

Datenbanken
Wintersemester 2015/16
 Prof. Dr. W. May

2. Übungsblatt: Algebra

Besprechung voraussichtlich am 8./15.12.2015

Aufgabe 1 (Relationale Anfragen an Mondial: Bedingungen) Geben Sie Ausdrücke der relationalen Algebra für die folgenden Anfragen an die Mondial-Datenbank an:

- a) Die Namen aller Städte, die mehr als 1.000.000 Einwohner haben.
- b) Die Namen aller Städte, die mehr Einwohner als Neuseeland haben.
- c) Die Namen aller Städte, in denen mehr als 25% der Bevölkerung des jeweiligen Landes leben.

Für spätere Übungsblätter:

- Geben Sie dieselben Anfragen in SQL an.
- Geben Sie dieselben Anfragen im relationalen Kalkül an.

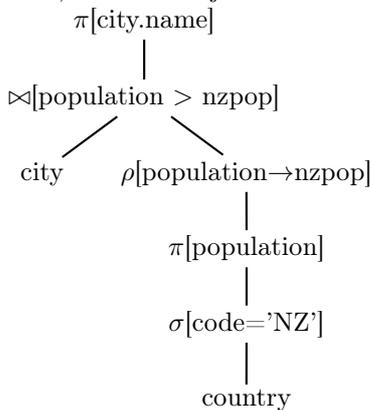
a) Einfache Selektion:

$$\pi[\text{name}](\sigma[\text{population} > 1000000](\text{city}))$$

b) Die Bevölkerungszahl von Neuseeland bekommt man als

$$\pi[\text{population}](\sigma[\text{code}='NZ'](\text{country})) .$$

In der Selektionsbedingung in Teil (1) ist aber nur eine Konstante erlaubt. Man benötigt ein Join, in dem man jedes Land mit dem Ergebnis der Subquery joint und den Vergleich durchführt:



Anstatt des “vollen” Join kann man auch ein Semijoin nehmen, da das Ergebnis des rechten Teilbaumes nur zum Vergleich verwendet wird, und keine Werte daraus weiter oben noch gebraucht werden.

Hinweis: in SQL kann man die Subquery in die WHERE-Klausel einbauen:

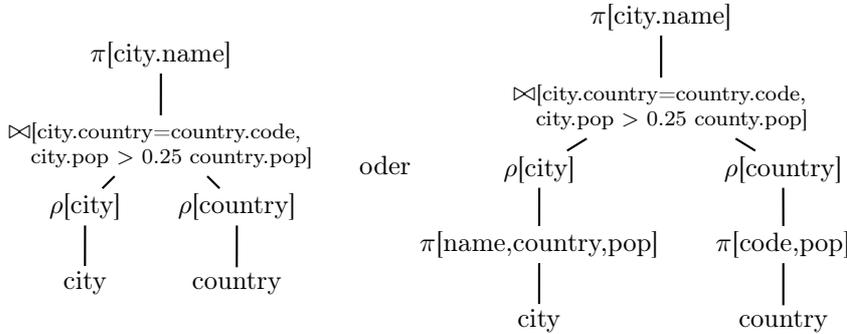
```

SELECT name
FROM city
WHERE population > (SELECT population FROM country WHERE code='NZ')
```

oder als Join formulieren:

```
SELECT name
FROM city, (SELECT population as nzpop FROM country WHERE code='NZ')
WHERE population > nzpop
```

c) Korreliertes Join mit Zusatzbedingung: Sei allgemein $\rho[alias]$ die Umbenennung, die alle Attribute mit dem Präfix *alias* qualifiziert (z.B. für *city.name*).



Aufgabe 2 (Äquivalenz von Ausdrücken) Gegeben seien folgende Relationen:

- $R(A,B,C)$
- $S(A,E,F)$
- $T(A,H)$

Die Wertebereiche aller nicht namensgleichen Attribute seien voneinander verschieden. Gegeben sei nun folgender relationaler Ausdruck:

$$\pi[E, H](\sigma[B = 10]((R \bowtie T) \bowtie S))$$

Sind die folgenden Ausdrücke äquivalent zu obigem Ausdruck? Begründen Sie Ihre Antwort.

- $\pi[E, H](\sigma[B = 10](R)) \bowtie (\pi[A, E](S) \bowtie T)$
- $\pi[E, H](\sigma[B = 10](((\pi[B](R)) \bowtie (\pi[A, E](S))) \bowtie (\pi[A, H](T))))$
- $\pi[E, H](\pi[A, B](\sigma[B = 10](R))) \bowtie ((\pi[A](S)) \bowtie T)$

a) ist äquivalent:

Der ursprüngliche Ausdruck:

$$\pi[E, H](\sigma[B = 10]((R \bowtie T) \bowtie S))$$

Join ist assoziativ:

$$\pi[E, H](\sigma[B = 10](R \bowtie (T \bowtie S)))$$

Attribut *B* existiert nur in Relation *R*, daher kann Selektion vor Join ausgeführt werden:

$$\pi[E, H](\sigma[B = 10]R \bowtie (T \bowtie S))$$

Von *S* werden nur die Attribute *A* (im Join mit *T*) und *E* (in der abschließenden Projektion) benötigt, man kann die Projektion also auch gleich auf *S* ausführen:

$$\pi[E, H](\sigma[B = 10]R \bowtie (T \bowtie (\pi[A, E]S)))$$

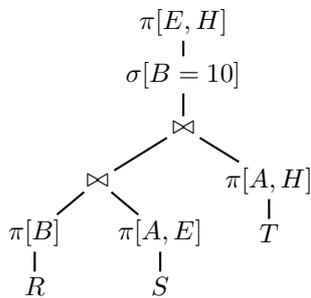
Join ist auch kommutativ:

$$\pi[E, H](\sigma[B = 10]R \bowtie ((\pi[A, E]S) \bowtie T))$$

Der letzte Ausdruck zeigt dass die Äquivalenz gilt.

Hinweis: in diesem Ausdruck werden alle π und σ so früh wie möglich ausgeführt.

b) nicht äquivalent:



$\pi[B](R)$ eliminiert das für den Join mit S und T wichtige Attribut A , hier wird stattdessen das Kreuzprodukt ausgeführt.

$\pi[A, E](S)$ ist wie in a) gezeigt zulässig – F wird nicht gebraucht.

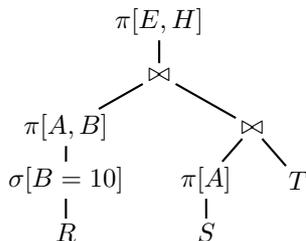
$\pi[A, H](T)$ ist nutzlos (Identität), stört aber keinen.

Der Ausdruck ist syntaktisch in Ordnung, hat aber ein anderes Ergebnis (eine Obermenge, da ein Join-Kriterium wegfällt).

Dazu gibt man dann ein (kleines) Beispiel an, z.B. einen Datenbankzustand \mathcal{S} mit $\mathcal{S}(R) = \{(a_0, b, c)\}$, $\mathcal{S}(S) = \{(a_1, e, f)\}$, $\mathcal{S}(T) = \{(a_1, h)\}$.

Die Auswertung des ersten Ausdrucks, $Q_0(\mathcal{S}) = \emptyset$, während die Auswertung des Ausdrucks in (a), $Q_a(\mathcal{S}) = \{(\langle \rangle, \langle \rangle)\}$ ergibt.

c) nicht äquivalent:



$\pi[A](S)$ eliminiert das Attribut E , welches in der Ausgabe enthalten sein soll. Der Ausdruck ist damit syntaktisch nicht zulässig.

Aufgabe 3 (Relationale Anfragen an Mondial: Schweizer Sprachen) Geben Sie Ausdrücke der relationalen Algebra für die folgenden Anfragen an die Mondial-Datenbank an:

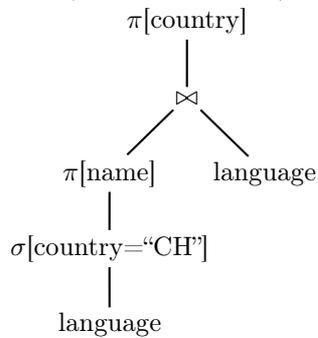
- Alle Landescodes von Ländern, in denen eine Sprache gesprochen wird, die auch in der Schweiz gesprochen wird.
- Alle Landescodes von Ländern, in denen ausschliesslich Sprachen gesprochen werden, die in der Schweiz nicht gesprochen werden.
- Alle Landescodes von Ländern, in denen nur Sprachen gesprochen werden, die auch in der Schweiz gesprochen werden.
- Alle Landescodes von Ländern, in denen alle Sprachen gesprochen werden, die in der Schweiz gesprochen werden.

Für spätere Übungsblätter:

- Geben Sie dieselben Anfragen in SQL an.
- Geben Sie dieselben Anfragen im relationalen Kalkül an.

Sei allgemein $\rho[alias]$ die Umbenennung, die alle Attribute mit dem Präfix *alias* qualifiziert (z.B. für *city.name*).

- a) Join zur Formulierung einer "Auswahlbedingung". Verwende Tabelle *language(country,name,percent)*, z.B. (CH, "german", 65).

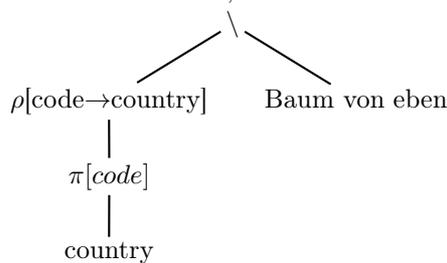


Der linke Ast ergibt die Tabelle

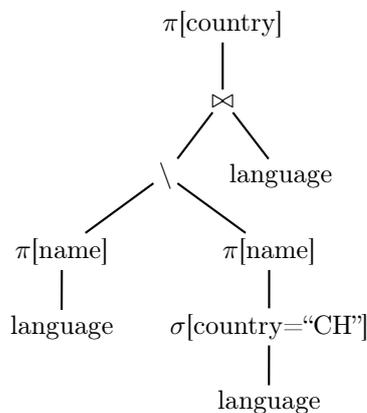
language
name
german
french
italian
romansch

Diese wird mit *language* gejoint (Join-Attribut "name"), womit genau die Einträge aus *language* übrigbleiben, die eine der aufgeführten Sprachen enthalten.

- b) Genau alle Länder, die eben nicht aufgezählt wurden:



- c) Löst man in mehreren Schritten. Erst alle Sprachen bestimmen, die nicht in CH gesprochen werden. Das Ergebnis wird mit *language* gejoint, und übrig bleiben die Einträge, die nicht-schweizer Sprachen beschreiben. Die Länder (bzw. Landescodes), in denen irgendeine nicht-schweizer Sprache gesprochen wird, bekommt man also folgendermassen:

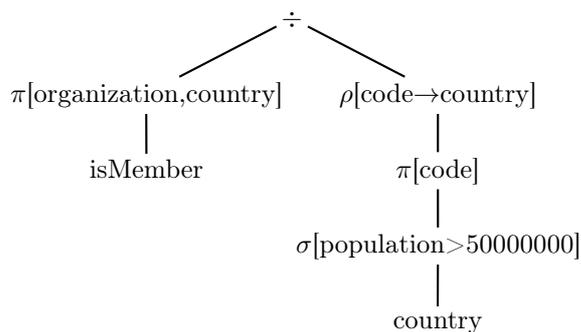


Bleibt nun, die Komplementmenge, eben die Länder in denen nur schweizer Sprachen gesprochen werden, zu bestimmen:

Hinweis: man sieht in SQL immer wieder Lösungen, die die Division auf die Mächtigkeit der Menge zurückführen; hier würde man einfach zählen, welche Organisation auf sovielen Kontinenten wie es gibt, vertreten ist. In diesem Fall wäre das richtig. Die gilt jedoch nicht für Teil (2) der Aufgabe:

```
select abbreviation from organization o
where
(select count(distinct continent)
 from ismember i, encompasses e
 where i.country = e.country
       and i.organization = o.abbreviation
 group by i.organization)
 = (select count (*) from continent)
```

b)



```
select name from organization o
where not exists
(select *
 from country
 where population > 50000000
       and code not in (select country
                        from ismember
                        where organization = o.abbreviation))
```

Hier würde ein Zählen der Mitgliedsländer nichts bringen, da es viele Länder gibt, die die zusätzliche Bedingung > 1000000 nicht erfüllen. Man könnte allenfalls die Mitgliedsländer zählen, die mehr als 1000000 Einwohner haben - womit die Anfrage aber auch nicht einfacher würde.

Aufgabe 5 (Division mit Basisoperationen) Beweisen Sie, daß die in der Vorlesung angegebene Darstellung der Division durch relationale Basisoperatoren als

$$r \div s = \pi[Z](r) - \pi[Z]((\pi[Z](r) \bowtie s) - r)$$

mit $r(X)$, $s(Y)$ und $Z = X \setminus Y$ äquivalent zu der gegebenen Definition

$$r \div s = \{\mu \in \text{Tup}(Z) \mid \{\mu\} \bowtie s \subseteq r\}$$

ist.

Veranschaulichen Sie sich Ihre Überlegungen anhand des Beispiels "Geben Sie die Namen derjenigen Organisationen an, die auf jedem Kontinent mindestens ein Mitglied haben".

Nach Definition der Division ist

$$r \div s = \{\mu \in \text{Tup}(Z) \mid \{\mu\} \bowtie s \subseteq r\} \quad (*)$$

Betrachte

$$r \div s = \pi[Z](r) - \underbrace{\pi[Z](\pi[Z](r) \bowtie s) - r}_{(***)} \quad (**)$$

Im Beispiel ist r die Menge aller Paare ($org, cont$) so dass Organisation o ein Mitglied auf Kontinent $cont$ hat. s ist die Menge der (Namen der) Kontinente. $r \div s$ ist das gesuchte Ergebnis.

Im allgemeinen Fall (z.B. "... auf allen Kontinenten mit mehr als 10.000.000 km² Fläche") kann r auch Tupel mit Y -Werten (Kontinenten) enthalten, die in $\{\mu\} \bowtie s$ nicht gefordert werden. Die Division prüft nur, ob die geforderten Kombinationen vorhanden sind.

Vorwärtsbeweis

- (1) $(**) \subseteq (*)$: Korrektheit des Ausdrucks. Alle Ergebnisse erfüllen die Charakterisierung:

Sei $\mu \in (**)$.

μ ist also in $\pi[Z](r)$, also $\in \text{Tup}(Z)$ – soweit nicht besonders interessant.

Interessanter ist der zweite Teil, der aussagt, dass μ nicht in $\pi[Z](\pi[Z](r) \bowtie s) - r$ ist. μ ist also kein Z -Tupel (im Beispiel: keine Organisation), so dass bei $(\{\mu\} \bowtie s) - r$ ein Rest übrigbleibt. D.h., $\{\mu\} \bowtie s \subseteq r$, und damit ist μ ein Tupel, das $(*)$ erfüllt.

- (2) $(*) \subseteq (**)$: Vollständigkeit der Charakterisierung – Ergebnis enthält alle geforderten Werte:

Sei μ ein Tupel über Z (Beispiel: eine Organisation), $\mu \in (*)$, also $\{\mu\} \bowtie s \subseteq r$.

Damit offensichtlich $\mu \in \pi[Z](r)$; jetzt also noch zu zeigen, dass μ nicht in $\mu \notin (***)$ ist.

Aus $\{\mu\} \bowtie s \subseteq r$ folgt, dass $(\{\mu\} \bowtie s) - r$ leer ist. Damit ist auch μ nicht in $\pi[Z](\pi[Z](r) \bowtie s) - r$, für beliebige Mengen $t \subseteq \text{Tup}Z$, also nicht in $(***)$.

Hinweis: Man könnte anstelle $(**)$ auch

$$r \div s = \pi[Z](r) - \underbrace{\pi[Z](\text{Tup}(Z) \bowtie s) - r}_{(***)} \quad (*)$$

schreiben. Damit müsste man allerdings eine unendliche Menge von Werten überprüfen. Durch die Einschränkung auf $\pi[Z](r)$ erhält man eine endliche Ausgangsmenge. (Im Beispiel entspräche $\text{Tup}(Z)$ der Menge aller möglichen Zeichenketten als Namen von Organisationen, die nicht mit jedem Kontinent in r auftreten – natürlich kommen dabei als Antworten nur solche in Frage, die in $\pi[Z](r)$ überhaupt enthalten sind).

Alternative: Widerspruchsbeweis (verwendet im Prinzip natürlich dieselbe Argumentation):

- (1) $(**) \subseteq (*)$: Korrektheit des Ausdrucks. Alle Ergebnisse erfüllen die Charakterisierung:

Sei $\mu \in (**)$.

Annahme: $\mu \notin (*)$, d.h. $\{\mu\} \bowtie s \not\subseteq r$.

Dann gibt es also ein $\nu \in s$, so dass $\mu\nu \notin r$.

$\mu \in \pi[Z](r)$ nach Konstruktion von $(**)$

Schauen wir uns wieder $(***)$ an: Da $\mu \in \pi[Z](r)$ ist, ist $\mu\nu \in \pi[Z](r) \bowtie s$ und ja $\notin r$, also $\mu\nu \in (\pi[Z](r) \bowtie s) - r$ und damit $\mu = \pi[Z](\text{diesem}) = (***)$.

Womit μ nicht in $(*)$ ist. Widerspruch zur Voraussetzung.

- (2) $(*) \subseteq (**)$: Vollständigkeit der Charakterisierung – Ergebnis enthält alle geforderten Werte:

Sei $\mu \in (*)$, also $\{\mu\} \bowtie s \subseteq r$. Für alle $\nu \in s$ ist also $\mu\nu \in r$.

Damit also erstmal $\mu \in \pi[Z](r)$.

Zu zeigen: $\mu \notin (***)$.

Annahme: $\mu \in (***)$ – also gäbe es ein ν' , so dass $\mu\nu' \in (\pi[Z](r) \bowtie s) - r$ ist. Der Anteil $\pi[Z](\mu\nu')$ ist gerade μ , also müsste $\nu' \in s$ sein, damit dies erfüllt ist, und $\mu\nu' \notin r$ – im Widerspruch zu oben.

Also $\mu \in (**)$.

Aufgabe 6 (Tupeloperatoren vs. Relationale Operatoren) In der Vorlesung wurden auf einzelnen Tupeln nur die Operatoren Projektion $\pi[\bar{X}](\mu)$, Selektion $\sigma[\alpha](\mu)$ und Renaming $\rho[A \rightarrow B](\mu)$ definiert. Die relationalen Operatoren wurden dann auf Basis dieser Operatoren definiert, wobei für das Join nur eine deklarative, auf tupelbasierter Projektion aufbauende Definition gegeben wurde.

- Geben Sie die Definition des relationalen Joins an: “Sei $r \in \text{Rel}(\bar{X})$ and $s \in \text{Rel}(\bar{Y})$. Dann ist $r \bowtie s = \{\text{was gehört hier hin?}\}$.
- Überlegen Sie, wie ein Join-Operator für Tupel $\mu \in \text{Tup}(\bar{X})$, $\nu \in \text{Tup}(\bar{Y})$, also $\mu \bowtie \nu$, definiert werden kann, und geben Sie darauf basierend eine Definition des relationalen Join-Operators an.
- Kann man eine entsprechende Definition auch für die Division angeben?

- $r \bowtie s = \{\mu \in \text{Tup}(\overline{XY}) \mid \mu[\bar{X}] \in r \text{ and } \mu[\bar{Y}] \in s\}$.
- Tupelbasiertes Join: sei $\mu \in \text{Tup}(\bar{X})$, $\nu \in \text{Tup}(\bar{Y})$. Die Verknüpfung $\mu \bowtie \nu$ bezeichnet das Join zweier Tupel und ist definiert als

$$\mu \bowtie \nu = \begin{cases} \mu \cup \nu \in \text{Tup}(\overline{XY}) & \text{falls } \mu[X \cap Y] = \nu[X \cap Y] \\ \text{nichts} & \text{sonst} \end{cases}$$

Was ist dabei $\mu \cup \nu$? Dank des formalen Vorgehens ist das ganz einfach und klar: Jedes Tupel μ ist eine Abbildung, die (Spalten)namen auf Werte abbildet, Mit $\bar{X} = (X_1, \dots, X_n)$ und $\bar{Y} = (Y_1, \dots, Y_m)$ ist $\mu = \{X_1 \mapsto v_1, \dots, X_n \mapsto v_n\}$ und $\nu = \{Y_1 \mapsto w_1, \dots, Y_m \mapsto w_m\}$ und wobei $v_i \in \text{dom}(X_i)$ und $w_i \in \text{dom}(Y_i)$ ist. Also ist $\mu \cup \nu$ die Vereinigung dieser Einzelabbildungen, d.h.

$$(\mu \cup \nu)(A) = \begin{cases} \mu(A) & \text{falls } A \in \bar{X} \\ \nu(A) & \text{falls } A \in \bar{Y} \\ \text{nicht definiert} & \text{sonst} \end{cases}$$

Da gefordert wird, dass μ und ν auf $\bar{X} \cap \bar{Y}$ übereinstimmen, ist $\mu \cup \nu$ wohldefiniert.

Relationales Join damit für $r \in \text{Rel}(\bar{X})$ und $s \in \text{Rel}(\bar{Y})$: Ergebnisformat ist (natürlich wie in der originalen Definition) \overline{XY} .

Ergebnisrelation:

$$r \bowtie s = \{\mu \bowtie \nu \mid \mu \in r \text{ and } \nu \in s\} .$$

Damit hat man jetzt eine konstruktive Charakterisierung des Joins (die die Berechnung als nested-loop-join direkt nahelegt).

- Division: lässt sich nicht auf eine Tupeloperation reduzieren. Die Division ist in der grundlegenden Definition eine Mengenoperation.

Aufgabe 7 (Äquivalenzen: Join, Division, Differenz) Seien $R(X), S(Y)$ Relations-Schemata. Zeigen oder widerlegen Sie:

- (a) Sei $X \cap Y = \emptyset$.

$$(R \bowtie S) \div S \equiv R.$$

(b) Sei $X = Y$ und $Z \subseteq X$.

$$\pi[Z](R - S) \equiv \pi[Z]R - \pi[Z]S.$$

(zu a) Nach Voraussetzung sind $R(X)$ und $S(Y)$ Relationen, die kein gemeinsames Attribut haben. Damit gilt $(X \cup Y) - Y = X$.

Behauptung: dann gilt: $(R \bowtie S) \div S = R$

Definition der Division: Sei $Y \subset X$, $Z = X - Y$.

$$r \div s = \{\mu \in \text{Tup}(Z) \mid \{\mu\} \times s \subseteq r\} = \pi[Z](r) - \pi[Z](\pi[Z](r) \times s - r).$$

$$\begin{aligned} (R \bowtie S) \div S & \quad \text{keine gemeinsamen Attribute} \\ & = (R \times S) \div S \quad \text{Definition der Division} \\ & = \pi[X](R \times S) - \pi[X](\pi[X](R \times S) \times S - R \times S) \\ & = R - \pi[X](\pi[X](R \times S) - (R \times S)) \\ & = R - \pi[X](\emptyset) \\ & = R \end{aligned}$$

(zu b) $\pi[Z](R - S) \neq \pi[Z](R) - \pi[Z](S)$

Bsp:

Sei $X = Y = \{A, B\}$ und $Z \subseteq X = \{A\}$

R	
A	B
1	2

S	
A	B
1	3

$$\pi[A](R - S) = \pi[A](R) = \{(A : 1)\} \quad (\text{da } R - S = R \text{ ist}), \text{ und } \pi[A](R) - \pi[A](S) = \{(A : 1)\} - \{(A : 1)\} = \emptyset$$

Aufgabe 8 (Algebra: Minimale- und Maximale Anzahl von Tupeln) Die Relationen $R(\bar{X})$ und $S(\bar{Y})$ enthalten n bzw. m Tupel. Wie groß ist die maximale und minimale Anzahl von Tupeln, die das Ergebnis folgender Operationen (bei geeigneten \bar{X}, \bar{Y}) enthalten kann?

- a) $R \cup S$
- b) $R \bowtie S$
- c) $\sigma[C](R) \times S$, für eine Bedingung C
- d) $\pi[Y](R) - S$
- e) $R \div S$

- a) Nur zulässig wenn $\bar{X} = \bar{Y}$.
 max: $n + m$ (keine gemeinsamen Tupel)
 min: $\max\{n, m\}$ (Teilmengenbeziehung)
- b) max: $n * m$, wenn entweder $\bar{X} \cap \bar{Y} = \emptyset$ oder im natürlichen Join jedes Tupel R mit jedem Tupel aus S auf den gemeinsamen Attributen $\bar{X} \cap \bar{Y}$ zusammenpasst (d.h., diese Attribute haben in allen Tupeln beider Relationen den selben Wert).

min: 0 (natürliches Join mit $\bar{X} \cap \bar{Y} \supseteq \emptyset$, bei dem aber nichts übrigbleibt, da die Einträge der gemeinsamen Spalte(n) nicht zusammenpassen.

Hinweis: man sieht, dass ein Join also die Ergebnismenge sowohl deutlich vergrößern, als auch einschränkend wirken kann.

- Verhältnis zwischen $|R \bowtie S|$ und $|R|$ bezeichnet man als *Selektivität* des Joins (bzgl. R).
- ist $\bar{X} \cap \bar{Y}$ z.B. Schlüssel von S und Fremdschlüssel in R , so hat man $|R \bowtie S| = |R|$.

c) max: $n * m$ (alle Tupel in R erfüllen C)

min: 0 (kein Tupel in R erfüllt C)

d) Nur sinnvoll wenn $\bar{Y} \subseteq \bar{X}$.

max: n , wenn $S = \emptyset$, oder zumindest $S \cap \pi[Y](R) = \emptyset$, und bei der Projektion keine Duplikate (die wegfallen würden) entstehen.

min: 0, wenn $\pi(Y)(R) \subseteq S$.

e) max: n/m , wenn alle Werte $\pi[\bar{X} \setminus \bar{Y}](R)$ mit jedem der Werte aus S in R vorkommen.

min: 0, wenn kein Wert die durch die Division bzgl. S gestellte "für alle"-Bedingung erfüllt.

Aufgabe 9 (Transitive Hülle) Gegeben sei eine Relation $R(A,B)$. Skizzieren Sie einen Algorithmus, der, bestehend aus Operationen der relationalen Algebra und einer while-Schleife, die transitive Hülle der Relation R berechnet.

Hinweis: Die transitive Hülle einer Relation R , bezeichnet als R^* , ergibt sich wie folgt: betrachte z.B. eine Relation $R(\text{von}, \text{nach})$ von Flugverbindungen. R^2 ist dann die Menge aller Verbindungen, die über eine Zwischenlandung zustandekommen, etc; R^n sind also diejenigen, Verbindungen, die sich aus n Teilverbindungen zusammensetzen. Die unendliche Vereinigung $R \cup R^2 \cup R^3 \cup \dots$ für $R \rightarrow \infty$ wird dann als R^* bezeichnet. In einer endlichen Datenbasis benötigt man nur endlich viele Schritte um diese zu berechnen. Ein anderes beliebtes Beispiel ist die aus $Kind(x,y)$ berechnete Vorfahren-Relation.

Problem der Algebra und SQL: jedes R^n kann ausgedrückt werden - aber die beliebig große Vereinigung nicht. Man weiß nicht, wo man abbrechen soll.

Über $R(A,B)$ soll die transitive Hülle gebildet werden. T und S sind binäre Relationen über A und B .

```

S := ∅
T := R
while T - S ≠ ∅ do
  begin
    S := T;
    T := T ∪ π[T.A, R.B](σ[T.B = R.A](T × R));
  end.

```

In diesem Programm enthält S jeweils den Wert von T aus der letzten Iteration der Schleife. Die Berechnung endet, wenn S und T übereinstimmen, d.h. während der zuletzt ausgeführten Iteration wurden keine weiteren Tupel mehr zu T hinzugefügt.

```

% ?- river(N,R,L,S,A,B,C,D,E,F,G).
% ?- river(N,R,L,S,A,B,C,D,E,F,G), S \= null.
% ?- lake(N,A,B,C,D,R,E,F), R \= null.
toSea(N,S) :- river(N,_R,_L,S,_,_,_,_,_,_), S \= null.
toSea(N,S) :- river(N,R,_L,_S,_,_,_,_,_), R \= null, toSea(R,S).

```

```
toSeaLake(N,S) :- lake(N,_,_,_,_,R,_,_), R \= null, toSea(R,S).  
toSea(N,S) :- river(N,_R,L,_,_,_,_,_,_,_), L \= null, toSeaLake(L,S).
```
