

Datenbanken Übungsblatt 4

Aufgabe 1 (Abfragen mit Relationaler Algebra)

Drücken Sie die gegebenen Abfragen in der relationalen Algebra aus.

```
employee (personname, street, city)
works (personname, companyname, salary)
company (companyname, city)
manages (personname, managername)
```

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.

solution

1.

```
select * from employee where works.companyname = "FBC" left join works on
employee.personname = works.personname
```

2.

```
select * from employee where works.companyname = "FBC" AND where
works.salary < 100000 left join works on employee.personname =
works.personname
```

3.

```
select * from employee where works.salary < 100000 left
join company on employee.city = company.city
```

4.

```
select * from employee as manager
join manages on employee.personname = managers.personname
```

5.

```
SELECT MAX(counts) FROM (SELECT COUNT(companyname) as counts
FROM works
GROUP BY companyname);
```

6.

```
SELECT MIN(sum) FROM (
    SELECT COUNT(salary) as lohn
    FROM works
    GROUP BY salary
);
```

7.

```
SELECT AVG(salary) FROM works where companyname = "FBC"
```

8.

```
(SELECT * FROM company company1, company company2 where
company1.companyname = "FBC" and company1.city = company2.city and
company1.companyname IS NOT company2.companyname)
```

operatoren

additional (helper) operator	equivalent basic relational algebra operation	description
$table_a \cap table_b$	$table_a - (table_a - table_b)$	intersection / schnittmenge

additional (helper) operator	equivalent basic relational algebra operation	description
table_a \bowtie table_b	nothing	Natural join (\bowtie) / takes only rows where column names and values are the same and merges them into one row on the result
table_a \rtimes table_b	-	left outer join
table_a \leftarrow table_b	-	right outer join
table_a \ltimes table_b	-	full outer join
nID, name, dept_name, salary/12 (table_human)	asdf	Generalized Projection / like normal projection but functions with arithmetic operations and other column names can be used as the selected column names, see "salary/12"
sum(column_name) (table_name)	-	Aggregation / a predefined function, which takes multiple records as input and returns single result (avg, min, max, sum, count)

Aufgabe 2 (NULL-Werte)

unknown operations

what is 'unknown'

if 'column_salary' value for this record is null, the result of the operation 'unknown' "column_salary > 50000" -> "null > 50000" => unknown

operation	result
unknown or true	true
unknown or false	unknown
unknown or unknown	unknown
true and unknown	unknown
false and unknown	false
unknown and unknown	unknown
not unknown	unknown
not NULL	NULL
NULL or false	NULL

operation	result
NULL or true	true
NULL or NULL	NULL
NULL and false	false
NULL and true	NULL
NULL and NULL	NULL

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

truth table / wahrheitstabelle

table/schema $r(A,B,C,D)$

A	B	C	D	$B \cdot C < 5000$	D is null	$p = (B \cdot C < 5000 \text{ or } D \text{ is null})$
"A"	1000	3	""	true	false	true
"A"	700	Null	"agh"	unknown	false	unknown
"A"	Null	0	"abcdf"	unknown	false	unknown
"A"	1000	4	Null	true	true	true
"B"	Null	Null	"bdf"	unknown	false	unknown
"B"	1500	Null	"c"	unknown	false	unknown
Null	1000	8	""	false	false	false
Null	700	12	Null	false	true	true

table/schema $s(A,E)$:

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

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

solutions

1. see $\text{###table/schema } r(A,B,C,D)$
2. $\sigma_p(r) \Rightarrow \text{select } * \text{ from } r \text{ where } p \Rightarrow \text{select } * \text{ from } r \text{ where } p = \text{true}$ result: $|A|B|C|D|B \cdot C < 5000|D \text{ is null}|p = (B \cdot C < 5000 \text{ or } D \text{ is null})|$
 $|"A"|1000|3|""|true|false|true|$
 $|"A"|1000|4|Null|true|true|true|$
 $|Null|700|12|Null|false|true|true|$

3. AGavg(B),sum(C)(r) => select avg(B), avg(c) from r result: |"A"|1000|3|""|true | false| true |
 |"A"|700|Null|"agh"|unknown| false| unknown | |"A"|Null|0|"abcdf"|unknown | false| unknown|
 |"A"|1000|4|Null|true | true | true | |"B"|Null|Null|"bdf"|unknown | false| unknown |
 |"B"|1500|Null|"c"|unknown | false| unknown | |Null|1000|8|""|false | false | false |
 |Null|700|12|Null|false | true | true |

average of column 'B' = (1000 + 700 + 1000 + 1500 + 1000 + 700) / 6 = 983,333333333 sum of column 'C' =
 (3+0+4+8+12) = 27

result:

B	C
983,333333333	27

4. AGavg(B)($\pi_{A,B}(r)$)

- $r_1 = \pi_{A,B}(r) \Rightarrow \text{select } A, B \text{ from } r \mid A \mid B \mid \text{---} \mid \mid "A" \mid 1000 \mid \mid "A" \mid 700 \mid \mid "A" \mid \text{Null} \mid \mid "A" \mid 1000 \mid \mid "B" \mid \text{Null} \mid$
 $\mid "B" \mid 1500 \mid \mid \text{Null} \mid 1000 \mid \mid \text{Null} \mid 700 \mid$
- AGavg(B)(r_1) average of column 'B' = (1000 + 700 + 1000 + 1500 + 1000 + 700) / 6 = 983,333333333
 result: |B| |---| |983,333333333|

5. $r \bowtie s$

6. $r \bowtie_{\text{c}} s$

7. $r \bowtie_{\text{c}} s$

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.

2