez	...
35181	Schraube M8x13	...
43749	Schraube M4x6	...
45438	Schraube M3,5x5	...
...

Teile2		
TeileNr	TeileBez	...
36799	Mutter M8	...
37244	Mutter M7	...
...

Abgeleitete Horizontale Partitionierung

Teile		
TeileNr	TeileBez	...
35181	Schraube M8x13	...
36799	Mutter M8	...
37244	Mutter M7	...
42351	Bolzen B33	...
43749	Schraube M4x6	...
44812	Bolzen B27	...
45438	Schraube M3,5x5	...
...

Lagerort	
TeileNr	Lager
35181	3
36799	1
37244	3
42351	2
43749	2
44812	1
45438	1
...	...

Abgeleitete Horizontale Partitionierung

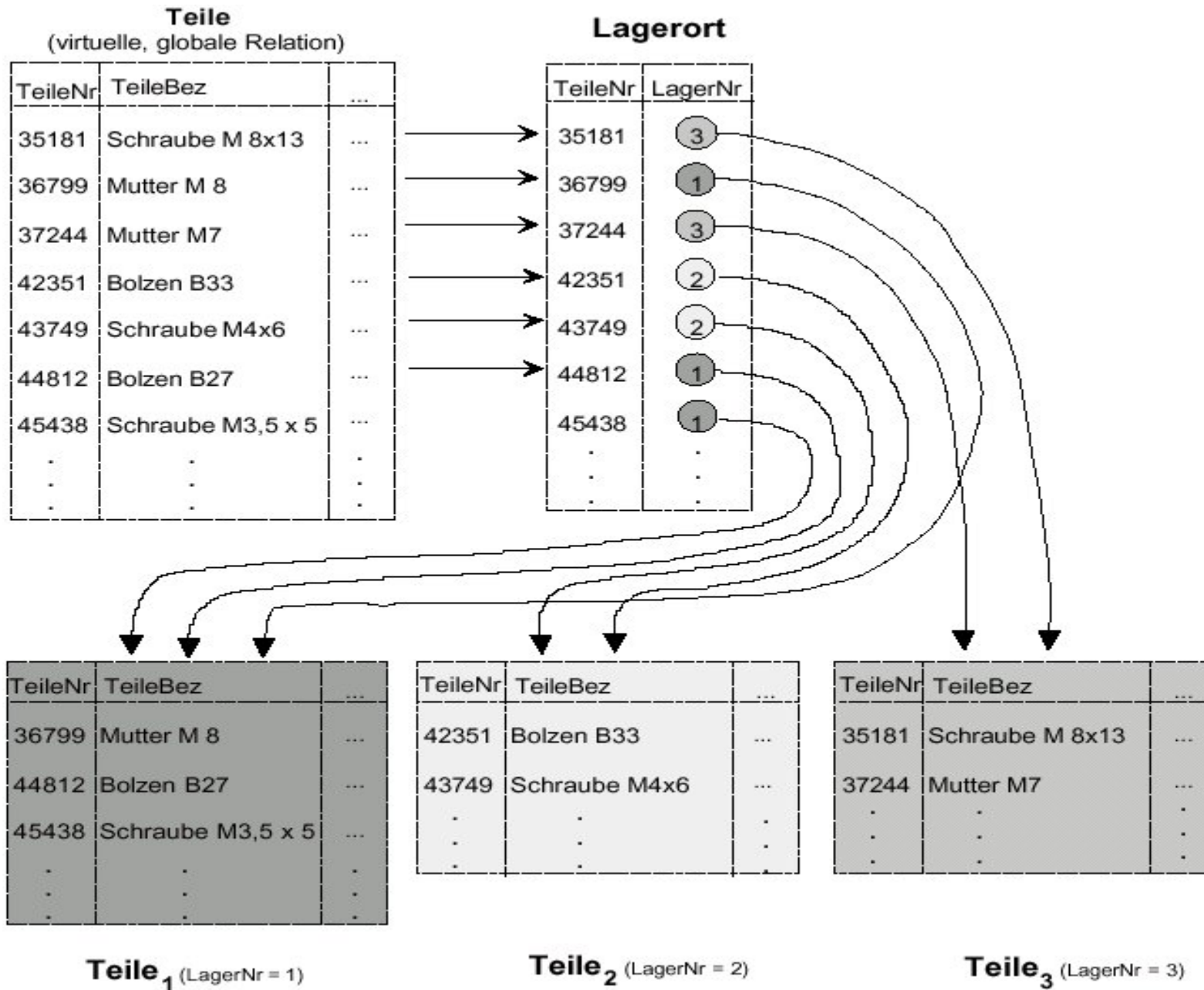
Teile		
TeileNr	TeileBez	...
35181	Schraube M8x13	...
36799	Mutter M8	...
37244	Mutter M7	...
42351	Bolzen B33	...
43749	Schraube M4x6	...
44812	Bolzen B27	...
45438	Schraube M3,5x5	...
...

Lagerort	
TeileNr	Lager
35181	3
36799	1
37244	3
42351	2
43749	2
44812	1
45438	1
...	...

Abgeleitete Horizontale Partitionierung

Teile		
TeileNr	TeileBez	...
35181	Schraube M8x13	...
36799	Mutter M8	...
37244	Mutter M7	...
42351	Bolzen B33	...
43749	Schraube M4x6	...
44812	Bolzen B27	...
45438	Schraube M3,5x5	...
...

Lagerort	
TeileNr	Lager
35181	3
36799	1
37244	3
42351	2
43749	2
44812	1
45438	1
...	...





- Semi-Join von L mit R

\otimes_{C}^{lsj}

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

\times

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat		
A	B	C
a_1	b_1	c_1

- Semi-Join von R mit L

\otimes_{C}^{rsj}

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

\times

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat		
C	D	E
c_1	d_1	e_1

Algebraische Beschreibung der abgeleiteten Partitionierung

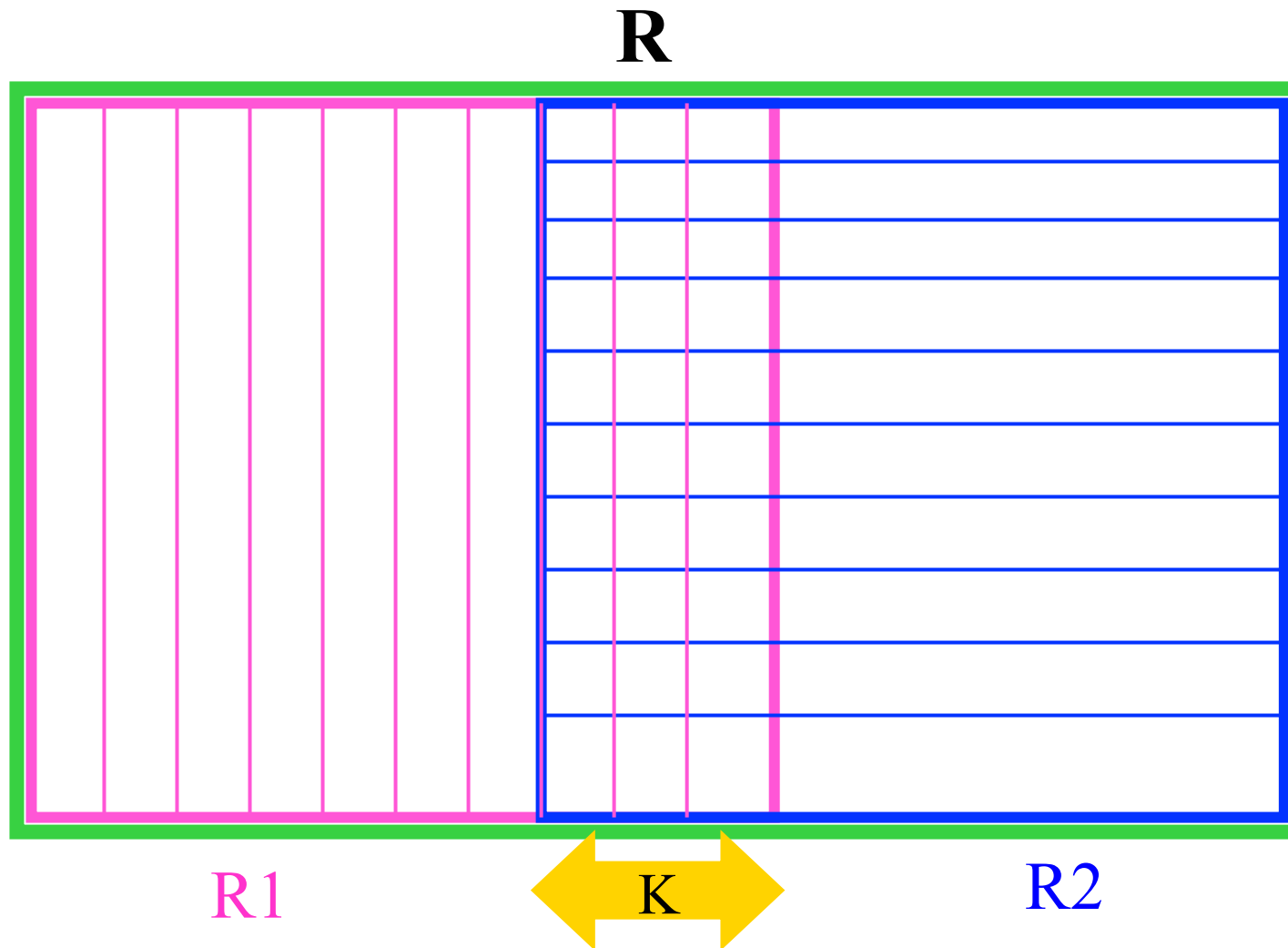
$$Teile1 := Teile \otimes_{TeileNr}^{lsj} \left(\sigma_{LagerNr=1} Lagerort \right)$$

$$Teile2 := Teile \otimes_{TeileNr}^{lsj} \left(\sigma_{LagerNr=2} Lagerort \right)$$

$$Teile3 := Teile \otimes_{TeileNr}^{lsj} \left(\sigma_{LagerNr=3} Lagerort \right)$$

$$Teile4 := Teile \otimes_{TeileNr}^{lsj} \left(\sigma_{LagerNr \notin \{1,2,3\}} Lagerort \right)$$

Vertikale Partitionierung



Verlustlosigkeit und Rekonstruktion von R

$$K = \text{schema}(R_1) \cap \text{schema}(R_2)$$

K muss $\left\{ \begin{array}{l} \text{Primärschlüssel von R sein oder} \\ \text{einen künstlich erzeugten Schlüssel enthalten (Surrogat)} \end{array} \right.$

$$R_1 := \prod_{\text{schema}(R_1)} R$$

$$R_2 := \prod_{\text{schema}(R_2)} R$$

$$R = R_1 \otimes_K R_2$$

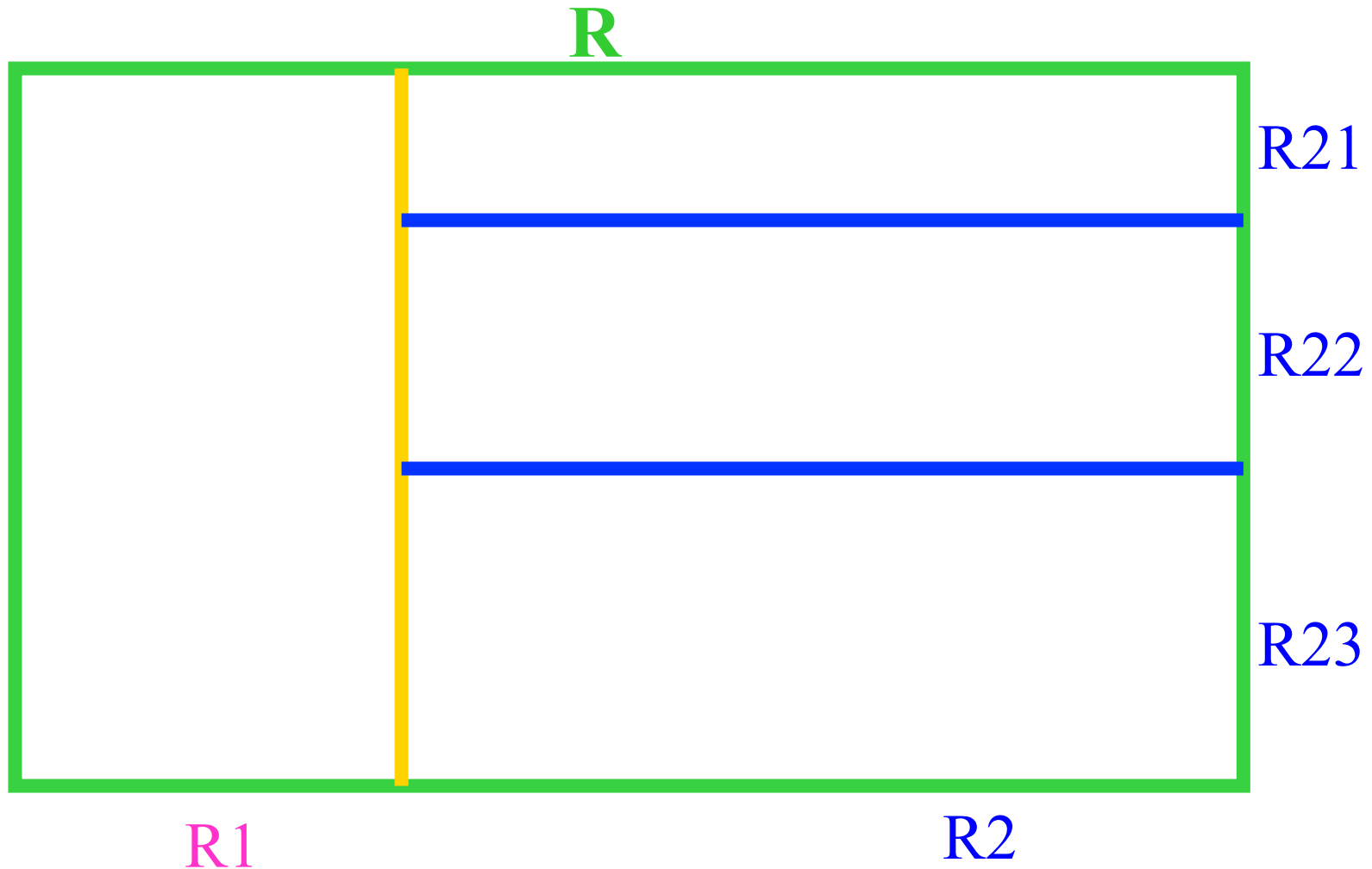
Eigentlich reicht für
die Verlustlosigkeit

$$K \rightarrow \text{schema}(R_1)$$

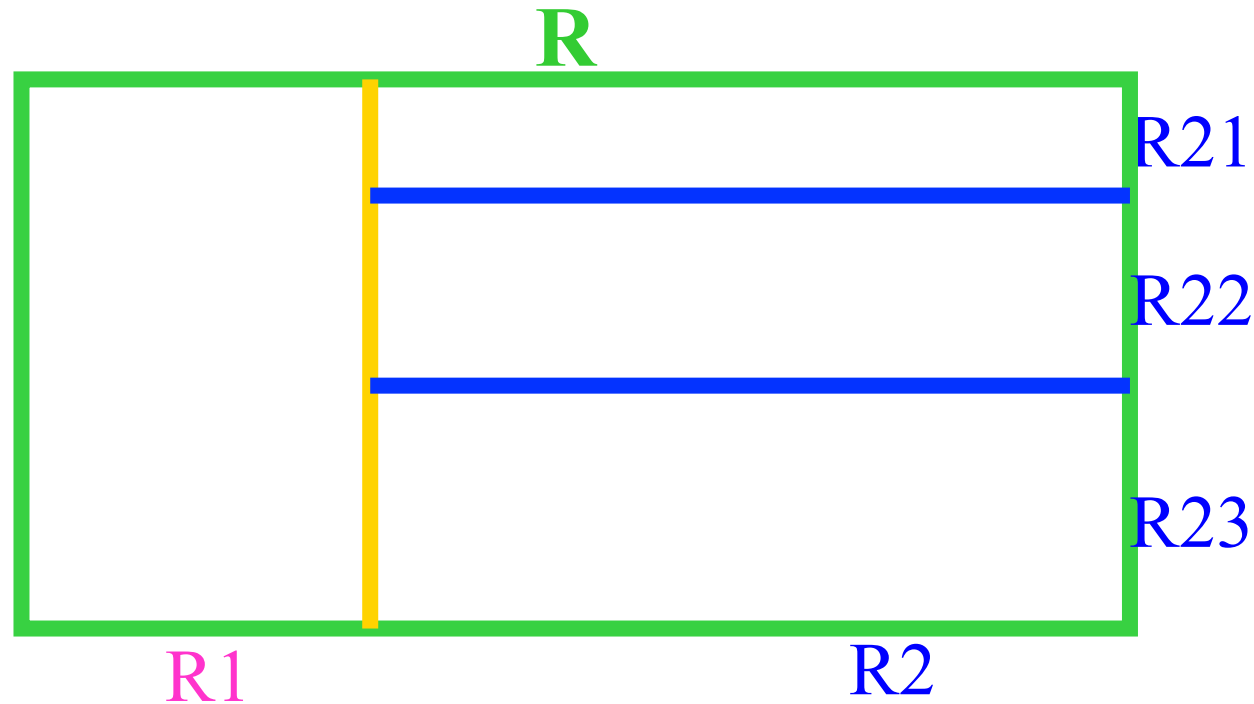
oder

$$K \rightarrow \text{schema}(R_2)$$

Kombinierte Partitionierung: Vertikale und Horizontale Part.

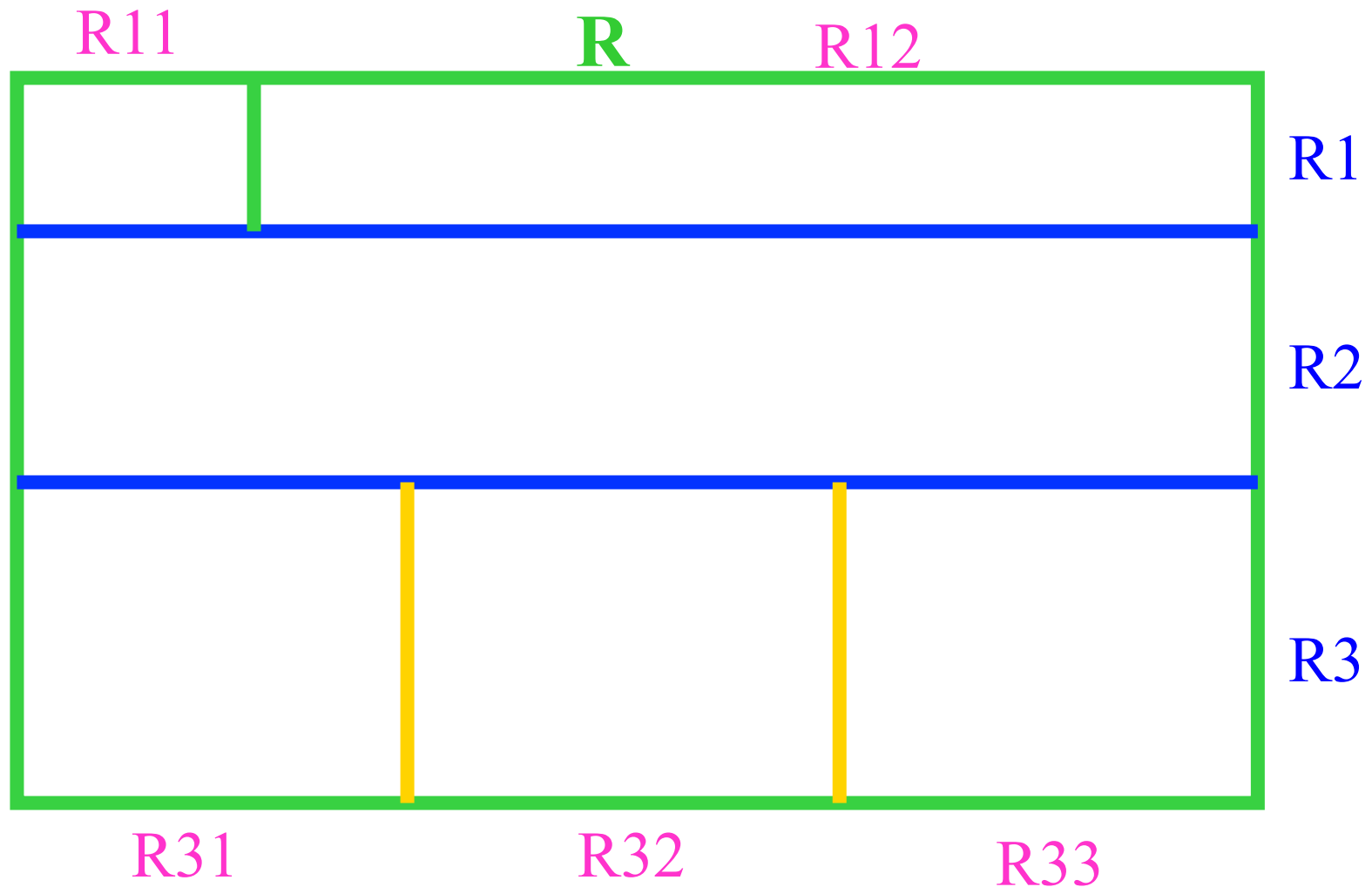


Kombinierte Partitionierung: Vertikale und Horizontale Part.

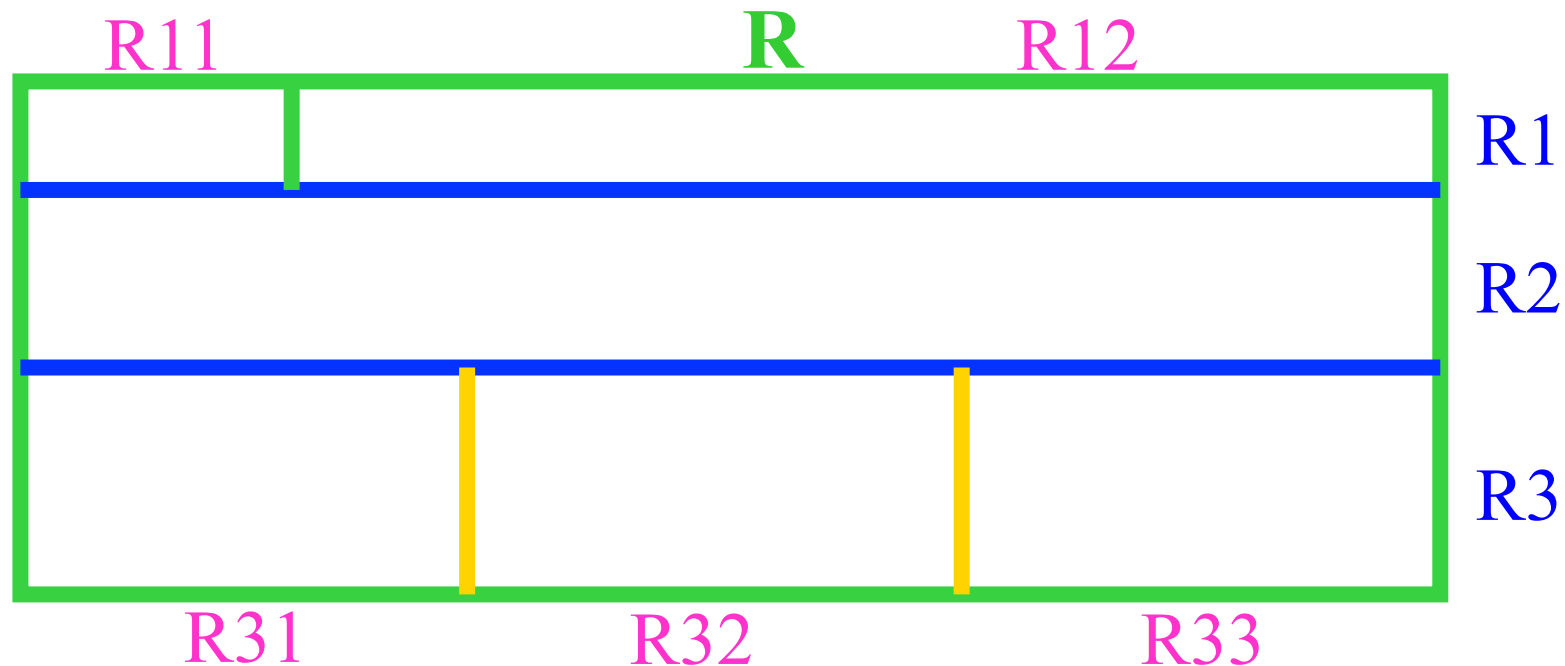


$$R = R_1 \otimes (R_{21} \cup R_{22} \cup R_{23})$$

Kombinierte Partitionierung: Horizontale und Vertikale Part.



Kombinierte Partitionierung: Horizontale und Vertikale Part.



$$R = \left(R_{11} \otimes_{join} R_{12} \right) \cup R_2 \cup \left(R_{31} \otimes R_{32} \otimes R_{33} \right)$$

Korrektheits-Anforderungen

- Rekonstruierbarkeit
- Vollständigkeit
- Disjunktheit

Beispiel-Schema

ANGEST : {[PersNr, AngName, Gehalt, AbtNr, Anschrift]}

ABT : {[AbtNr, AbtName, Bereich, MgrPersNr, Budget]}

TEILE : {[TeileNr, TeileBez, LiefNr, Preis]}

LAGERORT : {[TeileNr, LagerNr]}

LIEFERANT : {[LiefNr, LiefName, Stadt]}

INVENTAR : {[InvNr, Bezeichnung, AnschJahr, AktWert, AbtNr]}

Beispiel-Partitionierung

$$TEILE_1 := \sigma_{0 \leq TeileNr < 300} TEILE$$

$$TEILE_2 := \sigma_{300 \leq TeileNr < 500} TEILE$$

$$TEILE_3 := \sigma_{500 \leq TeileNr < \infty} TEILE$$

$$TEILE = \bigcup_{i \in \{1,2,3\}} TEILE_i$$

$$LIEFERANT_1 := \sigma_{Stadt="Passau"} LIEFERANT$$

$$LIEFERANT_2 := \sigma_{Stadt="München"} LIEFERANT$$

$$LIEFERANT_3 := \sigma_{Stadt \neq "Passau" \wedge Stadt \neq "München"} LIEFERANT$$

$$LIEFERANT = \bigcup_{i \in \{1,2,3\}} LIEFERANT_i$$

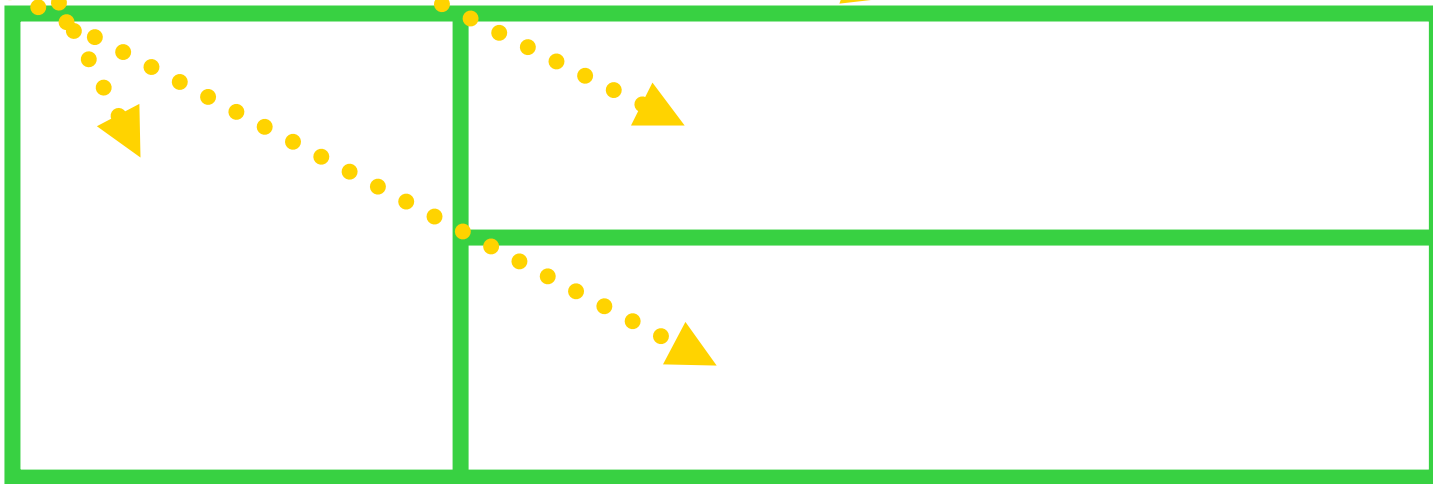
Beispiel: Gemischte Partitionierung

$$ANGEST_{\text{vertraulich}} := \prod_{\{PersNr, Gehalt\}} ANGEST$$

$$ANGEST_{\text{public}} := \prod_{\{PersNr, AngName, MgrPersNr, AbtNr, Anschrift\}} ANGEST$$

$$ANGEST_{\text{public1}} := \sigma_{AbtNr < 300} \left(\prod_{\{PersNr, AngName, MgrPersNr, AbtNr, Anschrift\}} ANGEST \right)$$

$$ANGEST_{\text{public2}} := \sigma_{AbtNr \geq 300} \left(\prod_{\{PersNr, AngName, MgrPersNr, AbtNr, Anschrift\}} ANGEST \right)$$



Beispiel aus dem Kemper/ Eickler-Buch



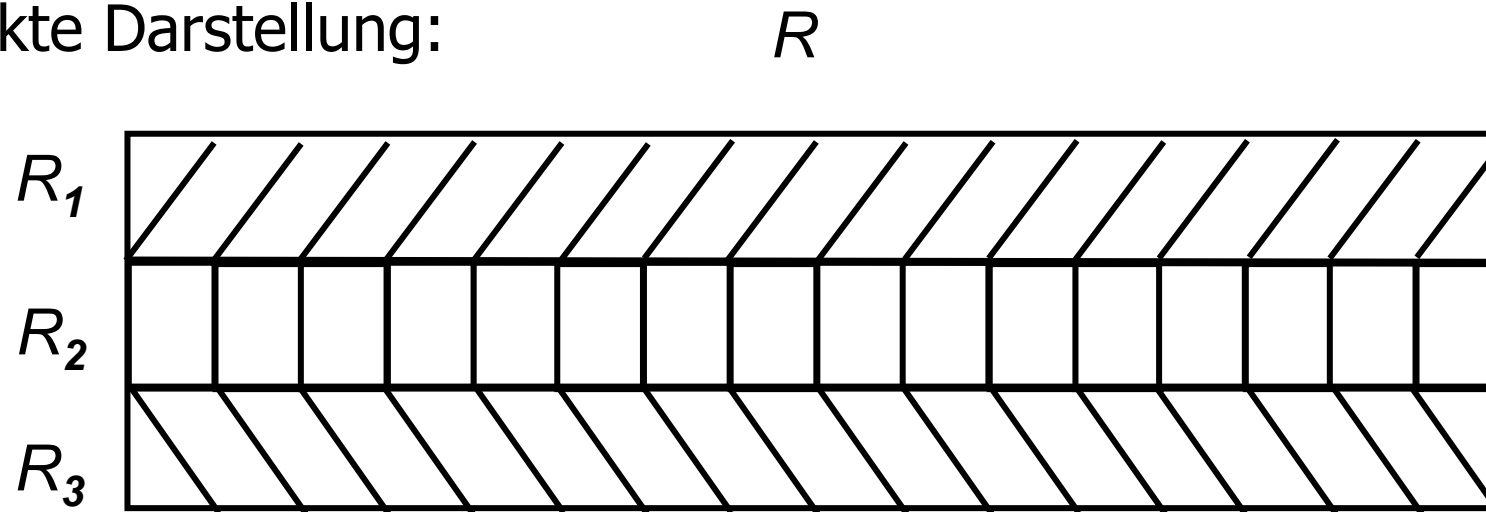
URL: <http://www-db.in.tum.de/research/publications/books/DBMSeinf>

Beispielrelation Professoren

Professoren						
PersNr	Name	Rang	Raum	Fakultät	Gehalt	Steuerkl.
2125	Sokrates	C4	226	Philosophie	85000	1
2126	Russel	C4	232	Philosophie	80000	3
2127	Kopernikus	C3	310	Physik	65000	5
2133	Popper	C3	52	Philosophie	68000	1
2134	Augustinus	C3	309	Theologie	55000	5
2136	Curie	C4	36	Physik	95000	3
2137	Kant	C4	7	Philosophie	98000	1

Horizontale Fragmentierung

abstrakte Darstellung:



Für 2 Prädikate p_1 und p_2 ergeben sich 4 Zerlegungen:

$$R1 := \sigma_{p_1 \wedge p_2}(R)$$

$$R2 := \sigma_{p_1 \wedge \neg p_2}(R)$$

$$R3 := \sigma_{\neg p_1 \wedge p_2}(R)$$

$$R4 := \sigma_{\neg p_1 \wedge \neg p_2}(R)$$



n Zerlegungsprädikate p_1, \dots, p_n ergeben 2^n Fragmente

sinnvolle Gruppierung der Professoren nach Fakultätszugehörigkeit:

→ 3 Zerlegungsprädikate:

$p_1 \equiv \text{Fakultät} = \text{„Theologie“}$

$p_2 \equiv \text{Fakultät} = \text{„Physik“}$

$p_3 \equiv \text{Fakultät} = \text{„Philosophie“}$

$\text{TheolProfs}' := \sigma_{p_1 \wedge \neg p_2 \wedge \neg p_3}(\text{Professoren}) = \sigma_{p_1}(\text{Professoren})$

$\text{PhysikProfs}' := \sigma_{\neg p_1 \wedge p_2 \wedge \neg p_3}(\text{Professoren}) = \sigma_{p_2}(\text{Professoren})$

$\text{PhiloProfs}' := \sigma_{\neg p_1 \wedge \neg p_2 \wedge p_3}(\text{Professoren}) = \sigma_{p_3}(\text{Professoren})$

$\text{AndereProfs}' := \sigma_{\neg p_1 \wedge \neg p_2 \wedge \neg p_3}(\text{Professoren})$

Abgeleitete horizontale Fragmentierung

Beispiel *Vorlesungen* aus dem Universitätsschema:
Zerlegung in Gruppen mit gleicher SWS-Zahl

2SWSVorls := $\sigma_{\mathbf{sWS}=2}$ (Vorlesungen)

3SWSVorls := $\sigma_{\mathbf{sWS}=3}$ (Vorlesungen)

4SWSVorls := $\sigma_{\mathbf{sWS}=4}$ (Vorlesungen)

 für Anfragebearbeitung schlechte Zerlegung

```

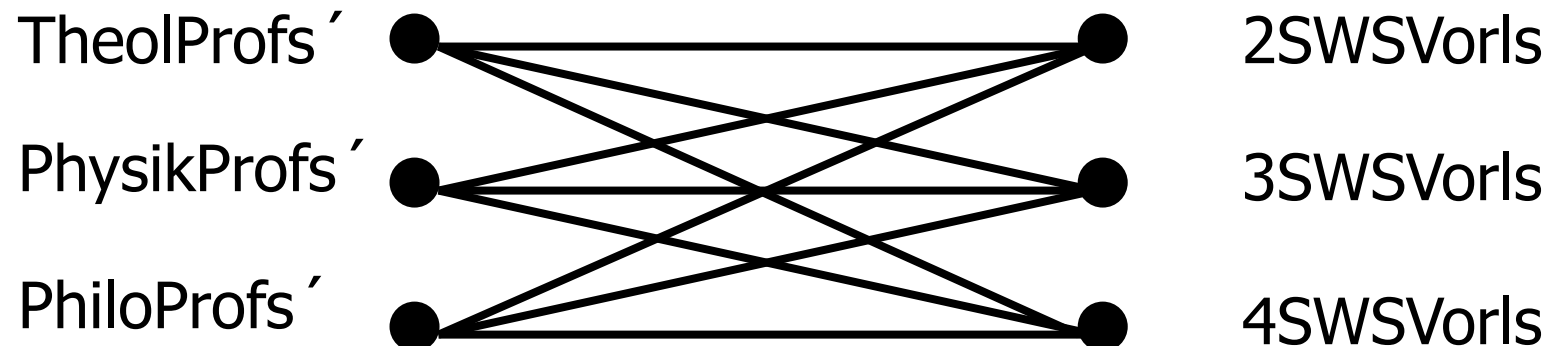
select Titel, Name
from Vorlesungen, Professoren
where gelesenVon = PersNr;

```

resultiert in:

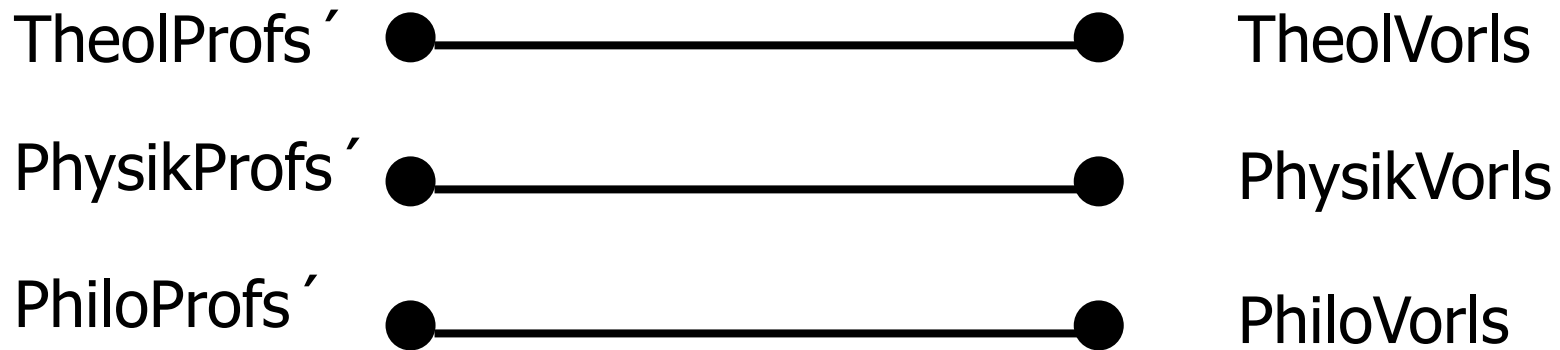
$$\Pi_{\text{Titel, Name}}((\text{TheolProfs}' \bowtie \text{2SWSVorls}) \cup (\text{TheolProfs}' \bowtie \text{3SWSVorls}) \cup \dots \cup (\text{PhiloProfs}' \bowtie \text{4SWSVorls}))$$

Join-Graph zu diesem Problem:





Lösung: abgeleitete Fragmentierung



TheolVorls := Vorlesungen $\bowtie_{\text{gelesenVon=PersNr}}$ TheolProfs'
PhysikVorls := Vorlesungen $\bowtie_{\text{gelesenVon=PersNr}}$ PhysikProfs'
PhiloVorls := Vorlesungen $\bowtie_{\text{gelesenVon=PersNr}}$ PhiloProfs'

$\Pi_{\text{Titel, Name}}((\text{TheolProfs}' \bowtie_p \text{TheolVorls}) \cup$
 $(\text{PhysikProfs}' \bowtie_p \text{PhysikVorls}) \cup$
 $(\text{PhiloProfs}' \bowtie_p \text{PhiloVorls}))$

mit $p \equiv (\text{PersNr} = \text{gelesenVon})$

Algebraische Äquivalenzen

$$S_i = S \bowtie_p R_i \quad \text{mit} \quad S = S_1 \cup \dots \cup S_n$$

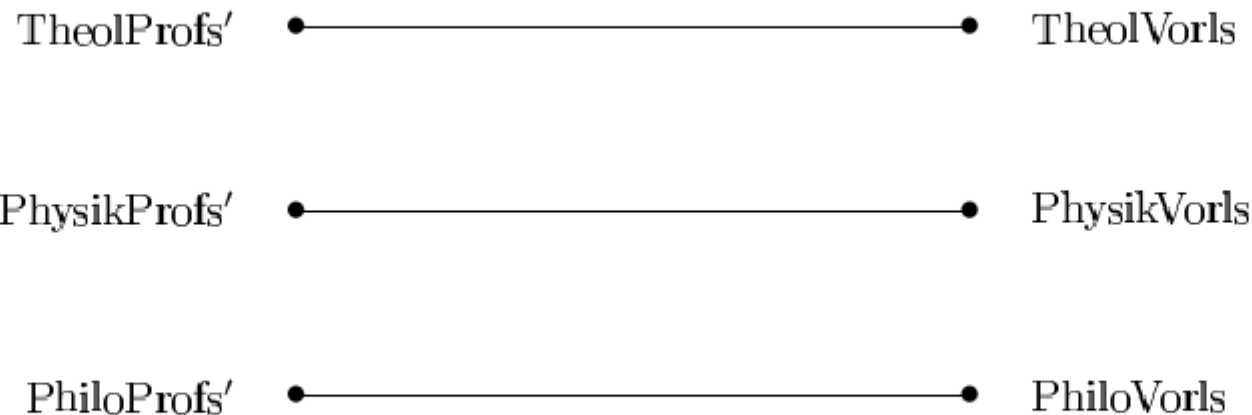
$$R_i \bowtie_p S_j = \emptyset \quad \text{für} \quad i \neq j.$$

$$(R_1 \cup \dots \cup R_n) \bowtie_p (S_1 \cup \dots \cup S_n) = (R_1 \bowtie_p S_1) \cup (R_2 \bowtie_p S_2) \cup \dots \cup (R_n \bowtie_p S_n)$$

Beispiel

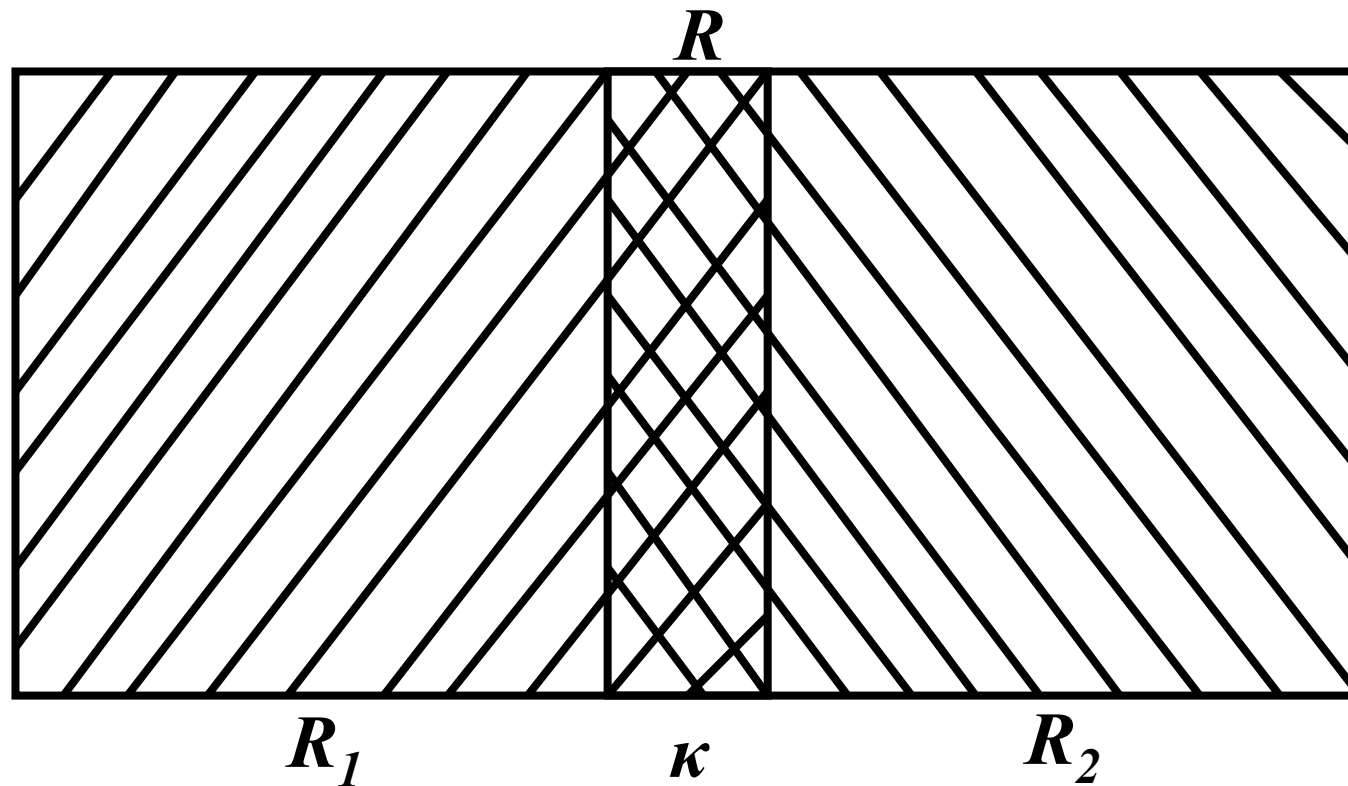
$$\begin{aligned} & (\text{TheolVorls} \cup \text{PhysikVorls} \cup \text{PhiloVorls}) \bowtie \dots \\ & (\text{TheolProfs} \cup \text{PhysikProfs} \cup \text{PhiloProfs}) \end{aligned}$$

es reichen folgende Joins



Vertikale Fragmentierung

abstrakte Darstellung:



Vertikale Fragmentierung

Beliebige vertikale Fragmentierung gewährleistet **keine Rekonstruierbarkeit**

2 mögliche Ansätze, um Rekonstruierbarkeit zu garantieren:

- jedes Fragment enthält den Primärschlüssel der Originalrelation. Aber: Verletzung der *Disjunktheit*
- jedem Tupel der Originalrelation wird ein eindeutiges **Surrogat** (= künstlich erzeugter Objektindikator) zugeordnet, welches in jedes vertikale Fragment des Tupels mit aufgenommen wird

Vertikale Fragmentierung (Beispiel)

für die Universitätsverwaltung sind PersNr, Name, Gehalt und Steuerklasse interessant:

ProfVerw := Π **PersNr, Name, Gehalt, Steuerklasse** (Professoren)

für Lehre und Forschung sind dagegen PersNr, Name, Rang, Raum und Fakultät von Bedeutung:

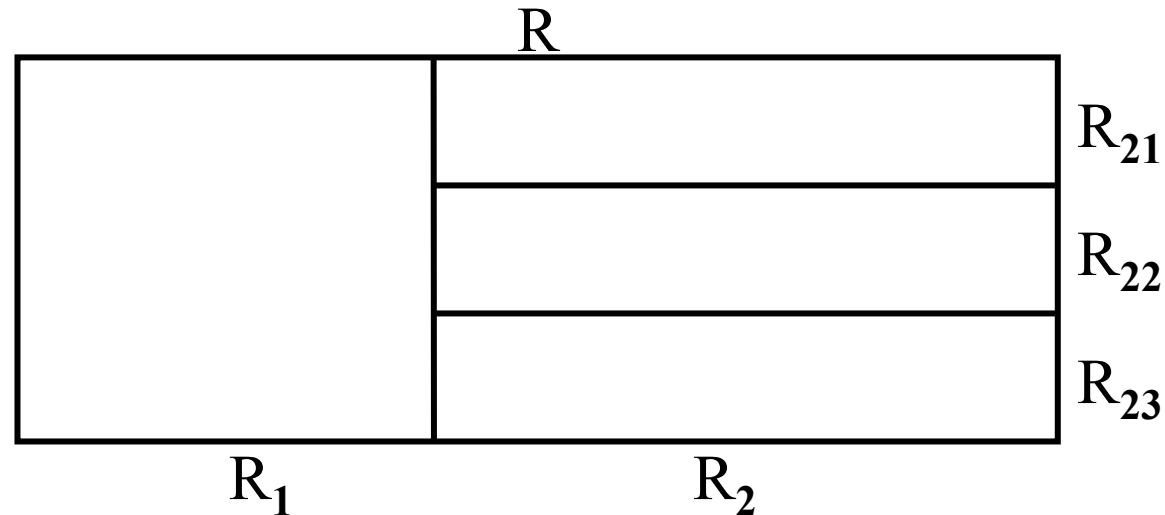
Profs := Π **PersNr, Name, Rang, Raum, Fakultät** (Professoren)

Rekonstruktion der Originalrelation *Professoren*:

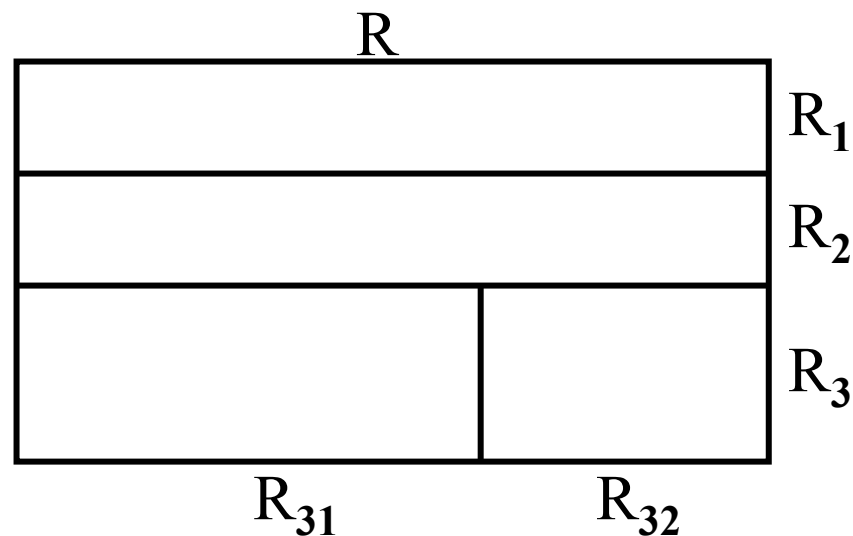
Professoren = ProfVerw \bowtie **ProfVerw.PersNr = Profs.PersNr** Profs

Kombinierte Fragmentierung

a) horizontale Fragmentierung nach vertikaler Fragmentierung



b) vertikale Fragmentierung nach horizontaler Fragmentierung



Rekonstruktion nach kombinierter Fragmentierung

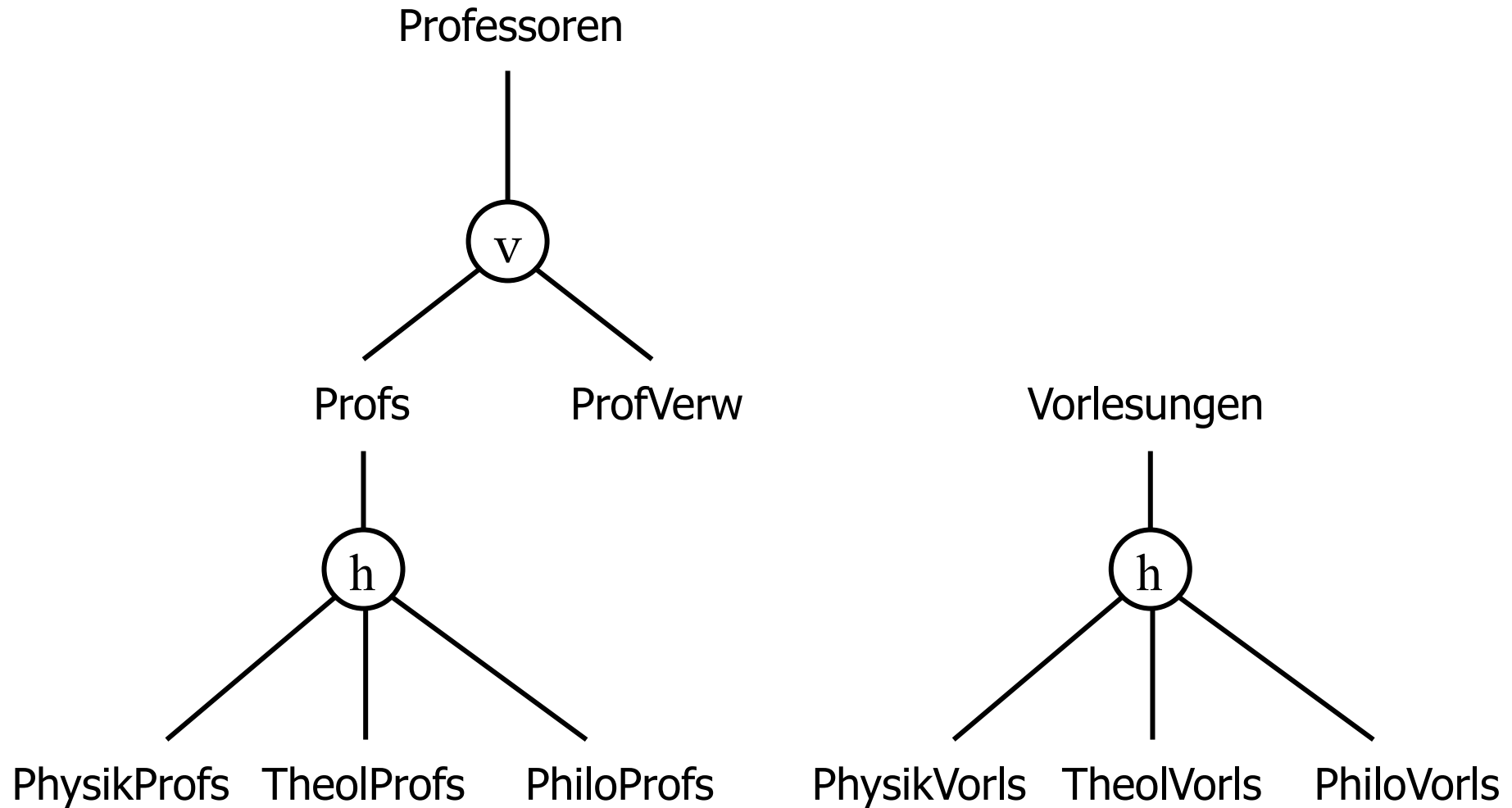
Fall a)

$$R = R_1 \bowtie_p (R_{21} \cup R_{22} \cup R_{23})$$

Fall b)

$$R = R_1 \cup R_2 \cup (R_{31} \bowtie_{R_{31} \cdot \kappa = R_{32} \cdot \kappa} R_{32})$$

Baumdarstellung der Fragmentierungen (Beispiel)



Allokation

- Dasselbe Fragment kann mehreren Stationen zugeordnet werden
- Allokation für unser Beispiel ohne Replikationen ⇒ **redundanzfreie** Zuordnung

Station	Bemerkung	zugeordnete Fragmente
S_{Verw}	Verwaltungsrechner	$\{ProfVerw\}$
S_{Physik}	Dekanat Physik	$\{PhysikVorls, PhysikProfs\}$
S_{Philo}	Dekanat Philosophie	$\{PhiloVorls, PhiloProfs\}$
S_{Theol}	Dekanat Theologie	$\{TheolVorls, TheolProfs\}$

Transparenz in verteilten Datenbanken

- Grad der Unabhängigkeit den ein VDBMS dem Benutzer beim Zugriff auf verteilte Daten vermittelt
- verschiedene Stufen der Transparenz:
 - ◆ Fragmentierungstransparenz
 - ◆ Allokationstransparenz
 - ◆ Lokale Schema-Transparenz

„Monitoring“ zur Bestimmung einer guten Fragmentierungen

- Welche Anwendungen greifen in welcher Form (lesend/schreibend)
- wie häufig
- mit welchen Auswahl/Selektionsprädikaten
- auf welche Relationen und Attribute zu
- und welche Datenmengen müssen dabei übertragen werden

Systematische Bestimmung horizontaler Partitionen

- Sei $R: \{[A_1, A_2, \dots, A_n]\}$ eine Relation
- Ein einfaches Prädikat ist wie folgt

$$p ::= A_i \Phi const$$

- Einfache Prädikate sind also einfache Vergleiche der Art
 - TeileNr=4711
 - Gehalt>5000
- p definiert Partitionierung von R wie folgt:

$$R^+ := \{t \in R \mid t \text{ erfüllt } p\} \quad R^- := \{t \in R \mid t \text{ erfüllt } p \text{ nicht}\}$$

- Vorsicht bei Nullwerten

$$R^+ = \sigma_p(R) \quad R^- = \sigma_{\neg p}(R)$$

Menge aller Minterme

$P = \{p_1, \dots, p_n\}$ einfache Prädikate für R

$$M_n(P) := \left\{ m \mid m = \bigwedge_{i \in \{1, \dots, n\}} p_i^{\pm}, p_i \in P \right\}$$

$$|M_n(P)| = 2^n$$

$$|M_4(P)| = 2^4 = 16$$

$p_1 \wedge p_2 \wedge p_3 \wedge p_4$	$p_1 \wedge p_2 \wedge p_3 \wedge \overline{p_4}$	$p_1 \wedge p_2 \wedge \overline{p_3} \wedge p_4$	$p_1 \wedge \overline{p_2} \wedge p_3 \wedge p_4$
$\overline{p_1} \wedge p_2 \wedge p_3 \wedge p_4$	$p_1 \wedge p_2 \wedge \overline{p_3} \wedge \overline{p_4}$	$p_1 \wedge \overline{p_2} \wedge \overline{p_3} \wedge p_4$	$\overline{p_1} \wedge \overline{p_2} \wedge p_3 \wedge p_4$
$\overline{p_1} \wedge p_2 \wedge \overline{p_3} \wedge p_4$	$\overline{p_1} \wedge p_2 \wedge p_3 \wedge \overline{p_4}$	$p_1 \wedge \overline{p_2} \wedge p_3 \wedge \overline{p_4}$	$\overline{p_1} \wedge \overline{p_2} \wedge \overline{p_3} \wedge p_4$
$\overline{p_1} \wedge \overline{p_2} \wedge p_3 \wedge \overline{p_4}$	$\overline{p_1} \wedge p_2 \wedge \overline{p_3} \wedge \overline{p_4}$	$p_1 \wedge \overline{p_2} \wedge \overline{p_3} \wedge \overline{p_4}$	$\overline{p_1} \wedge \overline{p_2} \wedge p_3 \wedge \overline{p_4}$

Partitionierung gemäß $M(P)$

- Es gilt folgendes

$$\bigcup_{m \in M(P)} (\sigma_m(R)) = R$$

$$\forall m_i, m_k \in M(P), m_i \neq m_k : (\sigma_{m_i}(R)) \cap (\sigma_{m_k}(R)) = \emptyset$$

- Also gilt bezüglich der Partitionierung
 - Vollständigkeit
 - Redundanzfreiheit

Beispiel

- Einfache Prädikate P
 - p1: Gehalt=5000
 - p2: Gehalt=7000
 - p3: AbtNr > 400
 - p4: AbtNr < 300
- In $M(P)$ gibt es **unerfüllbare** Min-Term-Prädikate, z.B.
 - Gehalt=5000 **and** Gehalt=7000 **and** AbtNr > 400 **and** AbtNr < 300
- Diese werden aus $M(P)$ entfernt

Subsumierte Prädikate

- Funktionale Abhängigkeit
 - $AbtNr \longrightarrow schema(ABT)$
- Betrachte folgende Prädikate
 - $p1: AbtNr=4711$ $p2: AbtName=„Einkauf“$
 - $p3: AbtNr=5910$ $p4: AbtName=„Verkauf“$
- Erfüllbar sind nur Minterme mit nur positiven oder nur negativen p_i 's und außerdem gilt wegen der FD:

$$\sigma_{p_1 \wedge p_2}(ABT) \equiv \sigma_{p_1}(ABT)$$

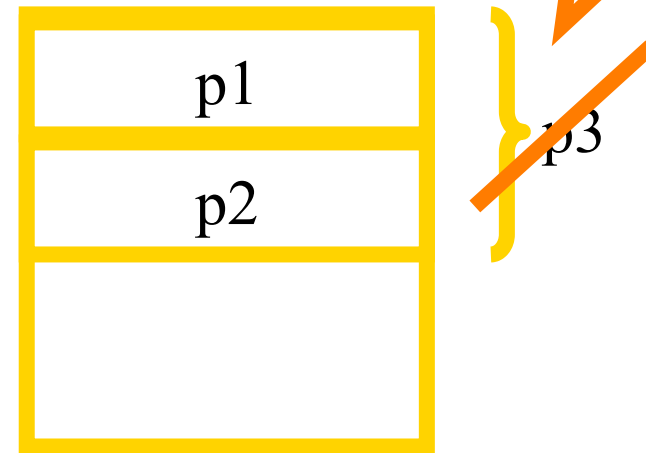
$$\sigma_{\neg p_1 \wedge \neg p_2}(ABT) \equiv \sigma_{\neg p_1}(ABT)$$

Menge der praktisch relevanten Minterme

- Entferne unerfüllbare Minterme aus $M(P)$
- Wenn ein Faktor p_i (oder **not** p_i) durch einen Faktor p_j (oder **not** p_j) impliziert wird, so entferne ihn aus dem Minterm
 - p_1 **and** ... **and** ~~p_i~~ **and** ... **and** p_j **and** p_n
 - Dadurch wird der Minterm $(n-1)$ -stellig
 - führe das ggfls. Mehrmals durch
- Was übrig bleibt sind die **praktisch relevanten Minterme**

Elimination irrelevanter Prädikate

- Beispiel
 - p1: $100 < \text{AbtNr} < 300$
 - p2: $300 < \text{AbtNr} < 600$
 - p3: $100 < \text{AbtNr} < 600$
- Allgemein sollte ein Prädikat p bei der Fragemntierung nur berücksichtigt werden, wenn mindestens eine Anwendung eine **wesentliche Verbesserung** der Zugriffskosten „erfährt“.



P3 ergibt keine wesentliche Verbesserung

Wesentliche Verbesserung ...

- ... ist gegeben wenn

$$\frac{\mathit{access}(R_p)}{\mathit{card}(R_p)} \ll \gg \frac{\mathit{access}(R_{p^-})}{\mathit{card}(R_{p^-})}$$

- R_p bzw. R_{p^-} sind die durch p bzw p^- definierten Fragmente von R
- $\mathit{access}(R_p)$: erwartete Anzahl der Zugriffe auf R_p
- $\mathit{card}(R_p)$: Kardinalität des Fragments R_p

Beispiel eines Prädikates mit wesentlicher Verbesserung

- 300 Zugriffe pro Woche auf ANGEST
- Davon 100 mit Prädikat $p = \text{„AbtNr} = 4711\text{“}$
- dadurch werden 50 von 10.000 Tupel selektiert

$$\frac{\text{access}(R_p)}{\text{card}(R_p)} = \frac{300}{50} = 6 \ll \gg \frac{200}{9.950} \approx 0.02 = \frac{\text{access}(R_{p^-})}{\text{card}(R_{p^-})}$$

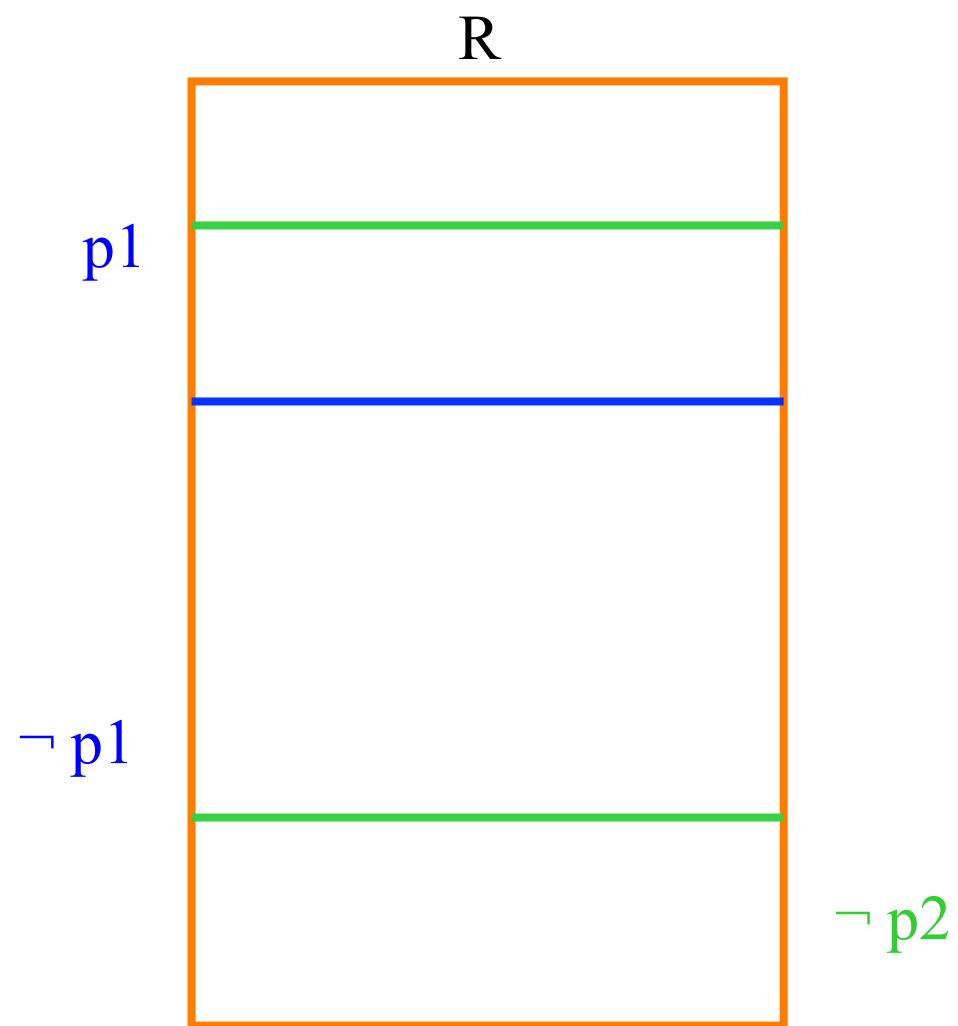
- Also ist p ein für die Partitionierung relevantes Prädikat
- Alle irrelevanten Prädikate sollten eliminiert werden

Fragwürdige Formel

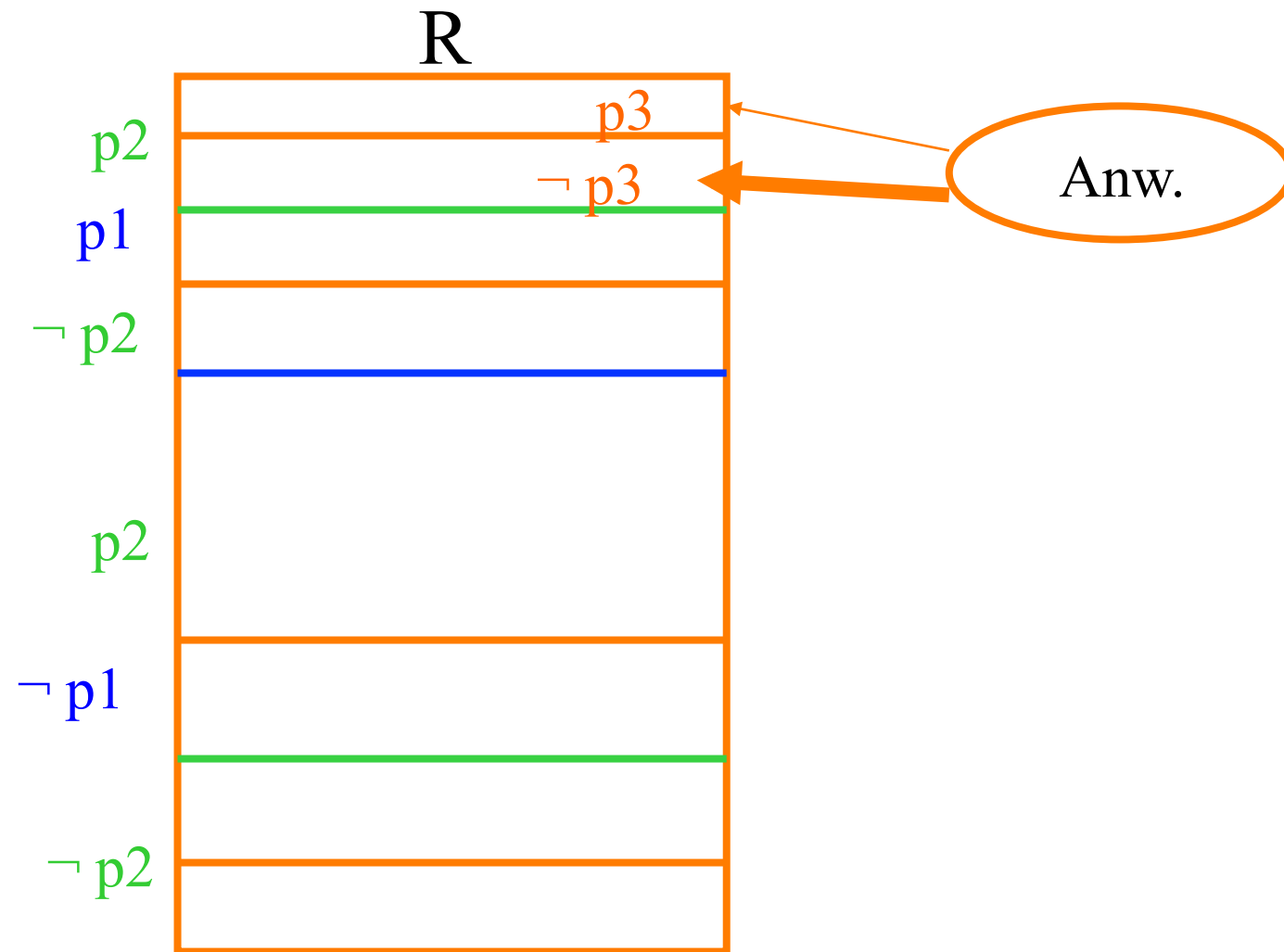
$\text{acc}(\dots) / \text{card}(\dots) \quad ???$

- Der „Wert“ der obigen Formel wurde in Frage gestellt --- zu Recht!
- Ceri und Pelagatti definieren die Relevanz eines Prädikats p_i bezogen auf eine Menge von Prädikaten P wie folgt:
 - es gibt mindestens zwei Minterme
 - $m_1 = p_1^* \wedge \dots \wedge p_i \wedge \dots \wedge p_n^*$ und
 - $m_2 = p_1^* \wedge \dots \wedge p_i^- \wedge \dots \wedge p_n^*$
 - aus $M(P)$, die sich **nur** in p_i unterscheiden (also einmal p_i und einmal $\neg p_i$ enthalten, bzw p_i^+ und p_i^-)
 - die resultierenden Fragmente $\sigma_{m_1}(R)$ und $\sigma_{m_2}(R)$ werden von mindestens einer Anwendung unterschiedlich referenziert
- Es wird also nur die Zugriffsfrequenz acc einer bestimmten Anwendung „zu Rate“ gezogen

Illustration



Illustration



Algorithmus HORIZ_PART: ¹⁰

Sei $Q := \emptyset$ und $M(Q)$, $F(Q)$ wie oben definiert.

for all $p \in P$ do

begin

setze $Q' := Q \cup \{ p \}$;

berechne $M(Q')$ und $F(Q')$;

vergleiche $F(Q')$ mit $F(Q)$;

if $F(Q')$ eine „wesentliche Verbesserung“ gegenüber $F(Q)$ then

begin

setze $Q := Q'$; /* d.h. nimm p in die Menge Q auf */

for all $q \in Q \setminus \{ p \}$ do /* prüfe auf unnötige Partitionierung */

begin

setze $Q' := Q \setminus \{ q \}$;

berechne $M(Q')$ und $F(Q')$;

vergleiche $F(Q')$ mit $F(Q)$;

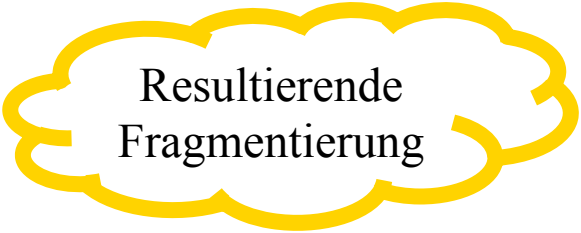
if $F(Q)$ keine „wesentliche Verbesserung“ gegenüber $F(Q')$ then

setze $Q := Q'$; /* d.h. entferne q aus Q */

end;

end;

end;



Resultierende
Fragmentierung

Beispiel-Rechnung

- Drei separate Bereiche
 - p1: Bereich = „LKW“
 - p2: Bereich = „PKW“
 - p3: Bereich = „Funk“
- Verwaltung gemäß des Budget
 - p4: Budget < 100.000
 - p5: Budget >= 100.000
- m1: Bereich = „LKW“ **and** Budget < 100.000
- m2: Bereich = „LKW“ **and** Budget >= 100.000
- m3: Bereich = „PKW“ **and** Budget < 100.000
- m4: Bereich = „PKW“ **and** Budget >= 100.000
- m5: Bereich = „Funk“ **and** Budget < 100.000
- m6: Bereich = „Funk“ **and** Budget >= 100.000

Bestimmung einer guten abgeleiteten Partitionierung

- $R \text{ --- } S$: Wenn sehr oft ein Join zwischen R und S auszuführen ist, mit einer vorangehenden Selektion auf R
- Partitioniere zunächst $R \text{ --}$ gemäß den Selektionsprädikaten
- Danach bestimmt man die abgeleiteten Partitionen von S

Bestimmung von vertikalen Partitionen

- $R: \{[K, A_1, A_2, \dots, A_n]\}$
- Bestimme die Fragmente bezüglich einer guten Ballung (Clustering)
 - $R_1: \{[K, A_{i1}, A_{i2}, \dots]\}$
 - $R_2: \{[K, A_{j1}, A_{j2}, \dots]\}$
- Oder: kleine Partitionen
 - $R_1: \{[K, A_1]\}$
 - $R_2: \{[K, A_2]\}$
 -

Beispielsweise würde man aus einer ANGEST-Rel. die Verwaltungsdaten (Gehalt, Steuerklasse) von Archivdaten (Lebenslauf, Paßbild, etc.) abspalten

Mathematisches Modell für Nicht-redundante Allokation

- K Anzahl Knoten
- P Anzahl zu allozierender Partitionen
- T Anzahl Typen von Lese- und Änderungs-Operationen
- M_i maximale Speicherkapazität am Knoten i
- S_i Speicherkosten pro Dateneinheit am Knoten i
- U_{ij} Übertragungskosten von Knoten i nach j
- G_p Größe in Dateneinheiten der Partition p
- O_{tp} Größe einer Teiloperation vom Typ t gegen Partition p
- R_{tp} Größe des Resultats einer Teilop t gegen Part. p
- H_{it} Häufigkeit mit der Teilop t an Knoten i auszuführen ist
- V_{pi} Verteilung der Partitionen auf die Knoten
 - $V_{pi}=1$: p ist auf Knoten i alloziert
 - $V_{pi}=0$: sonst

Kostenformeln

- Speicherkosten:

$$\sum_S = \sum_{p,i} G_p * V_{pi} * S_i$$

- Übertragungskosten

$$\sum_U = \left(\sum_{i,t,p,j} Hit * Otp * V_{pj} * U_{ij} \right) + \left(\sum_{i,t,p,j} Hit * Rtp * V_{pj} * U_{ji} \right)$$

- Code-Shipping + Result/Data-Shipping

- Nebenbedingung für nicht-redundante

Speicherung: $\forall_{p \in \{1, \dots, P\}} : \left(\sum_i V_{pi} \right) = 1$

- Einhaltung maximaler Speicherkapazität:

$$\left(\sum_p G_p * V_{pi} \right) \leq M_i \text{ für alle } i = 1, \dots, K$$

Kostenformeln für redundante Speicherung

- Speicherkosten $\sum_S = \sum_{p,i} G_p * V_{pi} * S_i$
- Übertragungskosten

$$\sum_U = \left(\sum_{i,t,p} Hit * \Phi^t \left(Otp * Uij + Rtp * Uji \right) \right)_{j:Vpj=1}$$

Φ^t : entweder MIN (Lese - Op) oder \sum (Schreib - Op)
 $_{j:Vpj=1}$

- Nebenbedingung für Speicherung:

$$\forall_{p \in \{1, \dots, P\}} : \left(\sum_i V_{pi} \right) \geq 1$$

- Einhaltung maximaler Speicherkapazität:

$$\left(\sum_p G_p * V_{pi} \right) \leq Mi \text{ für alle } i = 1, \dots, K$$

Optimierungsproblem

- Minimiere Speicher- und Übertragungskosten ...

$$\underbrace{\left(\sum_{p,i} G_p * V_{pi} * S_i \right)}_{\sum_s} + \underbrace{\left(\sum_{i,t,p,j} Hit * O_{tp} * V_{pj} * U_{ij} \right) + \left(\sum_{i,t,p,j} Hit * R_{tp} * V_{pj} * U_{ji} \right)}_{\sum_u}$$

- ... unter Einhaltung der Nebenbedingungen

$$\forall_{p \in \{1, \dots, P\}} : \left(\sum_i V_{pi} \right) = 1$$

$$\left(\sum_p G_p * V_{pi} \right) \leq M_i \text{ für alle } i = 1, \dots, K$$

Optimierungsproblem für redundante Speicherung

- Minimiere Speicher- und Übertragungskosten:

$$\underbrace{\left(\sum_{p,i} G_p * V_{pi} * S_i \right)}_{\sum_s} + \underbrace{\left(\sum_{i,t,p,j} Hit * \Phi^t (Otp * U_{ij} + Rtp * U_{ji}) \right)}_{\sum_U}$$

- unter Einhaltung der Nebenbedingungen

$$\forall_{p \in \{1, \dots, P\}} : \left(\sum_i V_{pi} \right) \geq 1$$

$$\left(\sum_p G_p * V_{pi} \right) \leq M_i \text{ für alle } i = 1, \dots, K$$

Beispiel-Konfiguration

- $K = 3$
- $P=4$
- $T=3$
- $(S_i)=(120 \ 100 \ 110)$

$$(U_{ij}) = \begin{bmatrix} 0 & 25 & 30 \\ 25 & 0 & 35 \\ 30 & 35 & 0 \end{bmatrix}$$

$$(G_p) = [1000 \ 1500 \ 500 \ 2000]$$

$$(R_{tp}) = \begin{bmatrix} 100 & 200 & 10 & 20 \\ 200 & 300 & 5 & 8 \\ 10 & 10 & 200 & 100 \end{bmatrix}$$

$$(Hit) = \begin{pmatrix} 50 & 7 & 10 \\ 5 & 75 & 2 \\ 3 & 8 & 50 \end{pmatrix}$$

Operation t_i wird bevorzugt an Knoten K_i ausgeführt

Ops t_1 und t_2 greifen vornehmlich auf Partitionen P_1 und P_2 zu; Op t_3 auf P_3 und P_4

P_1 auf K_1 oder K_2 , P_2 auf K_1 oder K_2 , P_3 auf K_3 , P_4 auf K_3

Optimal
 P_1 auf K_2 ,
 P_2 auf K_2 ,
 P_3 auf K_3 ,
 P_4 auf K_3

Redundante Speicherung

- $K = 3$
- $P=4$
- $T=3$
- $(S_i)=(120 \ 100 \ 110)$

$$(U_{ij}) = \begin{bmatrix} 0 & 25 & 30 \\ 25 & 0 & 35 \\ 30 & 35 & 0 \end{bmatrix}$$

$$(G_p) = [1000 \quad 1500 \quad 500 \quad 2000]$$

$$(R_{tp}) = \begin{bmatrix} 100 & 200 & 10 & 20 \\ 200 & 300 & 5 & 8 \\ 10 & 10 & 200 & 100 \end{bmatrix}$$

$$(Hit) = \begin{pmatrix} 50 & 7 & 10 \\ 5 & 75 & 2 \\ 3 & 8 & 50 \end{pmatrix}$$

Operation t_i wird bevorzugt an Knoten K_i ausgeführt

Ops t_1 und t_2 greifen vornehmlich auf Partitionen P_1 und P_2 zu; Op t_3 auf P_3 und P_4

Optimal
 P_1 auf K_1 und K_2 , P_2 auf K_1 und K_2 ,
 P_3 auf K_1 und K_3 , P_4 auf K_3

Berechnung des Optimums

- Vollständige Suche (exhaustive search)
- nicht-redundante Speicherung
 - $K \cdot P$ Möglichkeiten durchprobieren
- redundante Speicherung
 - $(2^K - 1) \cdot P$ Möglichkeiten
 - Jede der P Partition kann auf jedem der K Knoten entweder allokiert werden oder nicht, auf einem muß sie aber sein (deshalb „AnzTeilmengen - 1“)

Diskussion

- 0/1 Optimierungsproblem
- sehr hoher Rechenaufwand
- Ermittlung der Parameter (Anfragehäufigkeit, Selektionsprädikate, Kostenparameter wie z.B. Netzwerkbandbreite) ist schwierig
- Parameter ändern sich dynamisch
- Einsatz von Heuristiken

Schema-Architekturen verteilter Datenbanksysteme

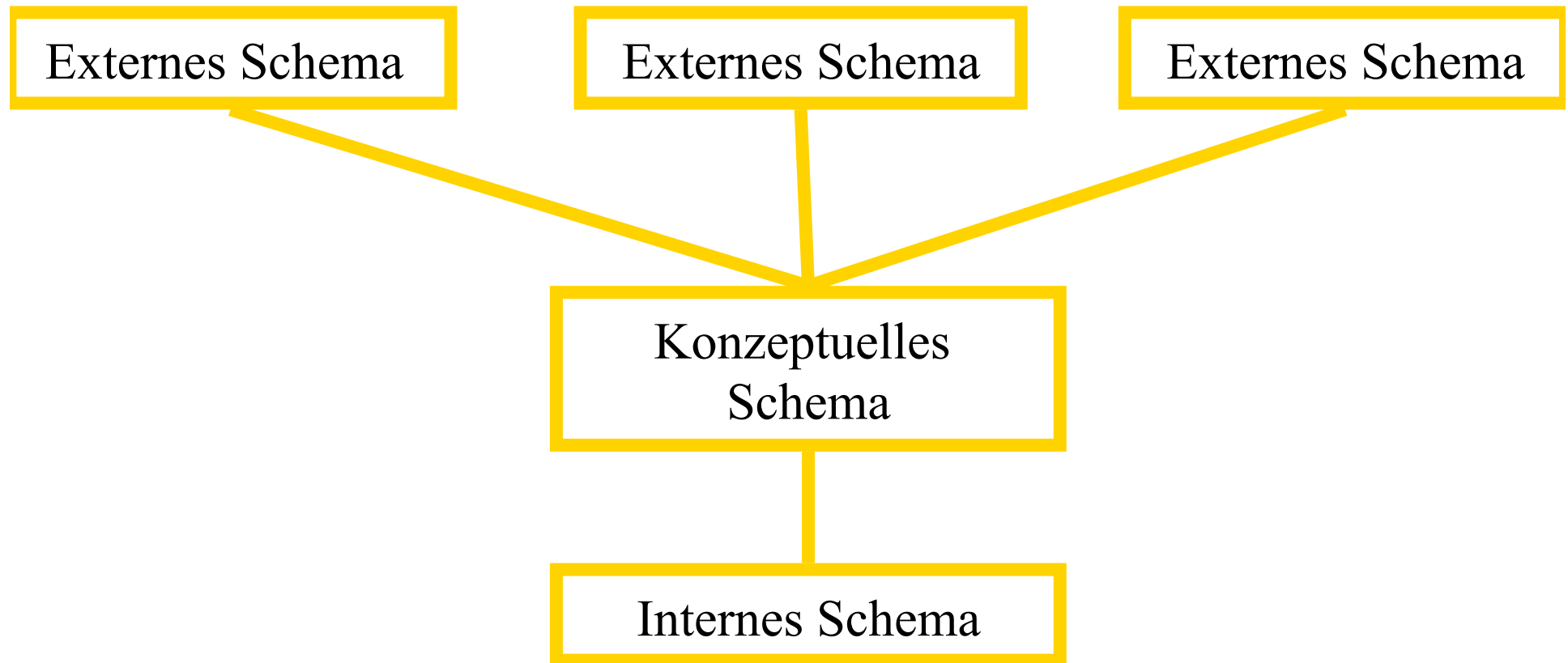


**Verwaltung der
Fragmentierungs- und
Allokationsinformation**

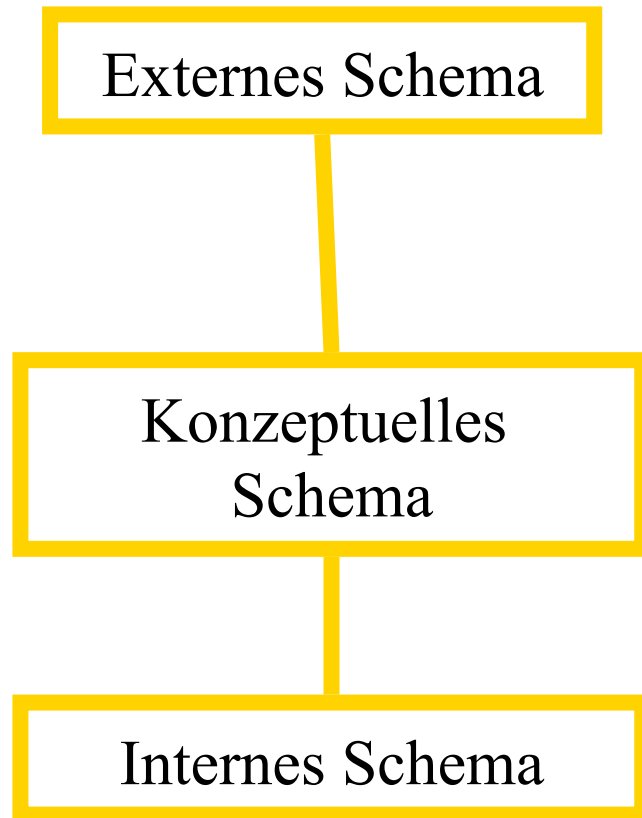
Ziel

- Verstecken/Verbergen (vor dem Benutzer)
 - von Heterogenität in den lokalen Schemata/
Datenmodellen
 - der Partitionierung der globalen Relation
 - der Allokation der Partitionen/Relationen
 - von redundanten Speicherung von Partitionen
 - zur Leistungssteigerung
 - zur Ausfallsicherheit
- Bereitstellung von Informationen für den
Query - Optimierer und Prozessor und
DBA

ANSI/SPARC Drei-Schema Architektur

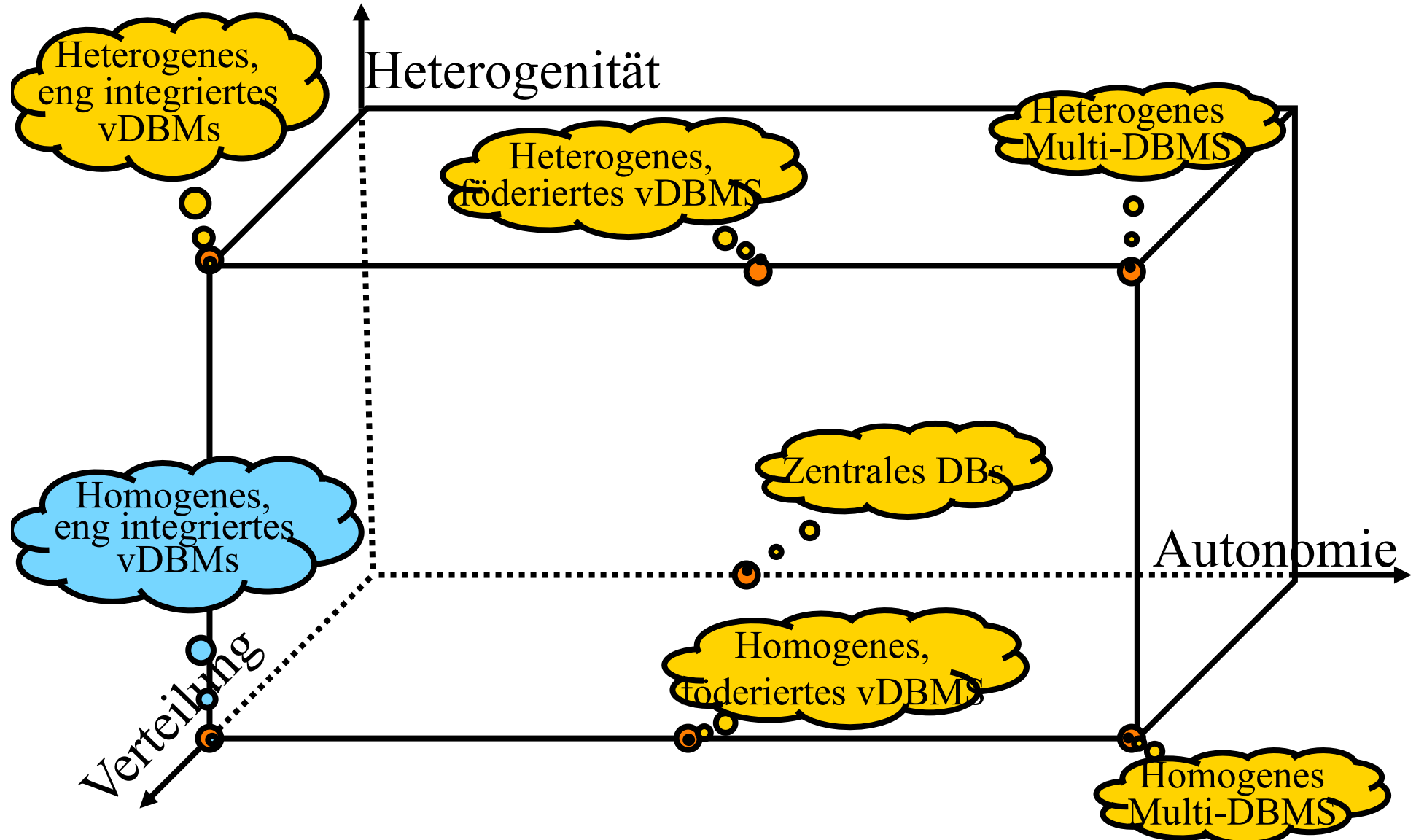


ANSI/SPARC Drei-Schema Architektur -- rel. DBMS



- Views (Sichten)
- Relationen
- Referentielle Integritätsbedingungen
- Indexe, Cluster, DB-Extents, Seiten

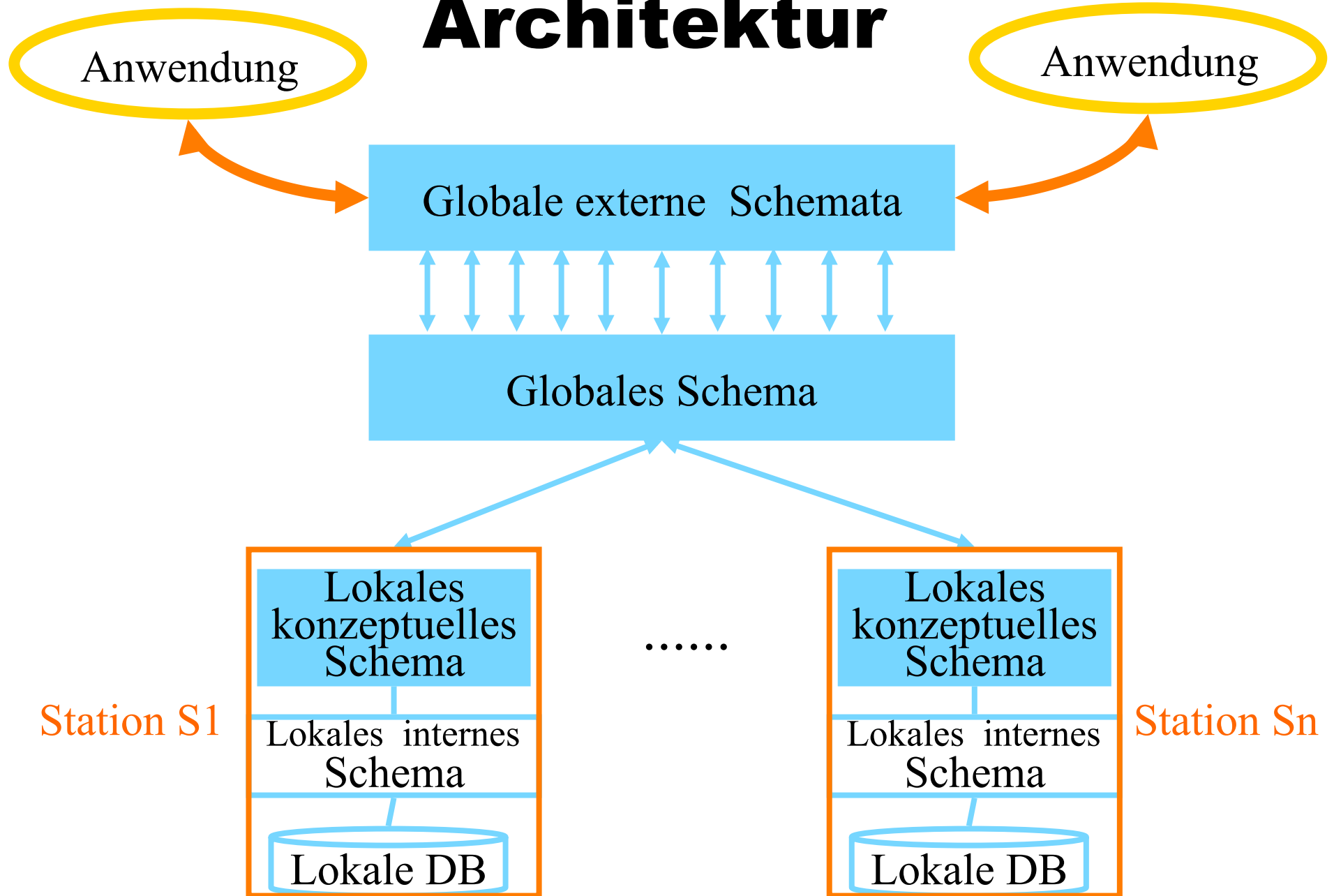
Klassifikation verteilter DBMS



Homogene, prä-integrierte DBMSs

- Von vornherein als verteilte DB ausgelegt
 - Performance-Steigerung
 - Ausfallsicherheit
 - inkrementelle Erweiterbarkeit
- keine Altlasten = „schöne heile Welt“
- Ausgangssituation
 - globales Schema
 - davon abgeleitete Fragmentierung/Partitionierung und Allokation

Architektur



Architektur

Anwendung

Anwendung

Globale externe Schemata

Globales Konzeptuelles Schema (GKS)

Globales Partitionierungs-Schema (GPS)

Globales Allokations-Schema (GAS)

3-
Schema-
Architektur

Lokales
konzeptuelles
Schema

Lokales internes
Schema

Lokale DB

Station S1

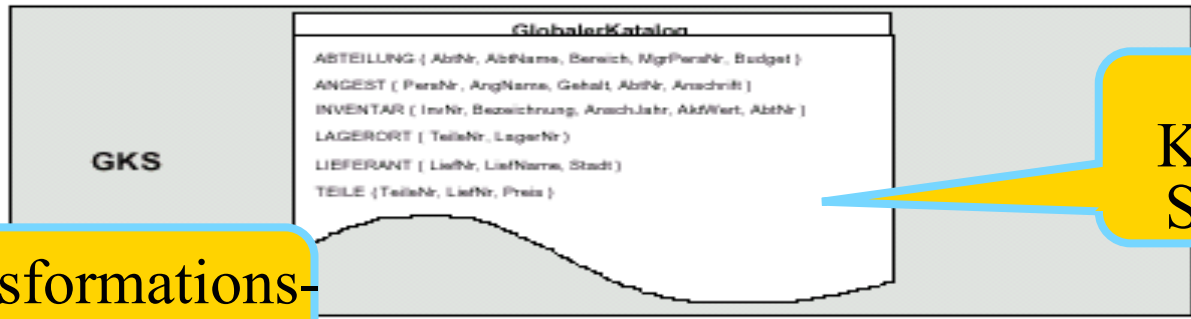
.....

Lokales
konzeptuelles
Schema

Lokales internes
Schema

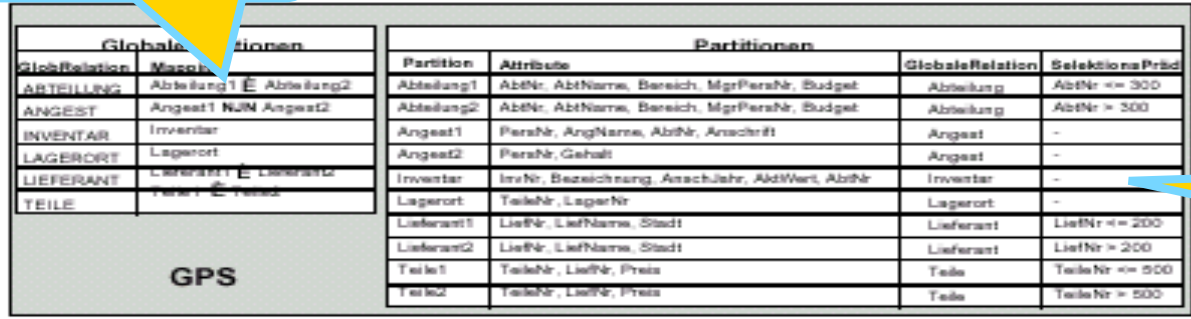
Lokale DB

Station Sn

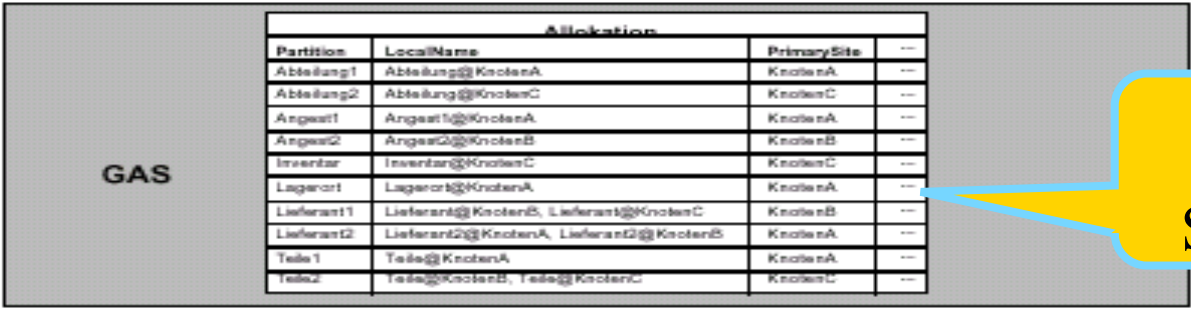


Globales Konzeptuelles Schema GKS

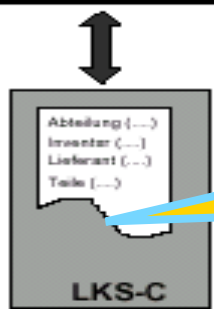
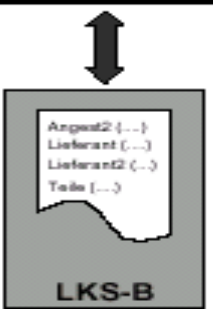
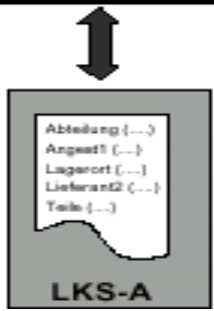
Transformations-Regeln



Globales Partitionierungs-Schema GPS



Globales Allokations-Schema GAS



Lokale Konzeptuelle Schemata LKS

Globaler
Katalog
GKS

ANGEST : {[PersNr, AngName, Gehalt, AbtNr, Anschrift]}

ABT : {[AbtNr, AbtName, Bereich, MgrPersNr, Budget]}

TEILE : {[TeileNr, TeileBez, LiefNr, Preis]}

LAGERORT : {[TeileNr, LagerNr]}

LIEFERANT : {[LiefNr, LiefName, Stadt]}

INVENTAR : {[InvNr, Bezeichnung, AnschJahr, AktWert, AbtNr]}

GPS

Transf.

ABT := *ABT1* ∪ *ABT2*

ANGEST := *ANGEST1*

⊗_{PersNr} *ANGEST2*

INVENTAR

LAGERORT

LIEFERANT := *LIEFERANT1*

∪ *LIEFERANT2*

TEILE := *TEILE1*

∪ *TEILE2*

Partitionen			
Partition	Attribute	Globale Rel.	Selektions-präd.
ABT1	AbtNr, AbtName, Bereich, MgrPersNr, Budget	ABT	AbtNr ≤ 300
ABT2	AbtNr, AbtName, Bereich, MgrPersNr, Budget	ABT	AbtNr > 300
ANGEST1	PersNr, AngName, AbtNr, Anschrift	ANGEST	---
ANGEST2	PersNr, Gehalt	ANGEST	---
INVENTAR	InvNr, Bezeichnung, AnschJahr, AktWert, AbtNr	INVENTAR	---
LAGERORT	TeileNr, LagerNr	LAGERORT	---
LIEFERANT1	LiefNr, LiefName, Stadt	LIEFERANT	LiefNr ≤ 200
LIEFERANT2	LiefNr, LiefName, Stadt	LIEFERANT	LiefNr > 200
TEILE1	TeileNr, LiefNr, Preis	TEILE	TeileNr ≤ 500
TEILE2	TeileNr, LiefNr, Preis	TEILE	TeileNr > 500

Globaler Katalog GKS

$ANGEST : \{[\underline{PersNr}, AngName, Gehalt, AbtNr, Anschrift]\}$
 $ABT : \{[\underline{AbtNr}, AbtName, Bereich, MgrPersNr, Budget]\}$
 $TEILE : \{[\underline{TeileNr}, TeileBez, LiefNr, Preis]\}$
 $LAGERORT : \{[\underline{TeileNr}, LagerNr]\}$
 $LIEFERANT : \{[\underline{LiefNr}, LiefName, Stadt]\}$
 $INVENTAR : \{[InvNr, Bezeichnung, AnschJahr, AktWert, AbtNr]\}$

Globales Partitionierungs-Schema

GPS

Transformationen

$ABT := ABT1 \cup ABT2$

$ANGEST := ANGEST1$

$\otimes_{PersNr} ANGEST2$

$INVENTAR$

$LAGERORT$

$LIEFERANT := LIEFERANT1$

$\cup LIEFERANT2$

$TEILE := TEILE1$

$\cup TEILE2$

Partitionen

Partition	Attribute	Globale Rel.	Selektions-präd.
ABT1	AbtNr, AbtName, Bereich, MgrPersNr, Budget	ABT	AbtNr ≤ 300
ABT2	AbtNr, AbtName, Bereich, MgrPersNr, Budget	ABT	AbtNr > 300
ANGEST1	PersNr, AngName, AbtNr, Anschrift	ANGEST	---
ANGEST2	PersNr, Gehalt	ANGEST	---
INVENTAR	InvNr, Bezeichnung, AnschJahr, AktWert, AbtNr	INVENTAR	---
LAGERORT	TeileNr, LagerNr	LAGERORT	---
LIEFERANT1	LiefNr, LiefName, Stadt	LIEFERANT	LiefNr ≤ 200
LIEFERANT2	LiefNr, LiefName, Stadt	LIEFERANT	LiefNr > 200
TEILE1	TeileNr, LiefNr, Preis	TEILE	TeileNr ≤ 500
TEILE1	TeileNr, LiefNr, Preis	TEILE	TeileNr ≤ 500

Partitionen-Allokation

Partition	LokalerName	PrimärKnoten	...
ABT1	ABT@KnotenA	KnotenA	
ABT2	ABT@KnotenC	KnotenC	
ANGEST1	ANGEST1@KnotenA	KnotenA	
ANGEST2	ANGEST1@KnotenB	KnotenB	
INVENTAR	INVENTAR@KnotenC	KnotenC	
LAGERORT	LAGERORT@KnotenA	KnotenA	
LIEFERANT1	LIEFERANT@KnotenB LIEFERANT@KnotenC	KnotenB	
LIEFERANT2	LIEFERANT@KnotenA LIEFERANT@KnotenB	KnotenA	
TEILE1	TEILE@KnotenA	KnotenA	
TEILE1	TEILE@KnotenB TEILE@KnotenC	KnotenB	

Globaler Katalog GKS

ANGEST : {[PersNr, AngName, Gehalt, AbtNr, Anschrift]}
 ABT : {[AbtNr, AbtName, Bereich, MgrPersNr, Budget]}
 TEILE : {[TeileNr, TeileBez, LiefNr, Preis]}
 LAGERORT : {[TeileNr, LagerNr]}
 LIEFERANT : {[LiefNr, LiefName, Stadt]}
 INVENTAR : {[InvNr, Bezeichnung, AnschJahr, AktWert, AbtNr]}

Globales Partitionierungs-Schema GPS

Trans

ABT := ABT1 \cup ABT2
 ANGEST := ANGEST1
 \otimes PersNr ANGEST2
 INVENTAR
 LAGERORT
 LIEFERANT := LIEFERANT1
 \cup LIEFERANT2
 TEILE := TEILE1
 \cup TEILE2

GKS

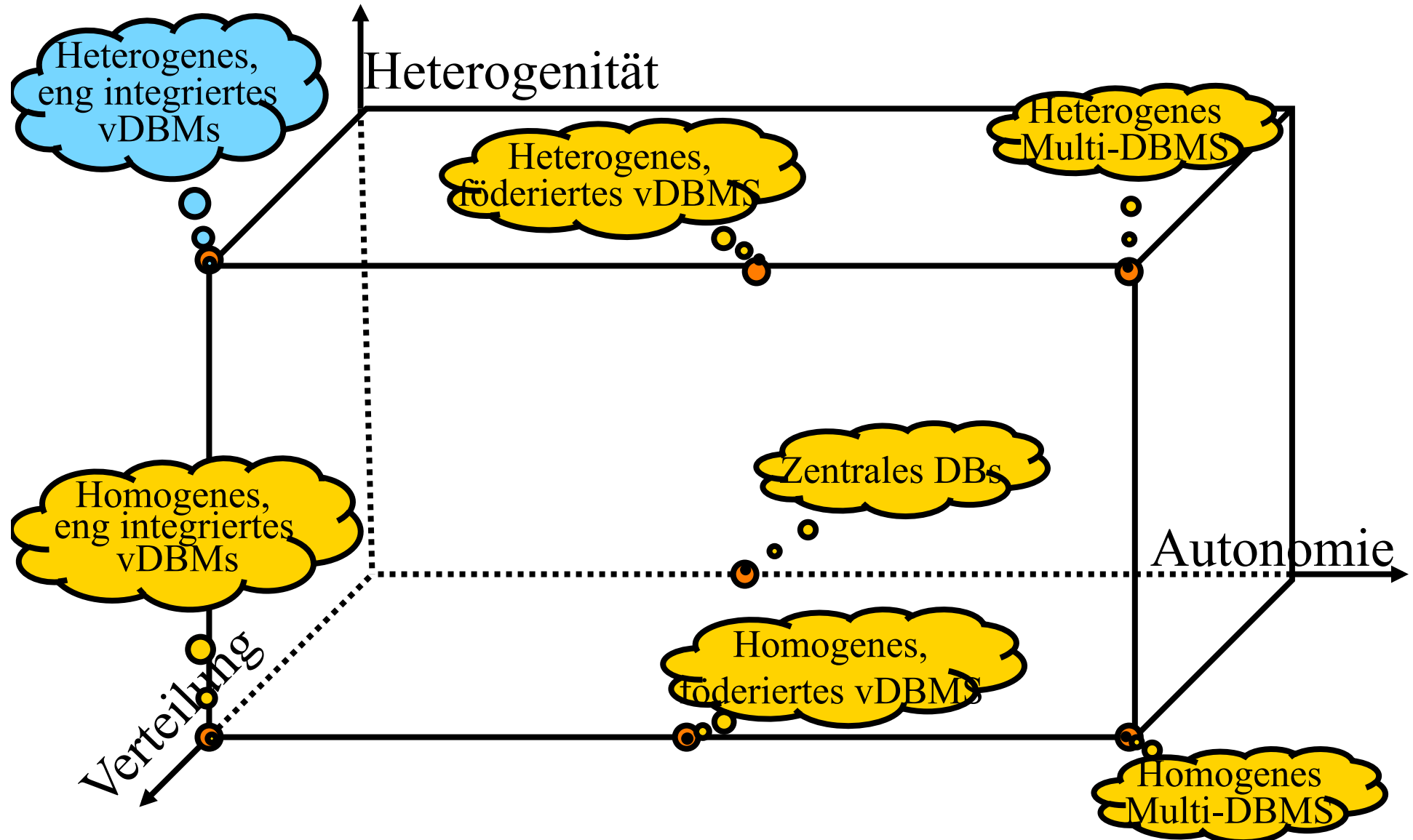
Partitionen			
Partition	Attribute	Globale Rel.	Selektions-präd.
ABT1	AbtNr, AbtName, Bereich, MgrPersNr, Budget	ABT	AbtNr <= 300
ABT2	AbtNr, AbtName, Bereich, MgrPersNr, Budget	ABT	AbtNr > 300
ANGEST1	PersNr, AngName, AbtNr, Anschrift	ANGEST	---
ANGEST2	PersNr, Gehalt	ANGEST	---
INVENTAR	InvNr, Bezeichnung, AnschJahr, AktWert, AbtNr	INVENTAR	---
LAGERORT	TeileNr, LagerNr	LAGERORT	---
LIEFERANT1	LiefNr, LiefName, Stadt	LIEFERANT	LiefNr <= 200
LIEFERANT2	LiefNr, LiefName, Stadt	LIEFERANT	LiefNr > 200
TEILE1	TeileNr, LiefNr, Preis	TEILE	TeileNr <= 500
TEILE2	TeileNr, LiefNr, Preis	TEILE	TeileNr > 500

Globales Allokations-Schema GAS

Partitionen-Allokation

Partition	LokalerName	PrimärKnoten
ABT1	ABT@KnotenA	KnotenA
ABT2	ABT@KnotenC	KnotenC
ANGEST1	ANGEST1@KnotenA	KnotenA
ANGEST2	ANGEST1@KnotenB	KnotenB
INVENTAR	INVENTAR@KnotenC	KnotenC
LAGERORT	LAGERORT@KnotenA	KnotenA
LIEFERANT1	LIEFERANT@KnotenB LIEFERANT@KnotenC	KnotenB

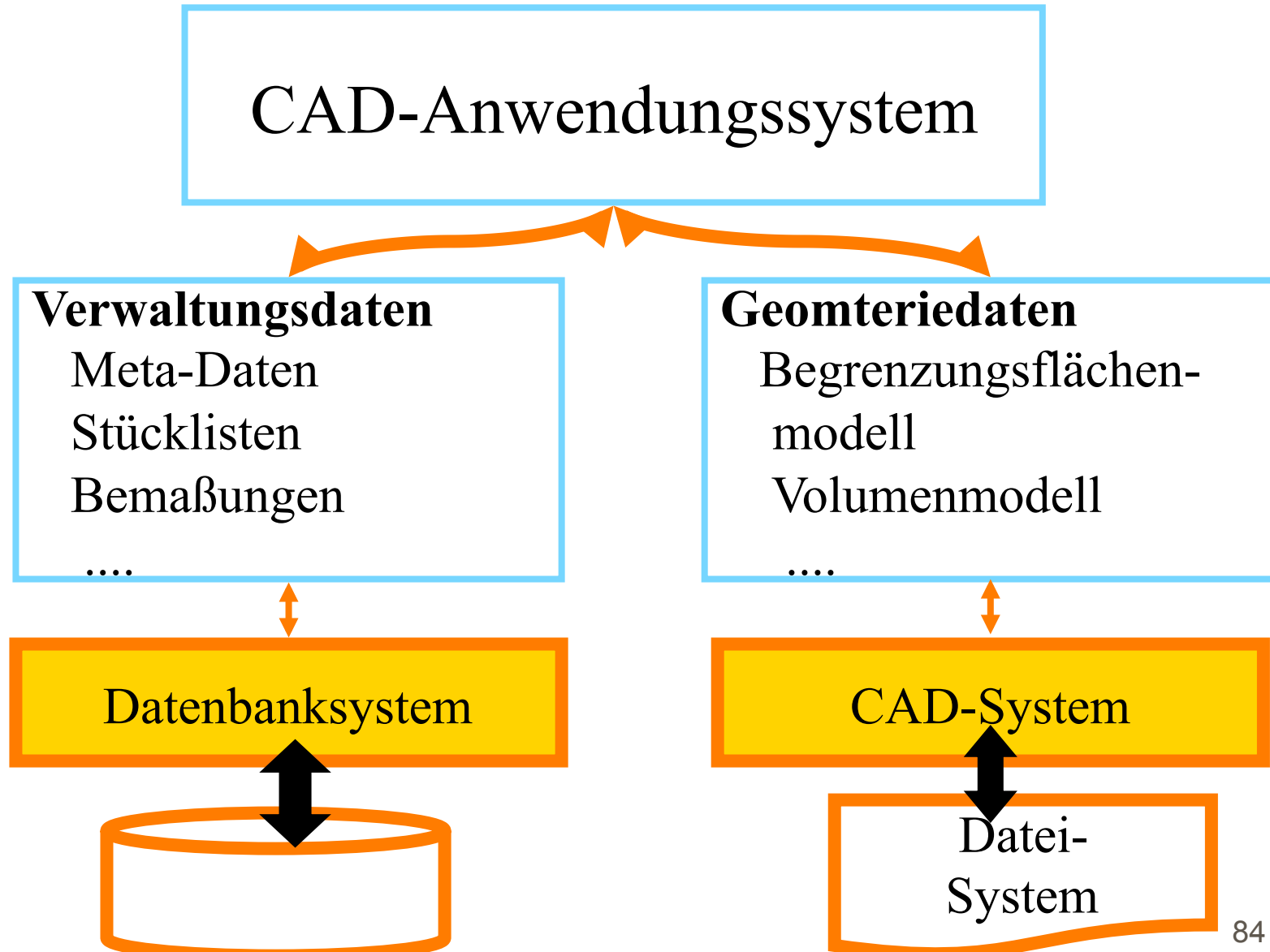
Klassifikation verteilter DBMS



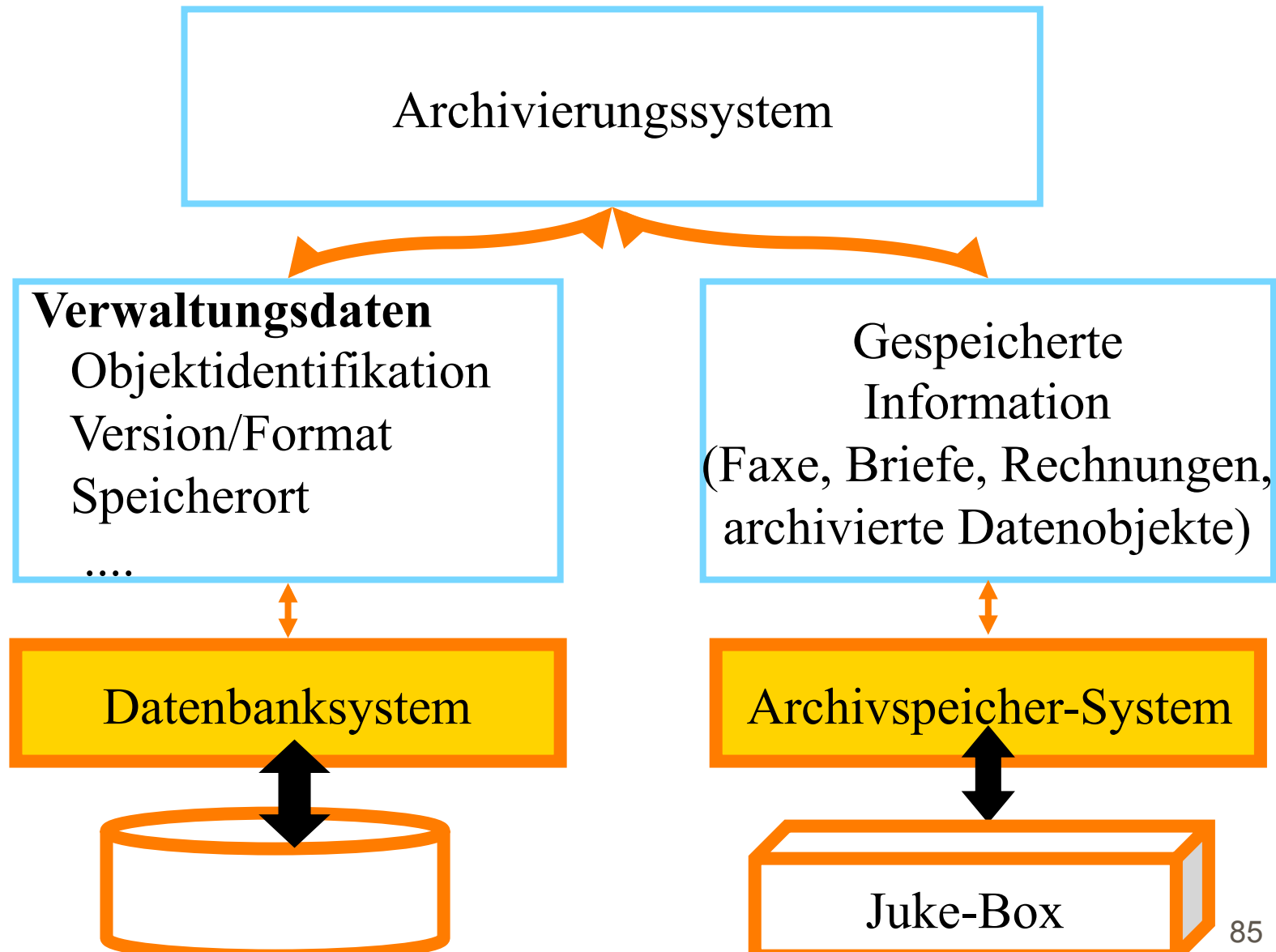
Heterogene, prä-integrierte Systeme

- Selten, nur in Spezialanwendungen
- Beispielszenarium
 - DBMS dient der Indexierung und Speicherung von Verwaltungsinformation
 - „aktive“ Nutzdaten werden im Dateisystem gespeichert
- Beispielanwendungen
 - CAD System
 - Archivierungssystem (IXOS)

Hybrid-Lösung: CAD-DB-Integration



Hybrid-Lösung: Archivierungssystem



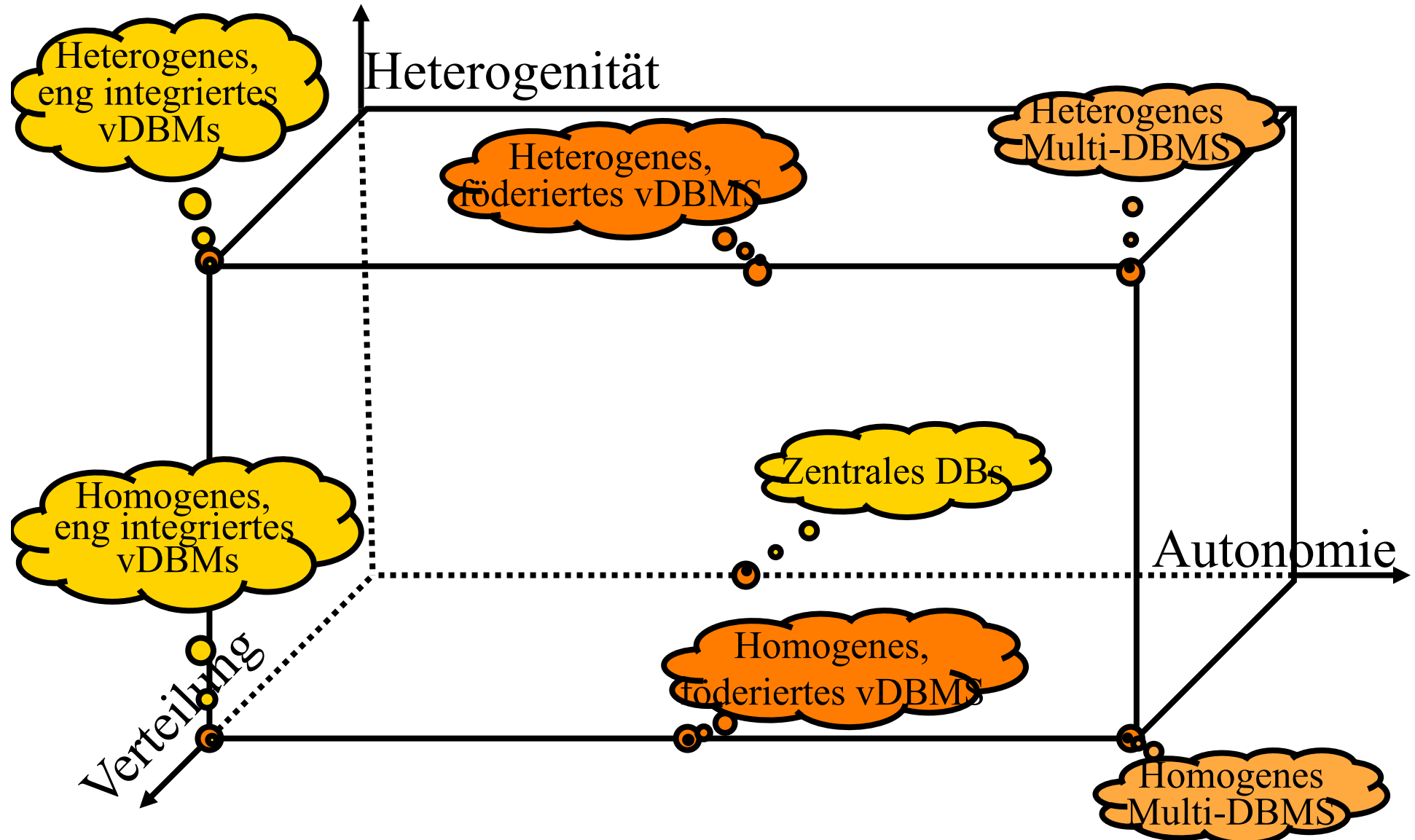
Resume

- Integration im Anwendungssystem
 - CAD-Anwendungssystem
 - Archivsystem
- Heterogenität ist auf Implementierungseben (des Anw.-Systems) voll sichtbar
- kein echtes verteiltes DBMS
- Interaktion der Komponenten ist im Anwendungssystem „verdrahtet“

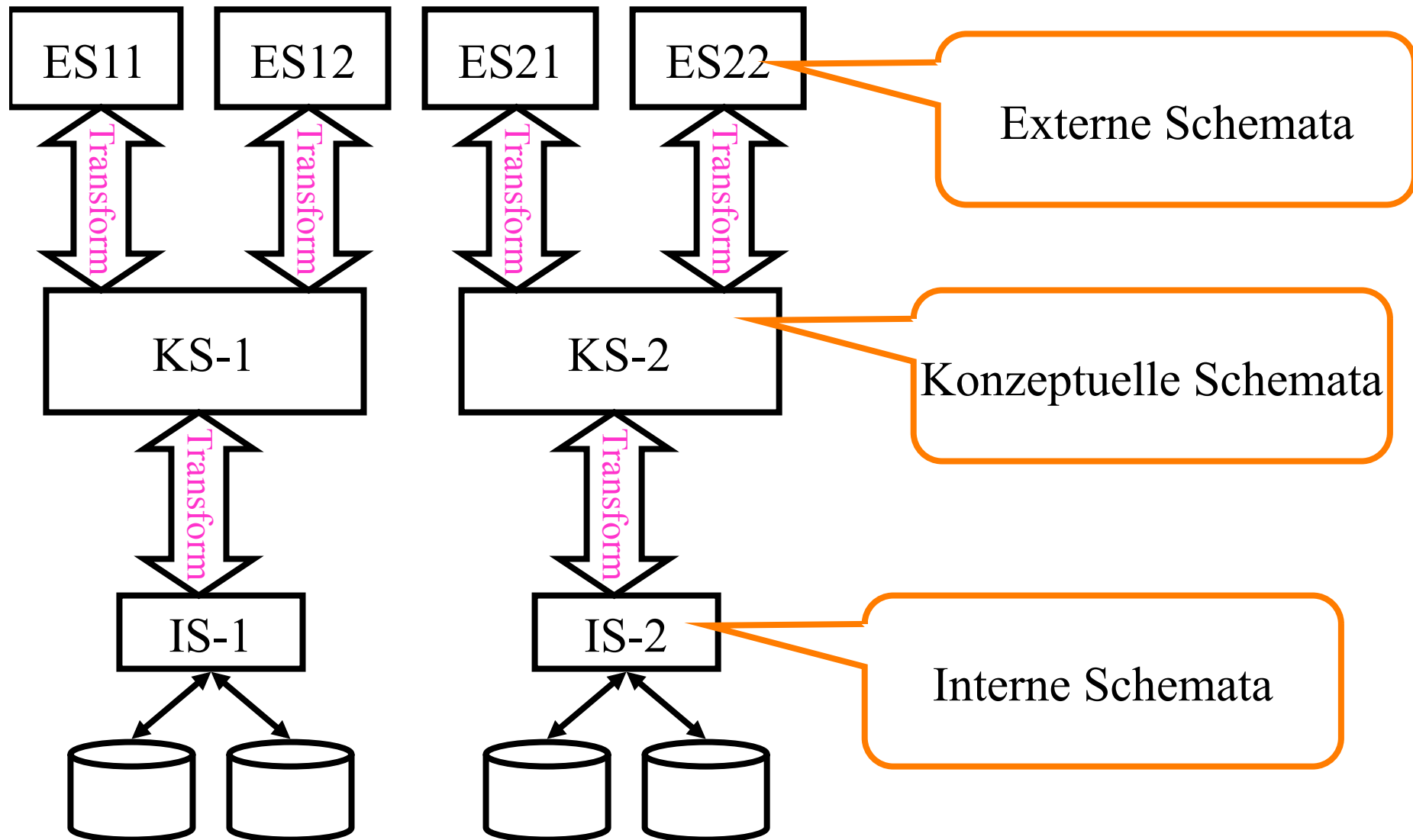
Heutige/zukünftige Entwicklungen

- Spezial-Datenbanksysteme für unterschiedliche Anwendungsdomänen
 - Geo-DBMS
 - Image-DBMS
 - Video-DBMS
- Realisierung durch Bausteine
 - Datablades in Informix
 - Erweiterungsmodule

Klassifikation verteilter DBMS: post-integrierte Systeme

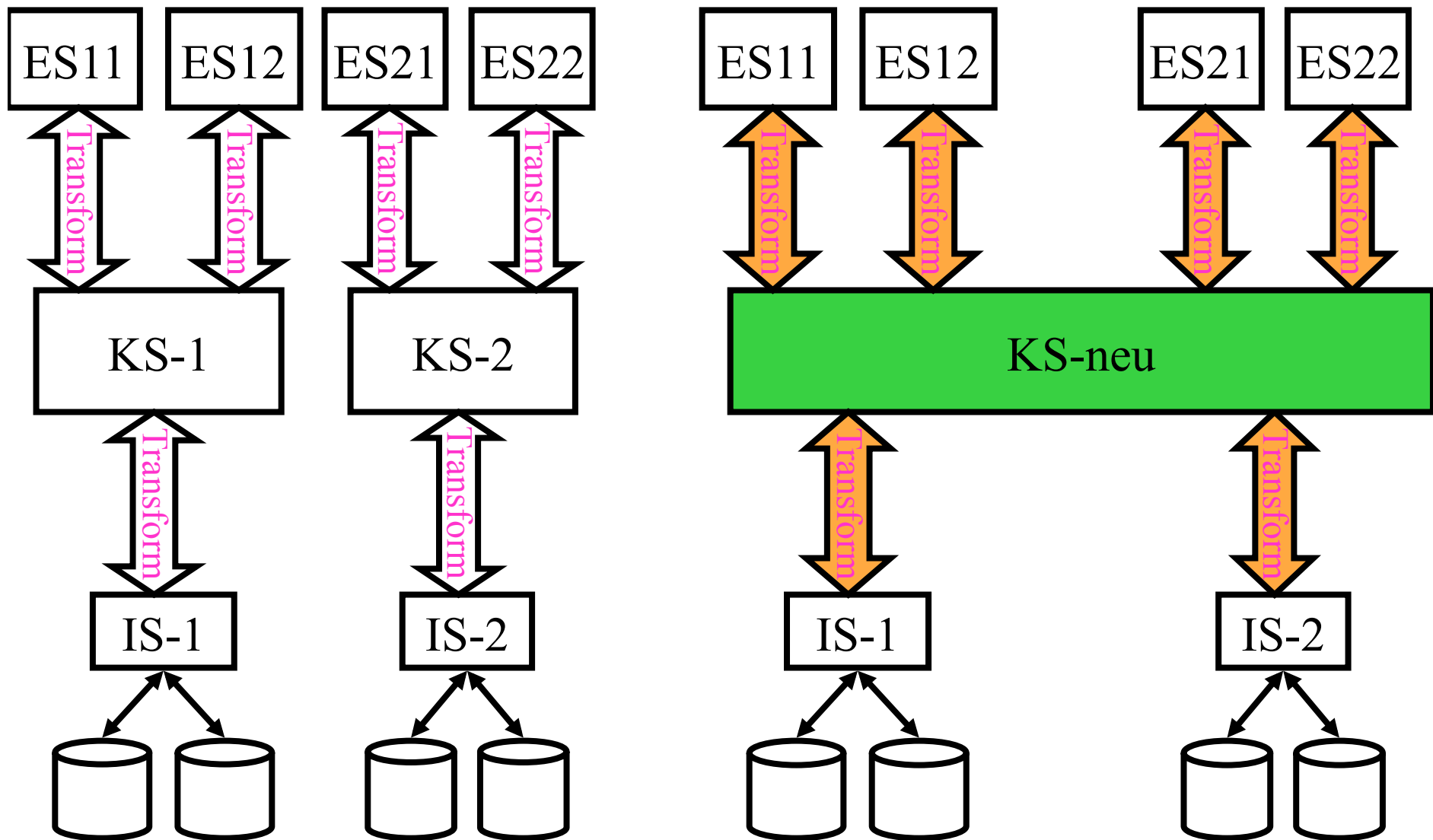


Ausgangssituation: Zwei „getrennte“ Datenbanksysteme



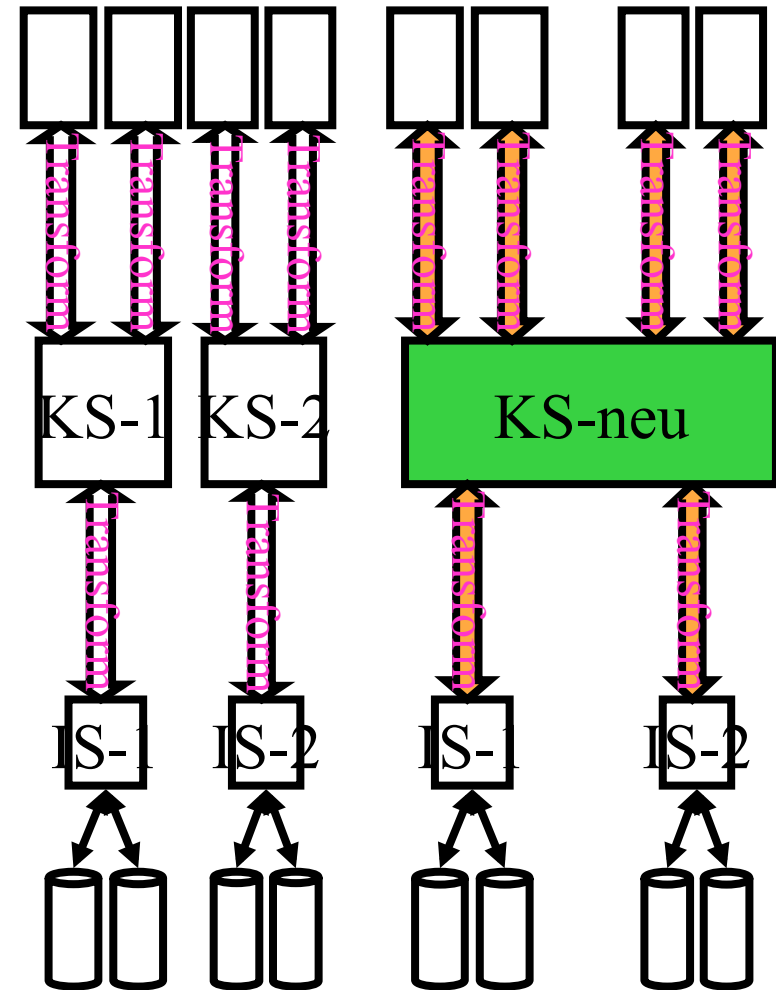
Integrationsversuch:

Gemeinsames Konz. Schema

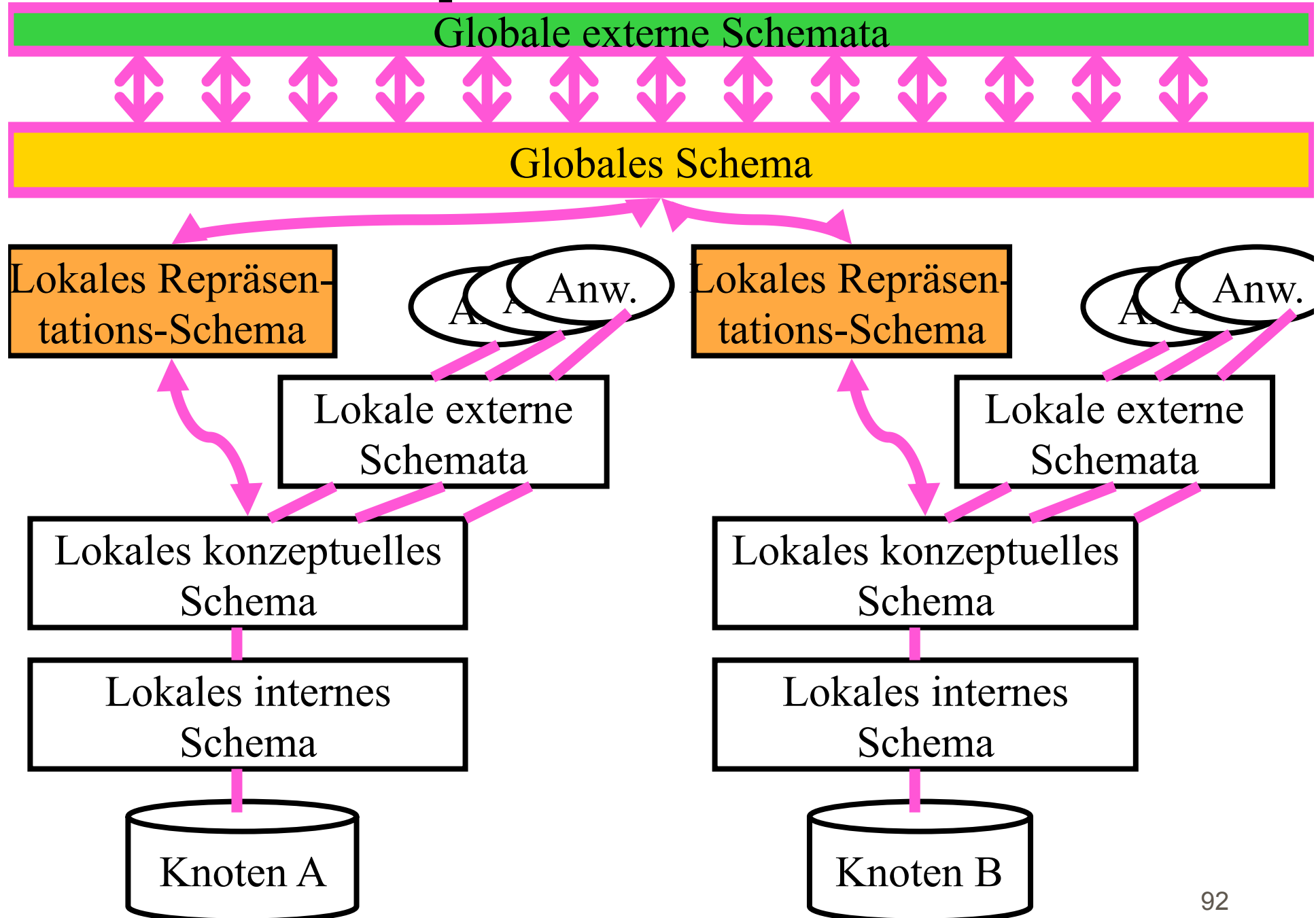


Integrationsversuch: Gemeinsames Konz. Schema

- In der Praxis nicht realisierbar
- gleichzeitige Änderung (über Nacht) aller Schemata
- Simulation des alten KS durch Bereitstellung der alten Sicht auf dem neuen KS



Lokale Repräsentationsschemata

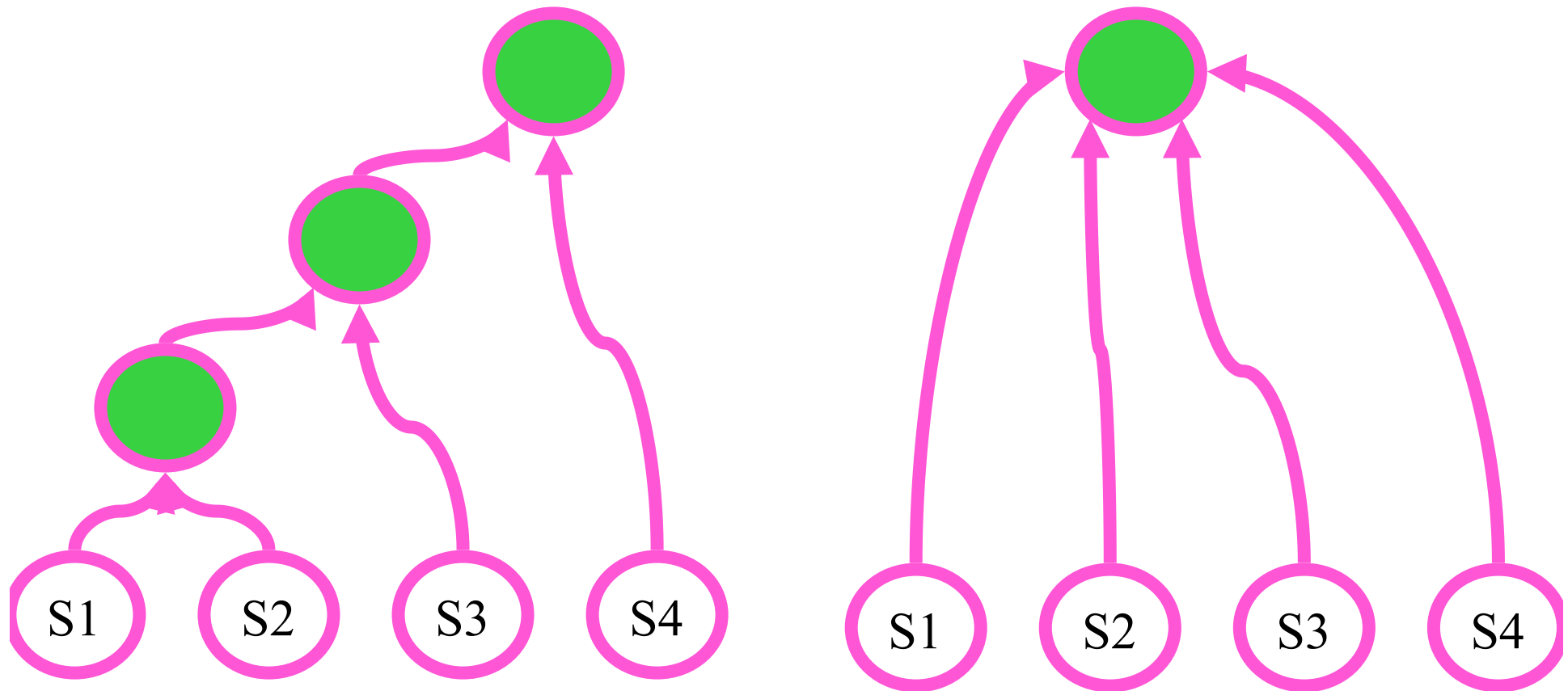


Schema-Integration

(analog zur Konsolidierungsphase im DB-Entwurf)

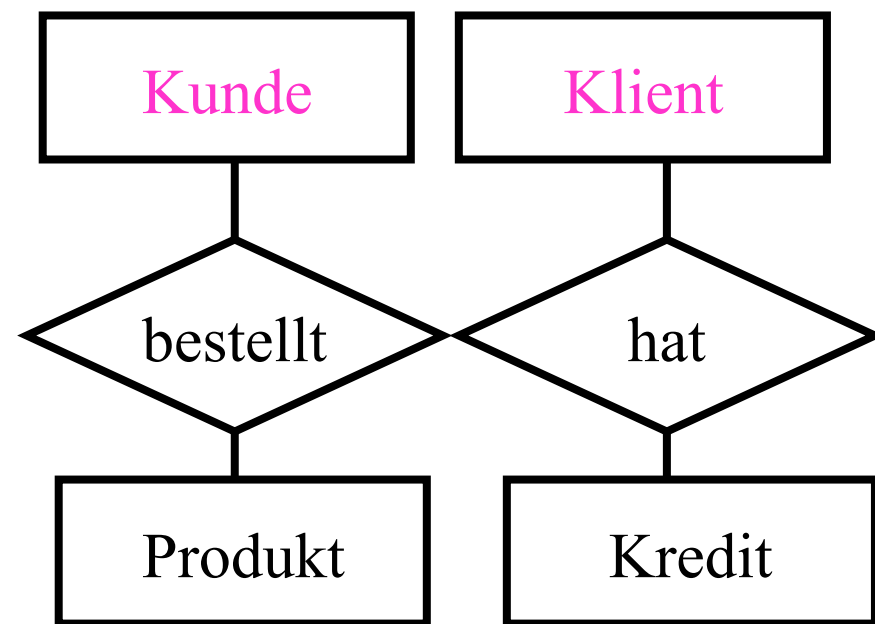
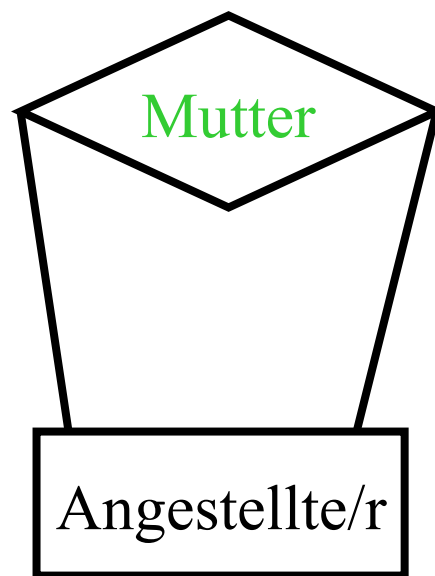
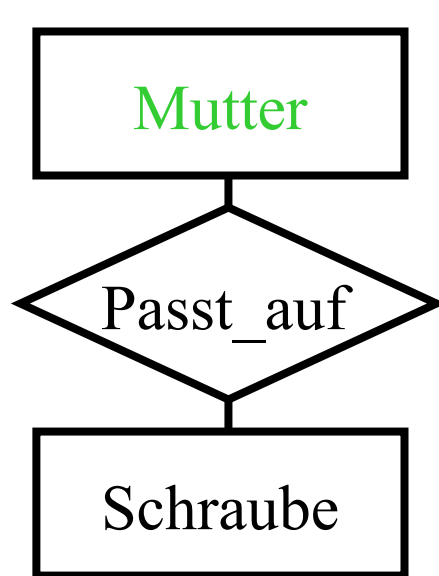
- Prä-Integrationsphase
 - Vorgehensweise festlegen
 - Information „sichten“
- Vergleichsphase
 - Namens-/Struktur- und Typ-Konflikte ermitteln
- Vereinheitlichungsphase
 - Festlegung des Zielschemas
- Restrukturierungs- und Zusammenfassungsphase
 - lokale Repräsentationsschemata festlegen
 - Abbildungen/transformationen realisieren

Vorgehensweise bei Integration binär vs. N-stellig



Vergleichsphase

- Namens- und Typkonflikte ermitteln
- **Homonyme** und **Synonyme**



Konflikte auf ...

- Strukturebene (Typeebene)
 - Typinkompatibilität
 - numerisch
 - alphanumerisch
 - Beziehungskonflikt (1:1 vs. N:M)
 - Schlüsselkonflikt
 - Verhaltenskonflikt (on delete cascade in nur einem Schema)
- Instanzebene
 - Mehrfachspeicherung mit unterschiedlichen Primärschlüsseln
 - Kunden und Lieferanten
 - Verwendung desselben Schlüsselwerts für unterschiedliche Entities
- funktionaler Ebene: unterschiedliche DB-Operatoren

Beispiel eines Schlüssel-Konflikts

Lieferanten@KnotenA		
<u>LiefNr</u>	Name	...
4711	Kölner AG	...
007	Bond Ltd	...
1984	BigBrother Inc	...
...

Lieferanten@KnotenB		
<u>LiefNr</u>	Name	...
1977	Kölner AG	...
007	Bond Ltd	...
1984	Orwell Inc	...
...

Konfliktresolution 1: Globale Schlüssel-Zuordnungstabelle

GlobaleLieferanten				
<u>GlobLiefNr</u>	Name	...	A_ID	B_ID
001	Kölner AG	...	4711	1977
002	Bond Ltd	...	007	007
003	Bigbrother Inc	...	1984	null
004	Orwell Inc	...	null	1984
...

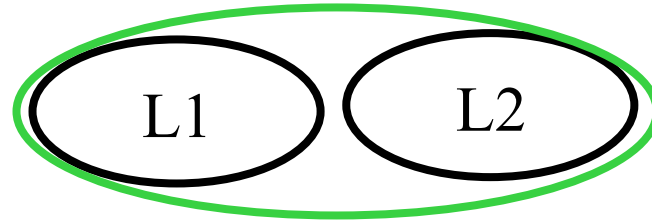
Konfliktresolution 2:

Globale Schlüssel-Zuordnungstabelle

GlobaleLieferanten			
<u>LiefNr</u>	<u>Knoten</u>	Name	...
4711	A	Kölner AG	...
1977	B	Kölner AG	...
007	A	Bond Ltd	...
007	B	Bond Ltd	...
1984	A	BigBrother Inc	...
1984	B	Orwell Inc	...

Änderungen auf post-integrierten Datenbanken

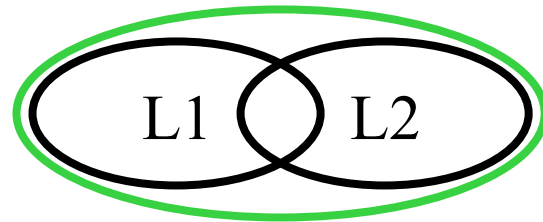
- ... Bei disjunkten lokalen DBs



- $\text{dom}(\text{PK_L1}) \cap \text{dom}(\text{PK_L2}) = \emptyset$
- jedem globalen Entity ist eineindeutig ein lokales Entity zugeordnet
- deshalb sind Updates prinzipiell möglich, d.h. propagierbar

Änderungen auf post-integrierten Datenbanken

- ... Bei überlappenden lokalen DBs



- $\text{dom}(\text{PK_L1}) \cap \text{dom}(\text{PK_L2}) \neq \emptyset$
- jedem globalen Entity können mehrere lokale Entities zugeordnet sein
- Problem: Einfügen neuer Entities. WO?
 - Überall, Primärknoten,...

Änderungen auf post-integrierten Datenbanken

- ... Bei vollständig überlappenden lokalen DBs

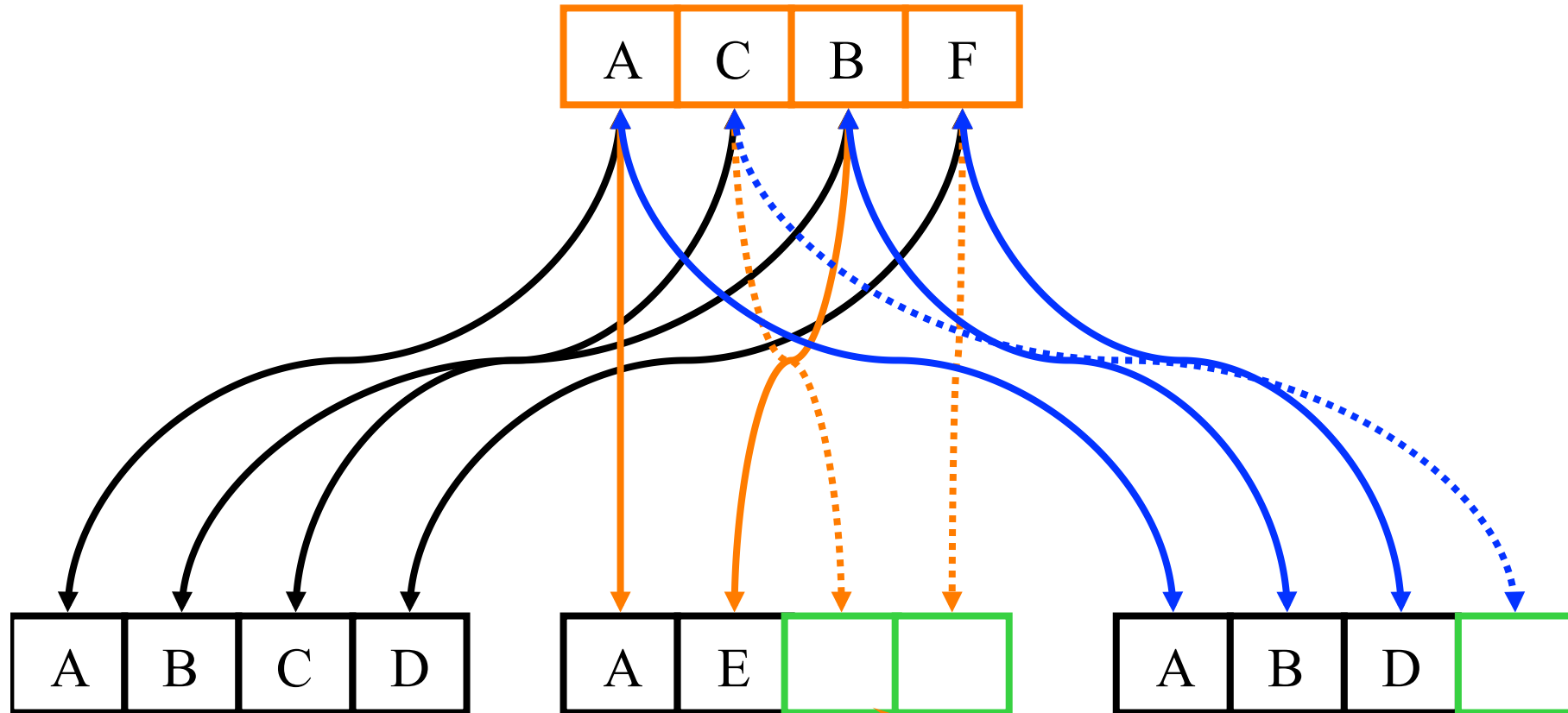


- $\text{dom}(\text{PK}_{L1}) \subseteq \text{dom}(\text{PK}_{L2})$
- Problemstellung wie bei partiell überlappenden lokalen DBs

Homogene, post-integrierte DBs

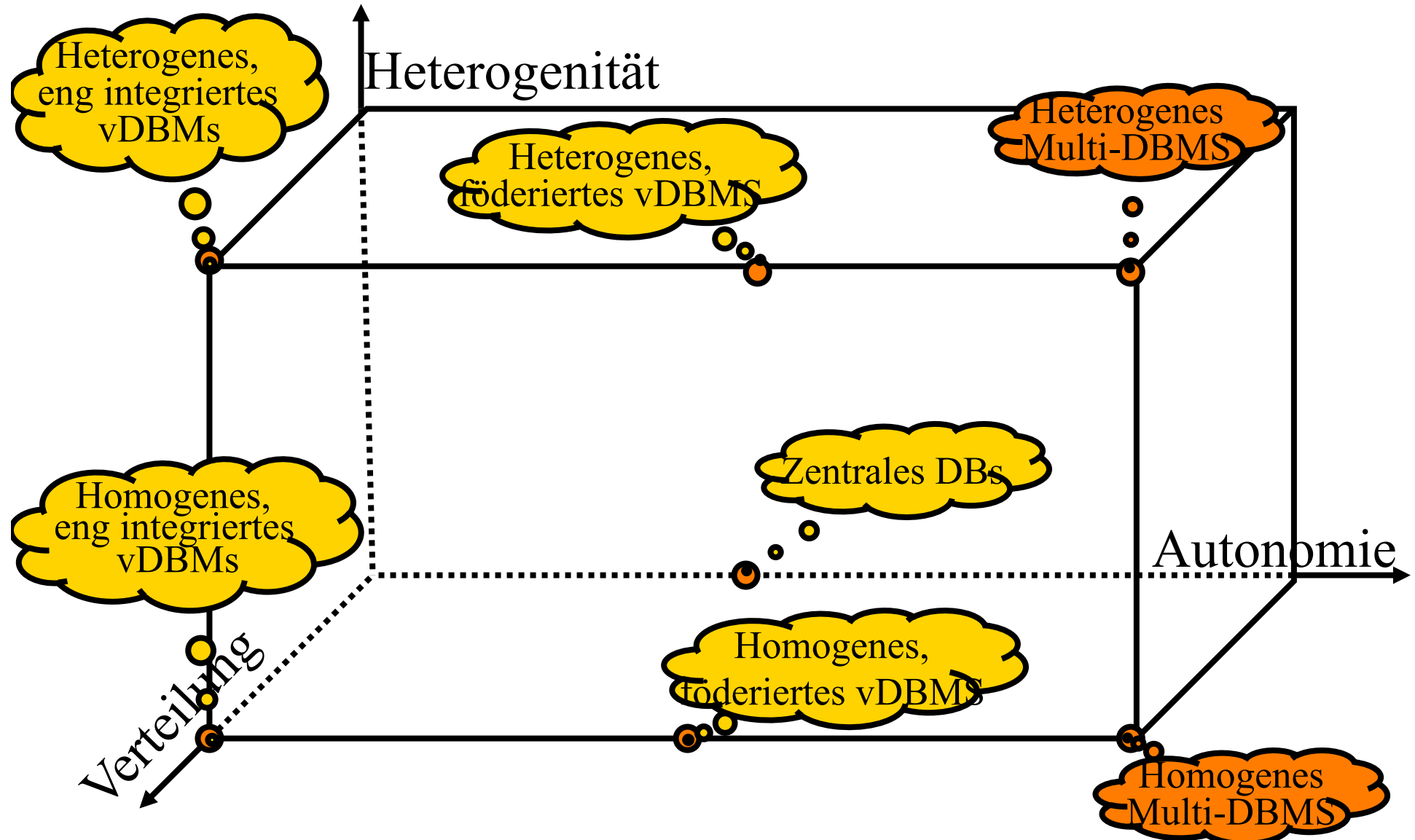
- Behandlung struktureller Heterogenität
- in einfachen Fällen durch Transformationsregeln in SQL/Relationenalgebra integrierbar
- Integration von Relationen unterschiedlicher Stelligkeit (\sim Breite)
 - Null-Werte
 - Problem: Einfügeoperation auf globaler Rel.
 - Dummy-Attributwerte einsetzen
- Problem unterschiedlicher Wertebereiche
 - numerisch vs. Alphanumerisch
 - Kunde007 vs 007

Illustration der Schema-Homogenisierung



Dummy Attribute
(lokal nicht vorhanden)

Informationsintegration in Multi-DBMS

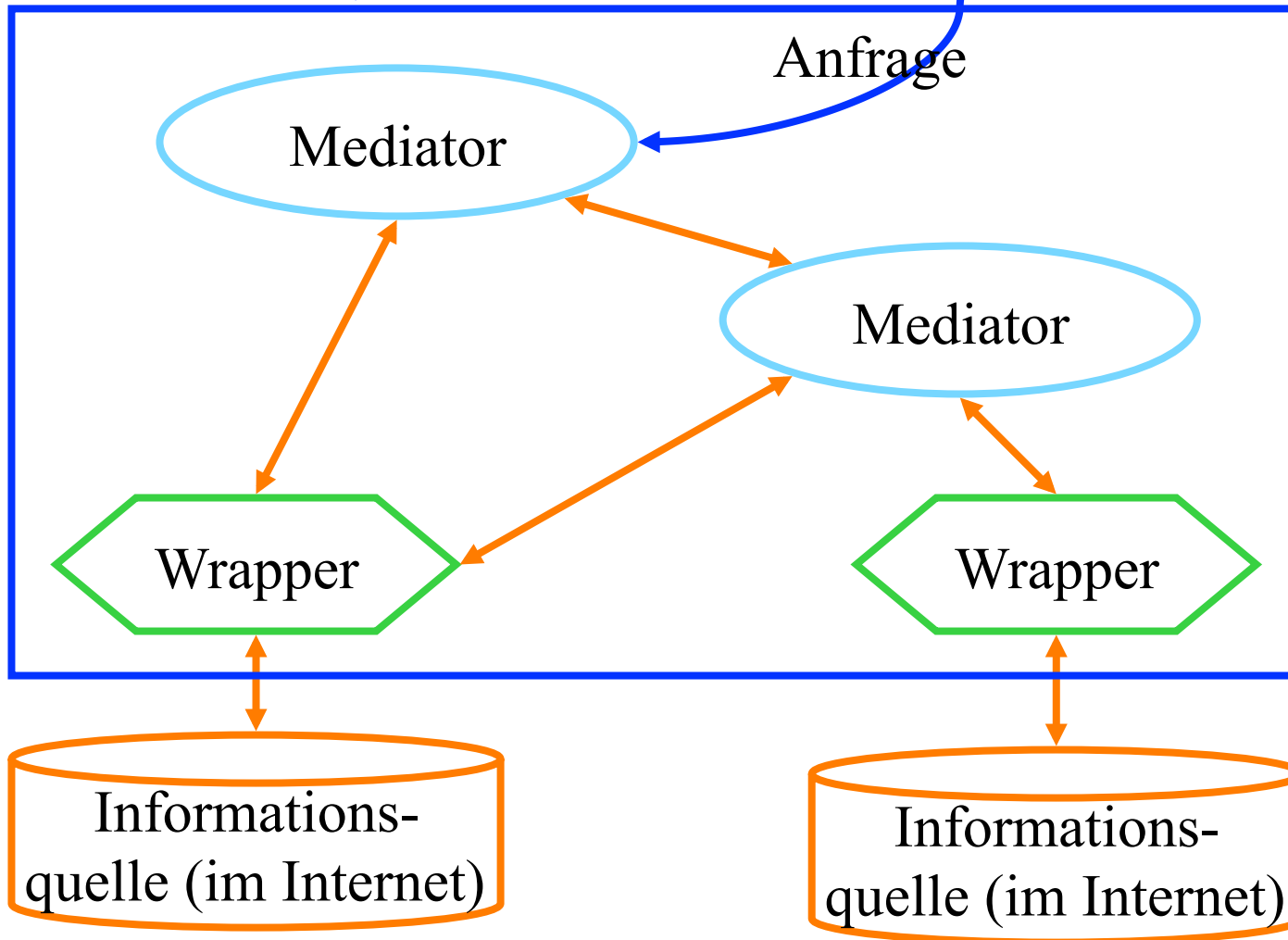
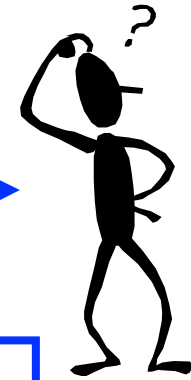


Problemstellung der Schemaintegration

- Keine Einflußnahme auf die jeweiligen Systeme möglich (also kein exportiertes Repräasentationsschema)
- Lokale Systeme „sträuben“ sich möglicherweise sogar gegen die Integration
 - Beispiel: Comparison Shopping
 - Cadabra.com
 - BottomDollar.com
 - mySimon.com
 - Junglee (wurde von Amazon gekauft [=eliminiert])

Middleware-System

Wrapper und Mediator



Beispiel-Systeme/Projekte

- Tsimmis
 - Stanford University
- Garlic
 - IBM
- Information Manifold
 - AT&T
- Fusion Tables
 - Google
- ObjectGlobe
 - Univ. Passau
 - dezentrale Architektur
- IDB
 - University of Wisconsin, Madison

Formale Grundlagen für Schemaintegration: **Datalog**

- **Database und Prolog**
- Logikprogrammiersprache bzw. logikbasierte Datenbank-Anfragesprache
- Ähnlichkeit zu Query by Example
- Basisrelationen = Fakten
 - extensionale Datenbank
- Regeln definieren Sichten/Views
 - intensionale Datenbank

Literatur zur formalen Schemaintegration

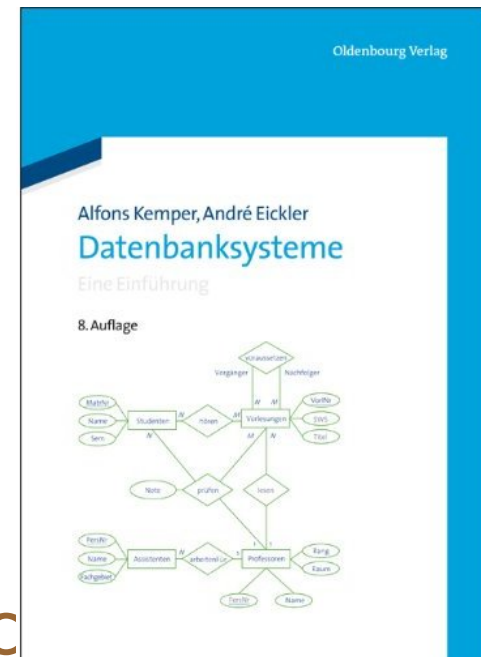
- Jeffrey D. Ullman: Information Integration Using Logical Views. ICDT 1997: 19-40
- Ulf Leser, Felix Naumann: Informationsintegration - Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen. dpunkt.verlag 2007: I-XIII, 1-464

Kurzeinführung zu Datalog

- Aus dem Kemper/Eickler-Buch
 - Kapitel 14

- <http://www-db.in.tum.de/research/publicDBMSeinf>

- <..\EIS\Deduktive.pptx>



Schema-Integration in Information Manifold

- Globale Prädikate

- emp(E)
- phone(E,P)
- office(E,O)
- mgr(E,M)
- dept(E,D)

Anfrage:

```
query(P,O) :- phone(sally,P) &  
              office(sally,O).
```

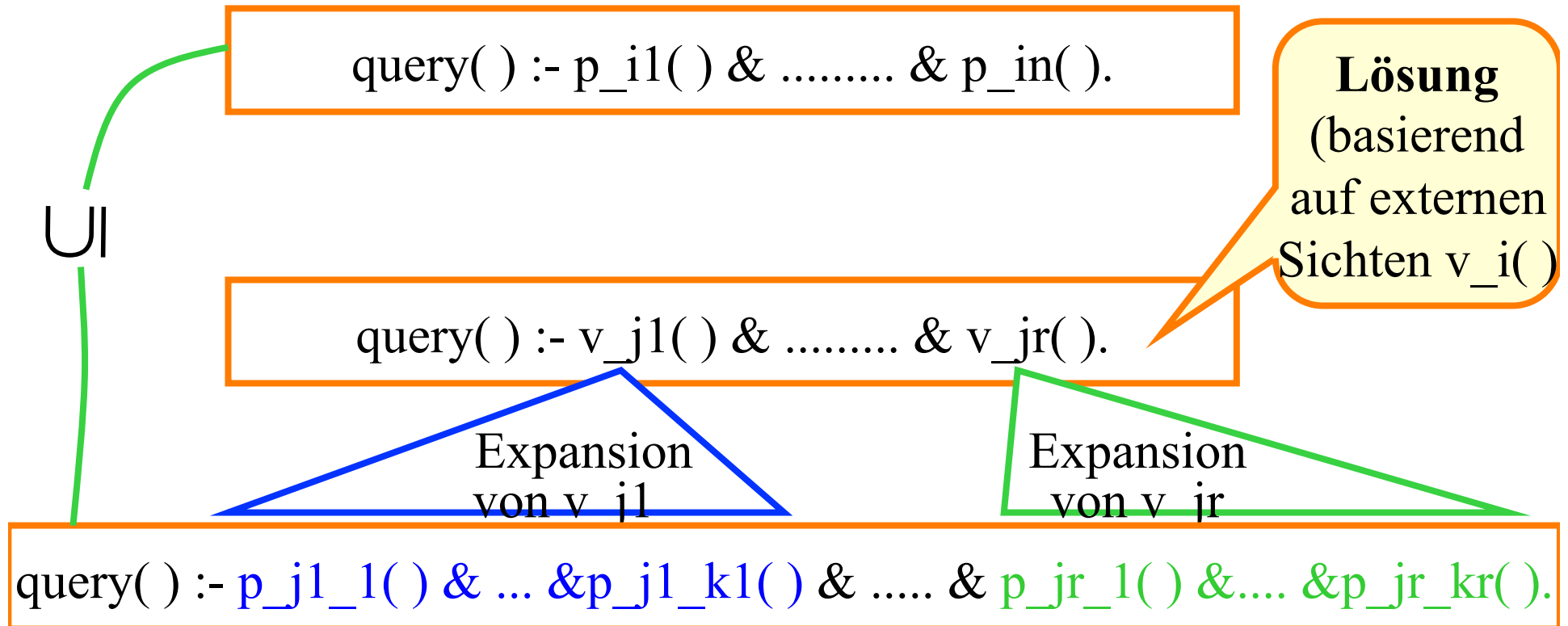
- Drei Datenquellen

- v1(E,P,M) :- emp(E) & phone(E,P) & mgr(E,M).
- v2(E,O,D) :- emp(E) & office(E,O) & dept(E,D).
- v3(E,P) :- emp(E) & phone(E,P) & dept(E,toy).

Schema-Integration in Information Manifold

- Drei Datenquellen
 - $v1(E,P,M) :- emp(E) \& phone(E,P) \& mgr(E,M).$
 - $v2(E,O,D) :- emp(E) \& office(E,O) \& dept(E,D).$
 - $v3(E,P) :- emp(E) \& phone(E,P) \& dept(E, toy).$

Anfrageauswertung in IM



Auswertung der Beispielanfrage

- `query(P,O) :- phone(sally,P) & office(sally,O).`
- Zwei Lösungen:
 - `query(P,O):-v1(sally,P,M) & v2(sally,O,D).`
 - `query(P,O):-v3(sally,P) & v2(sally,O,D).`
- Expansionen:
 - `query(P,O):-emp(sally) & phone(sally,P) & mgr(sally,M) & emp(sally) & office(sally,O) & dept(sally,D).`
 - `query(P,O):-emp(sally) & phone(sally,P) & dept(sally,toy) & emp(sally) & office(sally,O) & dept(sally,D).`

Mediatoren in Tsimmis

- Drei Datenquellen
 - $v1(E,P,M) :- emp(E) \& phone(E,P) \& mgr(E,M).$
 - $v2(E,O,D) :- emp(E) \& office(E,O) \& dept(E,D).$
 - $v3(E,P) :- emp(E) \& phone(E,P) \& dept(E, toy).$
- Mediatoren:
 - $epo(E,P,O) :- v1(E,P,M) \& v2(E,O,D).$
 - $epo(E,P,O) :- v3(E,P) \& v2(E,O,D).$
 - $edm(E,D,M) :- v1(E,P,M) \& v2(E,O,D).$
- Anfragen „gegen die Mediatoren“:
 - $query(P,O) :- epo(sally,P,O).$

Anfrageauswertung in Tsimmis

- Mediatoren:
 - $\text{epo}(E,P,O) :- v1(E,P,M) \ \& \ v2(E,O,D).$
 - $\text{epo}(E,P,O) :- v3(E,P) \ \& \ v2(E,O,D).$
 - $\text{edm}(E,D,M) :- v1(E,P,M) \ \& \ v2(E,O,D).$
- Anfragen „gegen die Mediatoren“:
 - $\text{query}(P,O) :- \text{epo}(\text{sally},P,O).$
- Expansion der Mediator-Definitionen:
 - $\text{query}(P,O) :- v1(\text{sally},P,M) \ \& \ v2(\text{sally},O,D).$
 - $\text{query}(P,O) :- v3(\text{sally},P) \ \& \ v2(\text{sally},O,D).$

Vergleich der beiden Ansätze: Integration neuer Datenquellen

- Information Manifold
 - Wrapper schreiben
 - Wrapper mittels externer Prädikate definieren
 - Ab dann werden neue Quellen automatisch mit berücksichtigt
- Local as view-Prinzip
- Tsimmis
 - Wrapper schreiben
 - Mediatoren „von Hand“ anpassen, damit die neuen Datenquellen mit berücksichtigt werden
- Global-as-view Prinzip

Anfragebearbeitung in Multi-DBMS-Systemen

- Später mehr ...
- erst beschäftigen wir uns mit der Anfragebearbeitung in homogenen, prä-integrierten DBMS



IDB: Towards the Scalable Integration of Queryable Internet Data Sources

- Namespaces zur Beschreibung von Datenobjekten
 - Namespace book#
 - vendor
 - category
 - title
 - url
 - author
 - year
 - price
 - isbn
 - publisher

```
book(vendor,category,title,url,author,...)
```

Wrapper „exportieren“ Attribute aus Namespaces

- Beispiel: Amazon-Wrapper

```
select `Amazon` as book#vendor, book#title  
  +,book#url, book#author, book#year, book#price  
  +,...
```

```
from http://www.amazon.com/ats-query ...  
  where <Wrapper-Implementierung>
```

Anfragen werden in Datalog transformiert

- **select** B.vendor, B.title, B.author, B.price, R.review
from book B, review R
where B.title $\sim =$ `Datenbanksysteme` and B.title = R.title
- $\text{query}^{\text{fbfff}}(\text{vendor}, \text{title}, \text{author}, \text{price}, \text{review}) :-$
 $\text{book}^{\text{fbff}}(\text{vendor}, \text{title}, \text{author}, \text{price}) \&$
 $\text{review}^{\text{bf}}(\text{title}, \text{review}) \&$
 $\text{title} \sim = \text{`Datenbanksysteme`}$

Datenquellen und Bindungen

■ Bücher

- amazon^{fb^bff}(vendor,title,author,year,price)
- borders^{fb^bff}(vendor,title,author,price)
- buch.de^{fb^bff}(vendor,title,price,isbn)
- bol^{fb}(author,isbn)

■ Reviews

- nytimes^{b^bff}(title,author,review)
- wpost^{b^bf}(title,isbn,review)

Anfragebearbeitung

- Es werden die Datenquellen herangezogen, die alle in der Anfrage referenzierten Attribute „liefern“ und
- mit den Bindungen der Anfrage verträglich sind

Berücksichtigung der Bindungen in der Anfrageoptimierung

Select b.Title, b.Author, b.ISBN, b.Price
From Books **as** b,

Bestseller **as** bs

where bs.Author = b. Author

Books

- buecher.de^{ffbf}(vendor,title,author,price)
- Bestseller
 - spiegel.de^{fff}(title,author,verlag)

