

## Aufgabe 1

Aus Platzgründen werden die Tabellen (e=employee, w=works, c=company, m=manages) und Attribute (pn=person-name, s=street, c=city, cn=company-name, ...) abgekürzt.

1.  $\pi_{e.pn}(\sigma_{m.s=e.s, m.c=e.c, m.pn=e.pn}(e \times (\pi_{e.s, e.c, m.pn, m.mn}(\sigma_{e.pn=m.mn}(e \times m))))))$
2.  $\pi_{w.pn}(w - \sigma_{w.cn="FBC''}(w))$
3.  $\pi_{R.name}(\sigma_{R.salary < S.salary}(\rho_R(e \bowtie w) \times \rho_S(\sigma_{cn="FBC''}(e \bowtie w))))$
4.  $\pi_{w.pn}(w - \sigma_{w.cn="FBC''}(w)) - \pi_{R.name}(\sigma_{R.salary \leq S.salary}(\rho_R(e \bowtie w) \times \rho_S(\sigma_{cn="FBC''}(e \bowtie w))))$
5.  $\pi_{cn}(\vartheta_{\max(pn)}(cn \vartheta_{\text{count}(pn)}(w)))$
6.  $\pi_{cn}(\vartheta_{\min(s)}(cn \vartheta_{\text{sum}(s)}(w)))$
7.  $\pi_{cn}(\sigma_{R.\text{avg}(s) > S.\text{avg}(s)}(\rho_R(cn \vartheta_{\text{avg}(s)}(w)) \times \rho_S(\sigma_{cn="FBC''}(cn \vartheta_{\text{avg}(s)}(w)))))$
8.  $\pi_{R.cn}(cn \vartheta_{\text{count}(s)}(\sigma_{R.cn=S.cn, R.c=S.c}(\vartheta_R(c) \times \vartheta_S(c))))$

## Aufgabe 2

Listing 1: Aufgabe 2

```

1 // Gibt das Tupel an Stelle i zurück
2 list get(list l, var i)
3 while i > 0
4     l <= tail(l)
5     i <= i - 1
6 return head(l)
7
8 // Gibt zurück, ob Liste leer ist
9 bool empty(list l)
10 if head(l) = NULL
11     return true
12 return false
13
14 // Gibt Grösse der Liste zurück
15 var size(list l)
16 int s <= 0
17 while empty(l) = false
18     s <= s + 1
19     l <= tail(l)
20 return s
21
22 // Fügt zwei listen zusammen
23 list concatenate(tuple x, tuple y)
24 for var a <= size(x) to a >= 0, a <= a - 1
25     cons(get(x, a), y)
26 return y
27
28 // Ergibt das kartesische Produkt r x s
29 list cartesianProduct(list r, list s)

```

```
30 list t
31 for var a <= size(r) - 1 to a >= 0, a <= a - 1)
32     for var b <= size(s) to b >= 0, b <= b - 1
33         cons(concatenate(get(r, a), get(s, b)), t)
34 return t
```

## Aufgabe 3

1. Für ein beliebiges Tupel  $a_n$  mit Attributen  $A, B, C$  gilt: Sollte es im Fall  $x$  zuerst durch eine Projektion auf die Spalten A und B reduziert werden, wird danach jedes Element, für das  $A > 10$  gilt, zurückgeliefert. Bei  $y$  werden im ersten Schritt exakt die selben Tupel ausgewählt, nämlich nur die, für welche auch  $A > 10$  gilt, und davon werden dann die zwei Attribute A und B zurückgegeben. Die Beiden Operationen Projektion und Auswahl (**select**) können in diesem Fall ausgetauscht werden, die beiden Ergebnisse sind äquivalent.
2. Hier sind  $x$  und  $y$  nicht gleichbedeutend. Wenn ein Tupel  $a_1$  einen Wert in  $r.A$  mit einem anderen Tupel  $a_2$  in  $t.A$  gemeinsam hat, aber  $r.B$  (in  $a_1$ ) nicht mit  $t.B$  (in  $a_2$ ) übereinstimmt, wird  $a_1$  in der Auswertung  $\pi_A(r - t)$  erscheinen, nicht jedoch in  $\pi_A(r) - \pi_A(t)$ . Die Differenz wird hier nicht aus den selben Mengen gebildet.