

## Aufgabe 1

- Zur Vereinfachung führen wir für die vorhandenen Attribute Aliase ein:  $A=TelefonNr$ ,  $B=Talort$ ,  $C=Skigebiet$ ,  $D=Lift$ ,  $E=Kapazität$ .  
Folgende candidate keys ergeben sich aus den eingetragenen Werten: [AD], [AE], [BD], [BE], [ABE], [ABD], [ACD], [ACE], [ADE], [BCD], [BCE], [BDE], [ABCD], [ABCE], [ABDE], [ACDE], [BCDE], [ABCDE].
- Am besten wird ein Primärschlüssel gewählt, bei welchem auch zukünftige Tupel keine redundanten Werte aufweisen. Dies wäre z.B. bei [BD] der Fall, also *Talort* und *Lift*, denn es ist anzunehmen, dass es keinen gleich benannten Lift in einem Ort (*Talort*) gibt.

## Aufgabe 2

- $\pi_{a1}(r)$

a1
a
b
c

- $\sigma_{a1="b"}(r)$

a1	a2
b	d
b	e

- $r \times s$

a1	r.a2	s.a2	a3
a	d	d	g
a	d	e	h
b	d	d	g
b	d	e	h
b	e	d	g
b	e	e	h
c	f	d	g
c	f	e	h

- $\sigma_{r.a2=s.a2}(r \times s)$

a1	r.a2	s.a2	a3
a	d	d	g
b	d	d	g
b	e	e	h

- $\pi_{a1}(r) - \pi_{a1}(\sigma_{a2="d"}(r))$

a1
c

## Aufgabe 3

- 1  $\pi_{person-name}(\sigma_{company-name="FBC"}(works))$
- 2  $\pi_{employee.person-name, employee.city}(\sigma_{works.company-name="FBC"}(employee \bowtie works))$
- 3  $\pi_{employee.person-name, employee.street, employee.city}(\sigma_{works.company-name="FBC", works.salary > 100000}(employee \bowtie works))$
- 4  $\pi_{employee.person-name}(\sigma_{employee.city=company.city}((employee \bowtie works) \bowtie company))$

## Aufgabe 4

- 1 Eine Menge kann im Unterschied zur Liste keine gleichen Elemente enthalten, denn jedes Element muss "wohlunterschieden" sein. Eine Liste hat von ihrem Aufbau her immer auch schon eine implizierte Reihenfolge ihrer Elemente. Zudem kann eine Menge endlich oder unendlich viele Elemente enthalten, wohingegen eine Liste nur eine endliche Anzahl von Elementen aufnehmen kann.

Listing 1: Aufgabe 4.2

```
1 boolean equals(list l1, list l2)
2
3 int m, n
4 for m=0 to l1.length
5     for n=0 to l2.length
6         if l1[m] == l2[n]
7             break
8         else if n == l2.length-1
9             return false
10 return true
```