Datenbanken
Wintersemester 03/04
Prof. Dr. W. May

2. Übungsblatt: Algebra

Besprechung am 21.11.

Aufgabe 1 (Division mit Basisoperationen) Beweisen Sie, daß die in der Vorlesung angebene 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

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

ist.

Betrachte

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

Nach Definition der Division ist

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

(1) $(**) \subseteq (*)$:
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 $\pi[Z](r)$ von $\mu\nu'$ ist gerade μ, also müsste $\nu' \in s$ sein, und $\mu\nu' \notin r$ – im Widerspruch zu oben.
Also $\mu \in (*)$.

(2) $(*) \subseteq (**)$:
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 $\mu = \pi[Z](diesem) = (***)$.
Womit μ nicht in $(*)$ ist. Widerspruch zur Voraussetzung.

Aufgabe 2 (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 enthalten kann?

a) $R \cup S$

b) $R \bowtie S$

c) $\sigma[C](R) \times S$, für eine Bedingung C

d) $\pi[Y](R) - S$

---

a) 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 jede Spalte aus X mit jeder Spalte aus Y zusammenpasst.
min: 0 (natürliches Join mit $\bar{X} \cap \bar{Y} \supsetneq \emptyset$, 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.

c) max: $n * m$ (alle Tupel in R erfüllen C)
min: 0 (kein Tupel in R erfüllt C)

d) max: n ($S = \emptyset$, oder zumindest $S \cup \pi[L](R) = \emptyset$)
min: 0 ($\pi(L)(R) \subseteq S$)

Aufgabe 3 (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.

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

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

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

---

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

c) nicht äquivalent: $\pi[A](S)$ eliminiert das Attribut E, was in der Ausgabe enthalten sein soll.

Aufgabe 4 (Relationale Anfragen an Mondial) Geben Sie Ausdrücke der relationalen Algebra für die folgenden Anfragen an die Mondial-Datenbank an:

a) Die Namen aller Städte, in denen mehr als 25% der Bevölkerung des jeweiligen Landes leben.

b) Die Namen aller Organisationen, die auf jedem Kontinent mindestens ein Mitgliedsland haben.

c) Alle Landescodes von Ländern, in denen eine Sprache gesprochen wird, die auch in der Schweiz gesprochen wird.

d) Alle Landescodes von Ländern, in denen ausschliesslich Sprachen gesprochen werden, die in der Schweiz nicht gesprochen werden.

e) Alle Landescodes von Ländern, in denen nur Sprachen gesprochen werden, die auch in der Schweiz gesprochen werden.

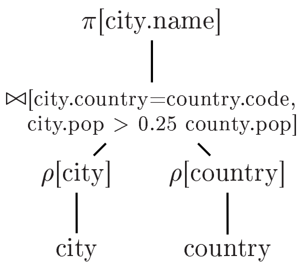
f) Alle Landescodes von Ländern, in denen alle Sprachen gesprochen werden, die in der Schweiz gesprochen werden.

Für spätere Übungsblätter:

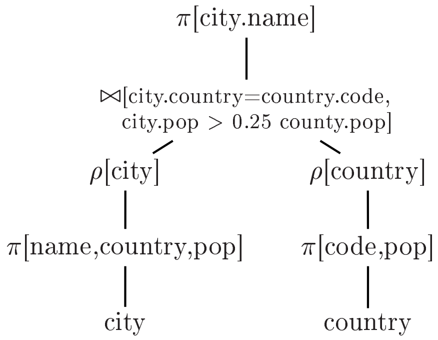
- Geben Sie dieselben Anfragen in SQL an.
- Geben Sie dieselben Anfragen im relationalen Kalkül an.

Sei allgemein $\rho[alias]$ die Umbenennung, die alle Attribute mit dem Präfix X qualifiziert.

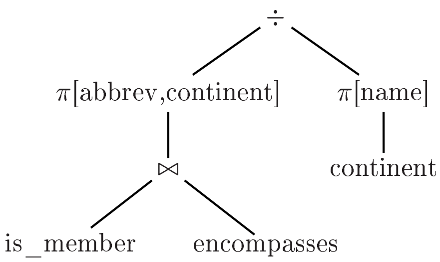
a) Join mit Zusatzbedingung:



oder

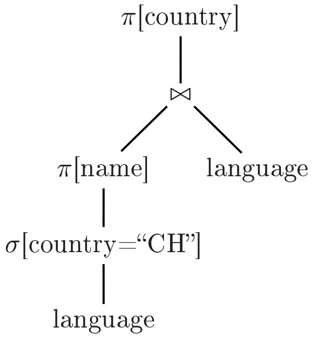


b) Division:



is_member(org,country,type) und encompasses(country,contintent,percent) werden über country gejoint. Davon wird nur (organisation,continent) (*) benötigt. Das wird dann durch die Menge der Namen der Kontinente geteilt, womit alle Organisationen übrig bleiben, die in (*) mit jedem Kontinent genannt werden.

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

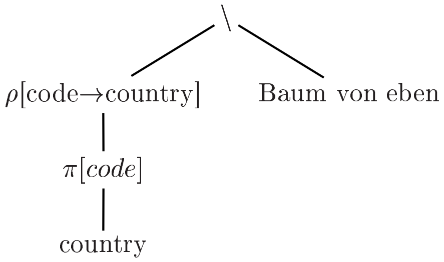


Der linke Ast ergibt die Tabelle

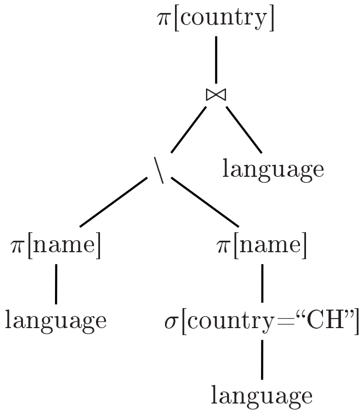
language
name
german
french
italian
romansch

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

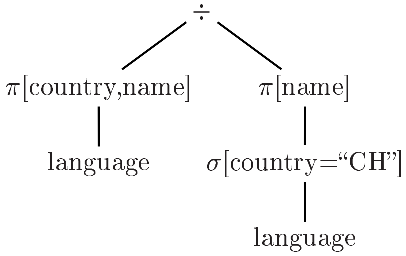
d) Genau alle Länder, die eben nicht aufgezählt wurden:



e) 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 betreffenden Länder gibt man aus.



f) "alle" – also mal wieder Division. $\pi[country, name](language)$ durch die Menge der schweizer Sprachen dividieren. Übrig bleiben die Länder die mit allen diesen Sprachen genannt werden:



Aufgabe 5 (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 $R(von, nach)$ 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 $R \cup R^2 \cup R^3 \cup ...$ für $R \to \infty$ 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.

$S := \emptyset$
$T := R$
while $T - S \neq \emptyset$ do
  begin
    $S := T$;
    $T := T \cup \pi[T.A, R.B](\sigma[T.B = R.A](T \times R))$;
  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.
