Informationssysteme

Sommersemester 2016

Prof. Dr.-Ing. Sebastian Michel
TU Kaiserslautern

smichel@cs.uni-kl.de

Wiederholung: Die Operatoren der relationalen Algebra

- Selektion σ
- Projektion π
- Kreuzprodukt $\times$
- Join (Verbund) $\bowtie$
- Umbenennung ρ
- Differenz $\setminus$
- Division $\div$
- Vereinigung $\cup$
- Schnitt $\cap$
- Semi-Join (linker) $\ltimes$
- Semi-Join (rechter) $\rtimes$
- linker äußerer Join $⟕$
- rechter äußerer Join $⟖$
- äußerer Join $⟗$

Dazu kommen später noch ein paar Erweiterungen.

Wiederholung: Grundlagen der relationalen Algebra

Es gibt **Relationen** (Ausprägungen: $R \subseteq D_1 \times D_2 \times ... \times D_n$) und **Relationenschema**:{[attributname1: datentyp1, attributname2: datentyp2, ...]}.

- Das Schema einer Relation wird auch mit $sch(R)$ oder $\mathcal{R}$ beschrieben.
- Relationen sind Mengen (keine Multimengen)
- Reihenfolge der Tupel sowie Reihenfolge der Attribute spielt keine Rolle (Attribute haben Namen).

Die Operatoren und ihre Verwendung:

- Eingabe eines Operators ist eine oder mehrere Relationen.
- **Ausgabe ist auch wieder eine Relation.**
- D.h. Operatoren sind kombinierbar (mit gewissen Regeln).

Die relationale Division

Wird für allquantifizierte Anfragen eingesetzt

Ein Tupel t ist in $R \div S$ falls für jedes $v \in S$ ein $u \in R$ existiert, so dass:

$$u.\mathcal{S} = v.\mathcal{S}$$

$$u.(\mathcal{R} \setminus \mathcal{S}) = t.(\mathcal{R} \setminus \mathcal{S})$$

Voraussetzung: $\mathcal{S} \subseteq \mathcal{R}$

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}\setminus\mathcal{S})}(R) \setminus \pi_{(\mathcal{R}\setminus\mathcal{S})}((\pi_{(\mathcal{R}\setminus\mathcal{S})}(R) \times S) \setminus R)$$

Die relationale Division: Beispiel

R	
M	V
m_1	v_1
m_1	v_2
m_1	v_3
m_1	v_4
m_2	v_1
m_2	v_2
m_3	v_1
m_3	v_3
m_4	v_3

S_1
V
v_1

S_2
V
v_1
v_2

S_3
V
v_1
v_2
v_3

$R \div S_1$
M
m_1
m_2
m_3

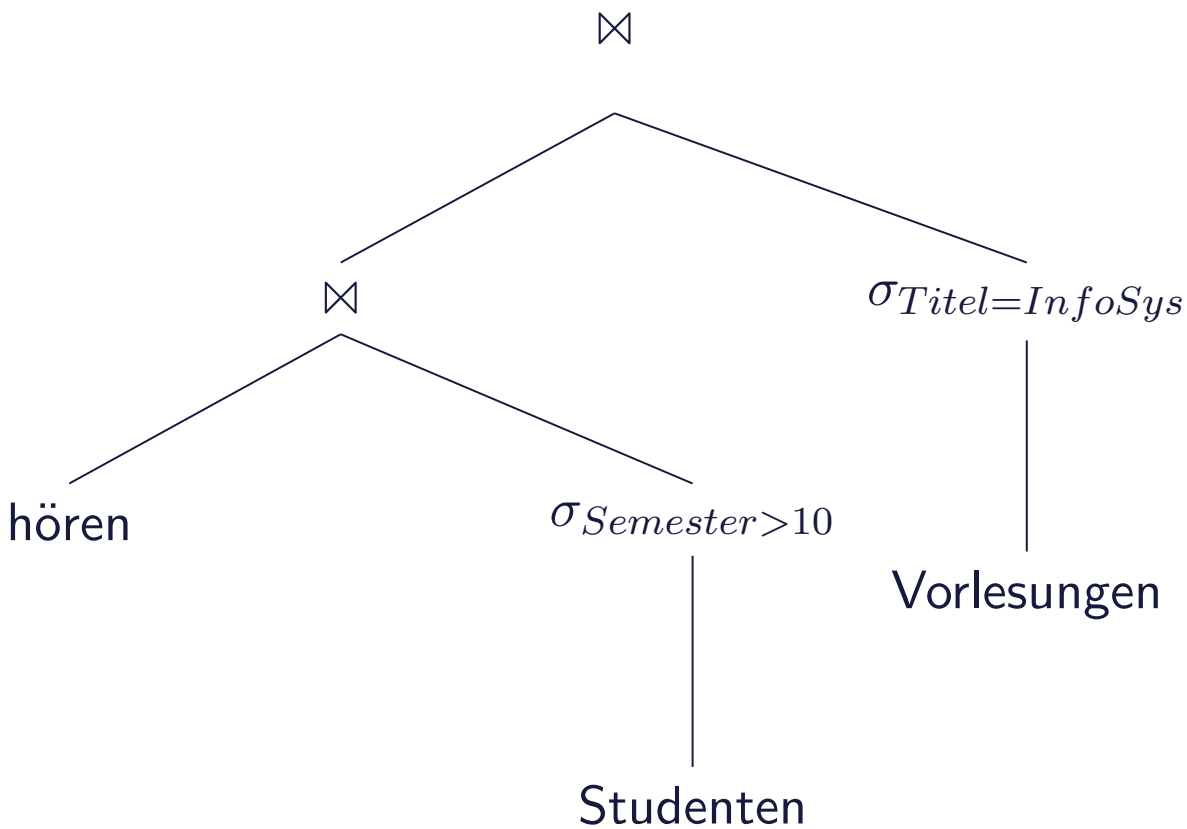
$R \div S_2$
M
m_1
m_2

$R \div S_3$
M
m_1

Operatorbaum-Darstellung

Alternative Darstellung von Ausdrücken der relationalen Algebra. Gerade für größere Ausdrücke viel übersichtlicher als "inline" Darstellung.

Beispiel:



Minimale Menge von benötigten Operatoren

Um alle Ausdrücke der relationalen Algebra ausdrücken zu können sind folgende Operatoren ausreichend:

- Projektion
- Selektion
- Kreuzprodukt
- Vereinigung
- Differenz
- Umbenennung

Wie kann z.B. der Mengendurchschnitt damit ausgedrückt werden?

Ausdrucksstärke

Theorem: Die folgenden Anfragesprachen besitzen die gleiche Ausdrucksstärke.

- Die **relationale Algebra**
- Das **relationale Tupelkalkül** eingeschränkt auf sichere Ausdrücke.
- Das **relationale Domänenkalkül** eingeschränkt auf sichere Ausdrücke.

Deklarative vs. Imperative Anfragesprachen

Im Gegensatz zur deklarativen (aka. deskriptiven) Formulierung von Anfragen in den beiden relationen Kalkülen bestehen die Anfragen in der relationalen Algebra aus expliziter Angabe von Berechnungsschritten in Form von Operatoren.

Für eine Anfrage in einem der Kalküle ist also nicht vorgegeben wie diese berechnet werden soll, in der relationalen Algebra hingegen schon.

Wir können für die relationale Algebra Regeln angeben wie die Operatoren innerhalb eines Ausdrucks verschoben werden können, bei Beibehaltung der Semantik der Anfrage.

Wir werden dies im Kapitel über Anfrageverarbeitung und Optimierung ausnutzen, um kostengünstigere "Anfragepläne" zu finden.

Äquivalenzerhaltende Transformationsregeln

1. **Aufbrechen von Konjunktionen im Selektionsprädikat**

$$\sigma_{c_1 \wedge c_2 \wedge ... \wedge c_n}(R) \equiv \sigma_{c_1}(\sigma_{c_2}(...(\sigma_{c_n}(R))...))$$

2. σ **ist kommutativ**

$$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R))$$

3. π**-Kaskaden**
   Falls $L_1 \subseteq L_2 \subseteq ... \subseteq L_n$, dann gilt

$$\pi_{L_1}(\pi_{L_2}(...(\pi_{L_n}(R))...)) \equiv \pi_{L_1}(R)$$

Äquivalenzerhaltende Transformationsregeln

4. **Vertauschen von σ und π**
   Falls die Selektion sich nur auf Attribute $A_1, ..., A_n$ der Projektionsliste bezieht, können die beiden Operationen vertauscht werden:

$$\pi_{A_1,...,A_n}(\sigma_c(R)) \equiv \sigma_c(\pi_{A_1,...,A_n}(R))$$

5. **$\cup, \cap$ und $\bowtie$ sind kommutativ**

$$R \bowtie_c S \equiv S \bowtie_c R$$

Äquivalenzerhaltende Transformationsregeln

6. **Vertauschen von σ mit $\bowtie$**
   Falls das Selektionsprädikat c nur auf Attribute der Relation R zugreift, kann man die beiden Operationen vertauschen:

$$\sigma_c(R \bowtie_j S) \equiv \sigma_c(R) \bowtie_j S$$

   Falls das Selektionsprädikat c eine Konjunktion der Form $c_1 \wedge c_2$ ist und c_1 sich nur auf Attribute aus R und c_2 sich nur auf Attribute aus S bezieht, gilt folgende Äquivalenz:

$$\sigma_c(R \bowtie_j S) \equiv \sigma_{c_1}(R) \bowtie_j \sigma_{c_2}(S)$$

Äquivalenzerhaltende Transformationsregeln

7. **Vertauschen von π mit $\bowtie$**
   Die Projektionsliste L sei: $L = \{A_1, ..., A_n, B_1, ..., B_m\}$, wobei A_i Attribute aus R und B_i Attribute aus S seien. Falls sich das Joinprädikat c nur auf Attribute aus L bezieht, gilt folgende Umformung:

$$\pi_L(R \bowtie_c S) \equiv (\pi_{A_1,...,A_n}(R)) \bowtie_c (\pi_{B_1,...,B_m}(S))$$

Äquivalenzerhaltende Transformationsregeln

8. **Die Operationen $\bowtie, \cap, \cup$ sind jeweils (einzeln betrachtet) assoziativ.** Wenn also Φ eine dieser Operationen bezeichnet, so gilt:

$$(R\Phi S)\Phi T \equiv R\Phi(S\Phi T)$$

9. **Die Operation σ ist distributiv mit $\cap, \cup, -$.** Falls Φ eine dieser Operationen bezeichnet, gilt:

$$\sigma_c(R\Phi S) \equiv (\sigma_c(R))\Phi(\sigma_c(S))$$

10. **Die Operation π ist distributiv mit $\cup$.**

$$\pi_c(R \cup S) \equiv (\pi_c(R)) \cup (\pi_c(S))$$

Äquivalenzerhaltende Transformationsregeln

11. **Die Join- und/oder Selektionsprädikate können mittels de Morgans Regeln umgeformt werden**

$$\neg(c_1 \wedge c_2) \equiv (\neg c_1) \vee (\neg c_2)$$

$$\neg(c_1 \vee c_2) \equiv (\neg c_1) \wedge (\neg c_2)$$

12. **Ein kartesisches Produkt, das von einer Selektionsoperation gefolgt wird, deren Selektionsprädikat Attribute aus beiden Operanden des kartesischen Produktes enthält, kann in eine Joinoperation umgeformt werden.**

Erweiterungen der Relationalen Algebra

Wir betrachten nun einige praxisrelevante Erweiterungen der relationalen Algebra.

- Multimengen
- Aggregation
- Verschiedene weitere Operatoren (Duplikateliminierung, Sortierung, erweiterte Projektion)

Gruppierung und Aggregation: Motivation

- Wie viele Studenten gibt es?
- Was ist die durchschnittliche Anzahl von Semestern?

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

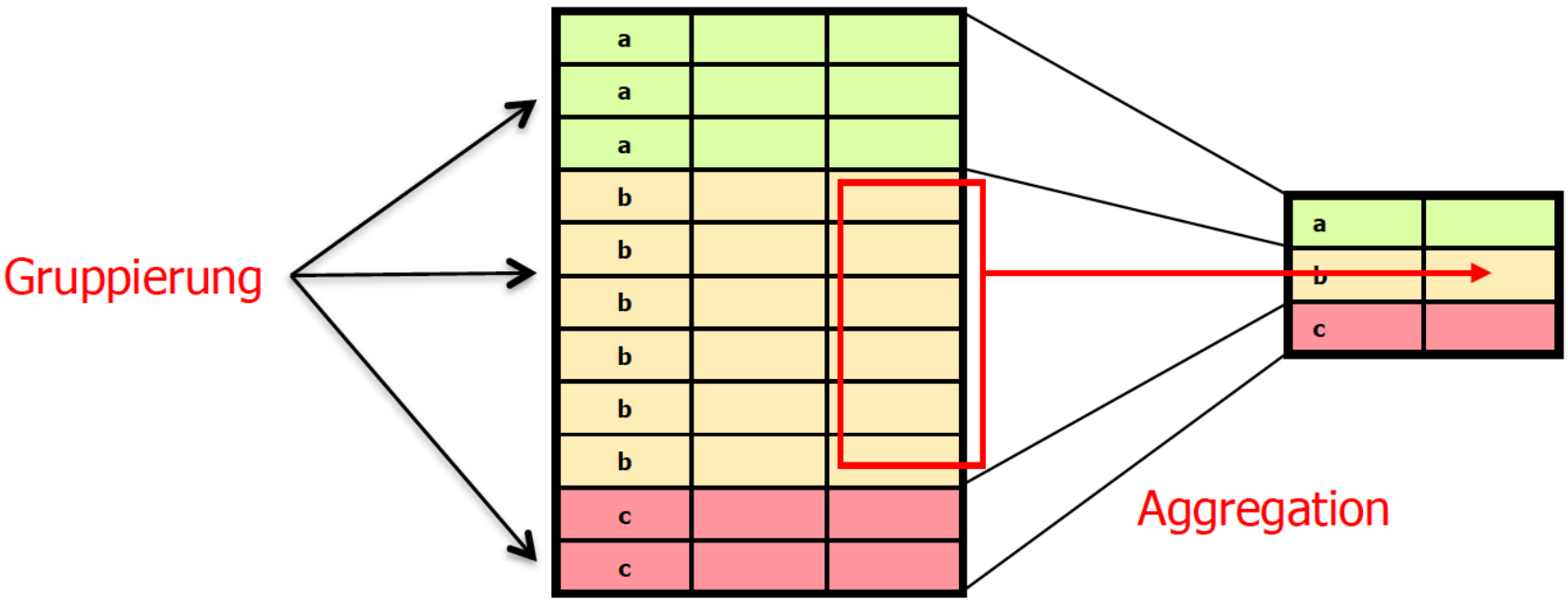
Gruppierung und Aggregation: Idee

Ermittlung von "verdichteten" Daten (Summe, Durchschnitt, ...)

Schritt 1 (optional): Bilde Gruppen von Tupeln aufgrund Wertegleichheit von Attributen, z.B. MatrNr.

Schritt 2: Wende für jede Gruppe Aggregatfunktion(en) an, z.B. AVG(Semester).

Ergebnis: Für jede Gruppe ein Tupel



Gruppierungsoperator

$$\gamma_{MatrNr,AnzahlVLs \leftarrow COUNT(VorlNr)}(hoeren)$$

Gruppierungsoperator γ ermöglicht Spezifikation von

- **Gruppierungsattributen** und **Aggregatfunktion** mit zu aggregierenden Attributen
- Gegeben Relation R mit Attributen $A = \{A_1, \ldots, A_n\}$
- Verwendung von Gruppierungs/Aggregationsoperator als

$$\gamma_{GA,AF}(R) \text{ mit}$$

  - Gruppierungsattributen $GA \subseteq A$
  - Aggregatfunktion $AF = AF_1, \ldots, AF_k$ mit

$$AF_i = AN_i \leftarrow (COUNT \mid SUM \mid AVG \mid MIN \mid MAX)(A_i)$$

  - AN_i ist Name des entstehenden Attributs (optional)
  - Zusätzlich COUNT(*) erlaubt, zum Zählen der Tupel einer Gruppe

Gruppierungsoperator (2)

Die Attribute in GA beschreiben wie die Tupel der Relation R gruppiert werden.

D.h. eine Gruppe besteht aus allen Tupeln aus R mit gleichen GA-Werten

Ergebnisrelation:

- Enthält für jede vorkommende Wertekombination(!) von GA ein Tupel
- Dieses Tupel besitzt GA-Werte und berechnete AF-Werte für jede Gruppe

Falls GA=$\emptyset$ wird auf der gesamten Relation (eine große Gruppe) aggregiert. Ergebnis besteht in diesem Fall nur aus den berechneten AF-Werten (1 Tupel). Und GA=$\emptyset$ wird auch einfach weggelassen in der Anfrage. Beispiele: $\gamma_{COUNT(*)}(Studenten)$ oder $\gamma_{SUM(SWS)}(Vorlesungen)$

Gruppierungsoperator: Beispiel mit Zählen (Count)

hoeren	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

$\gamma_{MatrNr,AnzahlVLs \leftarrow COUNT(VorlNr)}(hoeren)$

MatrNr	AnzahlVLs
26120	1
25403	1
27550	2
28106	4
29120	3
29555	2

Der Erweiterte Projektions-Operator

Wo in der ursprünglichen Definition der Projektion nur Attributnamen angegeben werden, also $\pi_{A_1,\ldots,A_k}(R)$ **lassen wir in der Erweiterung des Projektions-Operators auch Funktionen zu, die ein oder mehrere Attribute als Argumente haben.**

Ausgabe der Funktionen können benannt werden.

Beispiel:

R

A	B	C
1	2	3
1	3	4
3	2	1
4	4	4

$\pi_{X \leftarrow A+B, Y \leftarrow A*1.19, C}(R)$

X	Y	C
3	1.19	3
4	1.19	4
5	3.57	1
8	4.76	4

Sortierung

$$\tau_L(R)$$

ist eine **Liste** von Tupeln aus Relation R, geordnet nach den Attributen, die in L genannt sind. In lexikographischer Ordnung. Optional mit Angabe der Sortierung: Absteigend oder Aufsteigend, pro Attribut. Default: Aufsteigend.

$\tau_{A,B}(R)$

A	B	C
1	2	3
1	3	4
3	2	1
4	4	4

$\tau_{B,C}(R)$

A	B	C
3	2	1
1	2	3
1	3	4
4	4	4

$\tau_{C\downarrow}(R)$

A	B	C
1	3	4
4	4	4
1	2	3
3	2	1

Duplikate in den Daten: Multimengen (Bags)

Die bislang betrachteten Anfragesprachen arbeiten auf Mengen, im Sinne von Relationen, die Mengen von Tupeln sind.

Es werden bei dieser Mengen-Betrachgung dabei implizit Duplikate entfernt. Insbesondere nach einer Projektion treten oft Duplikate auf.

Datenbanksysteme arbeiten in der Regel mit **Multimengen** (aka. **Bags**), d.h. die Relationen (Tabellen) dürfen auch Duplikate enthalten.

Zum Beispiel sind {a,b,b} und {a,b} Multimengen, wobei letztere auch eine Menge ist.

Operationen auf Multimengen

Selektion, Projektion, Join, Kreuzprodukt, Umbenennung (sowieso), Aggregation/Gruppierung, Sortierung funktionieren auch für Multimengen.

Was ist mit Mengendifferenz, Mengendurchschnitt, Mengenvereinigung für Multimengen?

Operationen auf Multimengen: Selektion

$\tau_{A,B}(R)$

A	B	C
1	2	5
3	4	6
1	2	7
1	2	7

$\sigma_{C \geq 6}(R)$

A	B	C
3	4	6
1	2	7
1	2	7

Auf jedes Tupel wird Prädikat der Selektion angewendet. Sich qualifizierende Tupel werden in Ergebnis aufgenommen. Duplikate werden dabei nicht eliminiert.

Operationen auf Multimengen: Kreuzprodukt

Jedes Tupel aus der einen Relation wird mit allen Tupeln der anderen Relation gepaart. Egal ob Duplikat oder nicht. Tritt ein Tupel r m mal in R auf und ein Tupel s tritt n mal in S auf, dann gibt es da Tupel rs in $R \times S$ genau mn mal.

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \times S$

A	R.B	S.B	C
1	2	2	3
1	2	2	3
1	2	4	5
1	2	4	5
1	2	4	5
1	2	4	5

Operationen auf Multimengen: Natürlicher Join

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \bowtie S$

A	B	C
1	2	3
1	2	3

Tupel $(1,2)$ aus R wird mit $(2,3)$ aus S verknüpft. Da es zwei Kopien von $(1,2)$ in R gibt und eine Kopie von $(2,3)$ in S, gibt es zwei Paare von joinbaren Tupel, also **zwei Kopien von $(1,2,3)$ im Ergebnis**.

Operationen auf Multimengen: Theta Join

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \bowtie_{R.B<S.B} S$

A	R.B	S.B	C
1	2	4	5
1	2	4	5
1	2	4	5
1	2	4	5

Operationen auf Multimengen: Kardinalität und $\subseteq$

Für ein Bag B bezeichnen wir mit $\#B(x)$ die Anzahl der Auftreten von Element x in B (aka. Multiplizität).

Wir haben dann

- **Kardinalität** von Bag B als $|B| = \sum_x \#B(x)$
- $A \subseteq B$ als $\#A(x) \leq \#B(x)\ \forall x$

Operationen auf Multimengen: Vereinigung, Differenz, Schnittmenge

- **Vereinigung:** $\#(X \cup Y)(z) = \#X(z) + \#Y(z)$
- **Schnittmenge:** $\#(X \cap Y(z)) = min(\#X(z), \#Y(z))$
- **Differenz:** $\#(X \setminus Y)(z) = max(0, \#X(z) - \#(Y))$

Anmerkung: Die Definition der Vereinigung für Bags ist nicht identisch zu der Vereinigung von Bags aus der Mathematik, welche definiert wäre als $\#(X \cup_{max} Y)(z) = max(\#X(z), \#Y(z))$. Mit dieser Definition würden auch die Regeln für Mengen bei Bags gelten.

Die oben angegebene Definition $\#(X \cup Y)(z) = \#X(z) + \#Y(z)$ folgt jedoch genau dem was in DB-Systemen (siehe später SQL) oder Programmiersprachen benutzt wird, im Sinne der "**Konkatenation**" von Bags. $X \cup_{max} Y \equiv (X \setminus Y) \cup Y$

Beispiel: Verletzte Regel für Bags

$R \cap (S \cup T) \equiv (R \cap S) \cup (R \cap T)$ gilt für Mengen, aber nicht für Bags

Seien R, S und T jeweils das Bag $\{1\}$
Dann ist die linke Seite: $S \cup T = \{1,1\}$; $R \cap (S \cup T) = \{1\}$.

Und die rechte Seite: $R \cap S = R \cup T = \{1\}$
$(R \cap S) \cup (R \cap T) = \{1\} \cup \{1\} = \{1,1\} \neq \{1\}$

Operator für Duplikateliminierung

$$\delta(R)$$

ergibt eine Relation, in der jedes Tupel einmal auftritt für Tupel die in R einmal oder mehrfach auftreten.

R

A	B
1	2
3	4
1	2
1	2

$\delta(R)$

A	B
1	2
3	4

δ als Spezialfall von γ

Technisch gesehen ist der Duplikatseliminierungsoperator δ redundant. Für Relation $R(A_1,A_2,\ldots,A_n)$ ist die Relation $\delta(R)$ äquivalent zu $\gamma_{A_1,A_2,\ldots,A_n}(R)$.

- D.h. um Duplikate zu eliminieren gruppieren wir nach allen Attributen, **ohne Anwendung einer Aggregatfunktion.**
- Also jede Gruppe besteht aus einem Tupel, welches einmal oder mehrfach in R aufgetreten ist.

Für Mengen (keine Multimengen/Bags) ist γ auch eine Erweiterung der Projektionsoperators. D.h. $\gamma_{A_1,A_2}(R)$ ist identisch zu $\pi_{A_1,A_2}(R)$ falls R eine Menge ist.
Falls aber R eine Multimenge ist so eliminiert γ Duplikate und π nicht.