

### Aufgabe 1 (Abfragen mit Relationaler Algebra)

Drücken Sie die gegebenen Abfragen in der relationalen Algebra aus.

employee (person\_name, street, city)

works (person\_name, company\_name, salary)

company (company\_name, city)

manages (person\_name, manager\_name)

1. Finde die Namen und Wohnorte aller Angestellten, welche für FBC arbeiten.
2. Finde die Namen und Wohnorte mit Strasse aller Angestellten, welche für FBC arbeiten und die mehr als CHF 100'000.- verdienen.
3. Finde die Namen aller Angestellten, die in der Stadt arbeiten in der sie auch wohnen.
4. Finde die Namen aller Angestellten, die in derselben Stadt an derselben Strasse wohnen, wie ihr Manager.
5. Finde die Firma mit den meisten Angestellten.
6. Finde die Firma, welche insgesamt die kleinste Lohnsumme bezahlt.
7. Finde diejenigen Firmen, deren Angestellte im Durchschnitt mehr verdienen, als der Durchschnittslohn der FBC.
8. Finde alle Firmen, die in jeder Stadt sind, in der auch die FBC ist.

### Aufgabe 2 (NULL-Werte)

Gegeben sind folgende Relationen  $r(A, B, C, D)$  und  $s(A, E)$ :

A	B	C	D
"A"	1000	3	" "
"A"	700	Null	"agh"
"A"	Null	0	"abcdf"
"A"	1000	4	Null
"B"	Null	Null	"bdf"
"B"	1500	Null	"c"
Null	1000	8	" "
Null	700	12	Null

A	E
"B"	1
"C"	2
"C"	3

1. Evaluieren Sie für jede Zeile von  $r$  den Wert des Prädikats  $p$  mit

$$p = (B \cdot C < 5000 \text{ or } D \text{ is null})$$

Berechnen Sie:

2.  $\sigma_p(r)$
3.  $AG_{avg(B), sum(C)}(r)$
4.  $AG_{avg(B)}(\pi_{A,B}(r))$
5.  $r \bowtie s$

6.  $r \bowtie s$

7.  $r \Join s$

### Aufgabe 3 (Implementation Kartesisches Produkt und Natural Join)

Gegeben seien Relationen  $r(A, B)$  und  $s(B, C)$ . Die Relationen liegen in Form von Listen von Tupeln vor. Tupel sind Listen mit fester Länge, in unserem Falle mit Länge zwei. Zusätzlich zu den üblichen Funktionen `head` und `tail` gibt es auf Listen (und damit auch auf Tupeln) die Funktion `list cons(element a, list l)` die die Liste liefert, die durch Voranstellen von Element `a` an die Liste `l` entsteht. Die leere Liste heisst `nil`. Andere Funktionen auf Listen gibt es nicht.

1. Geben Sie einen Algorithmus an, um das kartesische Produkt der beiden Relationen zu berechnen.
2. Geben Sie einen Algorithmus an, um den natural join der beiden Relationen zu berechnen.