

13/2/2014 Úvod do databáz, **skúškový** test, max 25 bodov, 90 min

1. Daná je databáza (bez duplikátov a null hodnôt): lubi(Pijan, Alkohol), capuje(Krcma, Alkohol), navstivil(Idn, Pijan, Krcma), vypil(Idn, Alkohol, Mnozstvo).

Platí: Idn \rightarrow Pijan, Krcma; Idn, Alkohol \rightarrow Mnozstvo; Mnozstvo $>$ 0.

a) Sformulujte nasledujúci dotaz v relačnom kalkule, (2), Datalogu (2), SQL (2) a relačnej algebre (2). Nájdite dvojice [P, A] také, že pijan P ľúbi alkohol A; a v každej krčme, ktorá čapuje alkohol A, vypil P počas niektorej návštevy viacej alkoholu A než ktorýkoľvek iný pijan počas jednej návštevy. (Inak povedané: P ľúbi A, a je rekordérom v pití A na jedno posedenie v každej krčme, ktorá A čapuje.)

Relačný kalkul:

{[P, A]:

```
lubí(P, A)  $\wedge$   $\neg$  (
  /* niekde_nezvitazil(P, A) */
   $\exists$ K
  capuje(K, A)  $\wedge$   $\neg$  (
    /* rekord(P, A, K) */
     $\exists$ I  $\exists$ M
    navstivil(I, P, K)  $\wedge$  vypil(I, A, M)  $\wedge$   $\neg$  (
      /* iny_vypil_tolko(P, A, K, M) */
       $\exists$ I2  $\exists$ P2  $\exists$ M2
      navstivil(I2, P2, K)  $\wedge$  vypil(I2, A, M2)  $\wedge$  P2  $\neq$  P  $\wedge$  M2  $\geq$  M
    )
  )
)
```

Datalog:

```
answer(P, A)  $\leftarrow$ 
  lubí(P, A),
  not niekde_nezvitazil(P, A).
```

```
niekde_nezvitazil(P, A)  $\leftarrow$ 
  lubí(P, A), /* safety */
  capuje(K, A),
  not rekord(P, A, K).
```

```
rekord(P, A, K)  $\leftarrow$ 
  navstivil(I, P, K),
  vypil(I, A, M),
  not iny_vypil_tolko(P, A, K, M).
```

```
iny_vypil_tolko(P, A, K, M)  $\leftarrow$ 
  navstivil(I, P, K), /* safety */
  vypil(I, A, M), /* safety */
  navstivil(I2, P2, K),
  vypil(I2, A, M2),
  not P2 = P,
  M2  $\geq$  M.
```

```
SQL:
create temporary table iny_vypil_tolko as
select n1.Pijan, v1.Alkohol, n1.Krcma, v1.Mnozstvo
from navstivil n1, vypil v1, navstivil n2, vypil v2
where n1.Idn = v1.Idn and n2.Idn = v2.Idn and v1.Alkohol = v2.Alkohol and n1.Krcma = n2.Krcma and
n1.Pijan <> n2.Pijan and v2.Mnozstvo >= v1.Mnozstvo
```

```
create temporary table rekord as
select n.Pijan, v.Alkohol, n.Krcma
from navstivil n, vypil v
where n.Idn = v.Idn and not exists (
select *
from iny_vypil_tolko t
where n.Pijan = t.Pijan and v.Alkohol = t.Alkohol and n.Krcma = t.Krcma and v.Mnozstvo = t.Mnozstvo)
```

```
create temporary table niekde_nezvitazil as
select
from lubi l, capuje c
where l.Alkohol = c.Alkohol and not exists (
select *
from rekord r
where l.Pijan = r.Pijan and l.Alkohol = r.Alkohol and c.Krcma = r.Krcma)
```

```
/* answer */
select l.Pijan, l.Alkohol
from lubi l
where not exists (
select *
from niekde_nezvitazil nn
where l.Pijan = nn.Pijan and l.Alkohol = nn.Alkohol)
```

Relačná algebra:

$$\text{iny_vypil_tolko} := \pi_{n1.P, v1.A, n1.K, v1.M} ($$

$$(P_{n1(I, P, K)}(\text{navstivil}) \bowtie P_{v1(I, A, M)}(\text{vypil})) \bowtie_{v1.A=v2.A \wedge n1.K = n2.K \wedge n1.P \neq n2.P \wedge v2.M \geq v1.M}$$

$$(P_{n2(I, P, K)}(\text{navstivil}) \bowtie P_{v2(I, A, M)}(\text{vypil})))$$

$$\text{rekord} := \pi_{n.P, v.A, n.K} ($$

$$\pi_{n.P, v.A, n.K, v.M} (P_{n(I, P, K)}(\text{navstivil}) \bowtie P_{v(I, A, M)}(\text{vypil})) - \text{iny_vypil_tolko})$$

$$\text{niekde_nezvitazil} := \pi_{l.P, l.A} (\pi_{l.P, l.A, c.K} (P_{l(P, A)}(\text{lubi}) \bowtie P_{c(K, A)}(\text{capuje})) - \text{rekord})$$

```
/* answer */
P_{l(P, A)}(\text{lubi}) - \text{niekde\_nezvitazil}
```

b) Sformulujte nasledujúci dotaz v relačnom kalkule (2) a SQL (2). Nájdite dvojice [P1, P2] také, že rozdiel v počtoch krčiem, ktoré navštívili pijani P1 a P2, je aspoň 7. (Pijan, ktorý ľúbi nejaký alkohol, nemusel navštíviť žiadnu krčmu.)

Relačný kalkul:

{[P1, P2]:

∃C1 ∃C2

(

♥ C1=count(K) (∃I navstivil(I, P1, K))

∨ (C1=0 ∧ ∃A lubi(P1, A) ∧ ¬ (∃I ∃K navstivil(I, P1, K)))

)

∧

(

♥ C2=count(K) (∃I navstivil(I, P2, K))

∨ (C2=0 ∧ ∃A lubi(P2, A) ∧ ¬ (∃I ∃K navstivil(I, P2, K)))

)

∧

C1 - C2 >= 7 /* ak chceme aj opačné dvojice pijanov, tak |C1 - C2| >= 7 */

}

SQL:

```
create temporary table pocet_krciem as
select n.Pijan, count(distinct n.Krcma) as C
from navstivil n
group by n.Pijan
union
select l.Pijan, 0 as C
from lubi l
where not exists (
  select *
  from navstivil n
  where n.Pijan = l.Pijan)
```

/* answer */

```
select pk1.Pijan as P1, pk2.Pijan as P2
from pocet_krciem pk1, pocet_krciem pk2
where pk1.C - pk2.C >= 7
```

/* ak chceme aj opačné dvojice pijanov, tak |pk1.C - pk2.C >= 7| */

2. Uvažujte SQL dotaz nad reláciou $r(X, Y)$ bez duplikátov a null hodnôt:

```
select r1.X, sum(r1.Y) from r r1 where not exists (  
    select * from r r2 where r1.Y = r2.X) group by r1.X having sum(r1.Y) > 3
```

a) Sformulujte daný dotaz ekvivalentne v Datalogu. (2)

? answer(X, S).

```
answer(X, S) ←  
    subtotal(from_where(X, Y), [X], [S = sum(Y)]),  
    S > 3.
```

```
from_where(X, Y) ←  
    r(X, Y),  
    not r(Y, _).
```

b) Vypočítajte výsledok daného dotazu pre reláciu

$r(X, Y) = \{[0, 1], [0, 2], [0, 3], [2, 0], [2, 5], [3, 1], [3, 6], [5, 4]\}$. (2)

```
from_where = {[0, 1], [3, 1], [3, 6], [5, 4]}
```

/* Vyberieme z r také záznamy, pre ktoré sa hodnota Y nenadobúda nikde v r na atribúte X (t.j. nenachádza sa v r na prvej pozícii). Takými hodnotami Y sú 1, 6, 4. */

```
answer = {[3, 7], [5, 4]}
```

/* Zgrupujeme reláciu from_where podľa X. Sú 3 grupy: 0, 3, 5. V každej grupe zosumujeme hodnoty Y.

Dostaneme $(X, S) = \{[0, 1], [3, 7], [5, 4]\}$. Po aplikácii selekcie $S > 3$ sa táto relácia zredukuje na $\{[3, 7], [5, 4]\}$. */

Výsledkom dotazu je relácia $\{[3, 7], [5, 4]\}$.

3. Uved'te príklad rozvrhu dvoch commitovaných transakcií, ktorý je:

a) konflikt-sériovateľný a zároveň nie obnoviteľný; **(1)**

$w_1(X), r_2(X), c_2, c_1$

b) konflikt-sériovateľný, obnoviteľný, ale nie vyhýbajúci sa kaskádovým abortom; **(1)**

$w_1(X), r_2(X), c_1, c_2$

c) striktný a zároveň nie konflikt-sériovateľný; **(1)**

$r_1(X), w_2(X), c_2, w_1(X), c_1$

d) konflikt-sériovateľný a zároveň striktný a zároveň taký, že sa nedá sa generovať schedulerom používajúcim striktný dvojfázový zamykací protokol. **(1)**

$r_1(X), w_2(X), c_1, c_2$

4. $B(r)$ označuje počet blokov relácie $r(X, Y)$, $B(s)$ označuje počet blokov relácie $s(W, Z)$, všetky bloky sú maximálne naplnené. V reláciách nie sú null hodnoty ani duplikované záznamy. Bloky na disku a v operačnej pamäti sú rovnako veľké.

M označuje počet voľných blokov v operačnej pamäti vyhradených pre výpočet dotazu

$\pi_{X, Y}(r \bowtie_{Y=W} s)$, pričom $B(r)=2*M$, $B(s)=M$.

a) Popíšte organizáciu operačnej pamäte pri výpočte daného dotazu algoritmom nested-loop join. (1)

1 blok sa rezervuje pre zápis výstupu

1 blok sa rezervuje pre čítanie relácie r (lebo r je väčšia relácia než s)

$M - 2$ blokov sa rezervuje pre čítanie relácie s

b) Vyjadrite počet vstupných diskových prenosov (z disku do operačnej pamäte), ktoré nested-loop join urobí na danom dotaze, ako funkciu M . Vysvetlite. (2)

Najskôr sa prečíta prvých $M-2$ blokov relácie s . Potom sa postupne prečíta celá relácia r , t.j. $2*M$ blokov (záznamy v blokoch pre s a r sa priebežne joinujú).

Potom sa prečítajú zvyšné 2 bloky relácie s a znovu sa postupne prečíta celá relácia r , t.j. $2*M$ blokov (záznamy v blokoch pre s a r sa priebežne joinujú).

Dokopy sa urobí $M-2 + 2*M + 2 + 2*M$ blokových prenosov z disku do RAM, teda **$5*M$ vstupných diskových prenosov**.

Jeden diskový prenos sa dá ušetriť. Pred druhou iteráciou je totiž posledný blok relácie r už v operačnej pamäti, takže ho netreba čítať znovu.

c) Predpokladajte, že W je kľúčom v relácii s . Určte minimálnu možnú veľkosť a maximálnu možnú veľkosť výsledku daného dotazu (v počte blokov). Vysvetlite. (2)

Minimálna veľkosť výsledku je 0 blokov. Tak to dopadne, keď množiny hodnôt $\{r.Y\}$ a $\{s.W\}$ majú prázdny prienik.

Maximálna veľkosť výsledku je $B(r)$ blokov (t.j. $2*M$ blokov). Keď sú množiny hodnôt $\{r.Y\}$ a $\{s.W\}$ rovnaké, tak je výsledkom celá relácia r . Keďže W je kľúčom v relácii s , pre každú hodnotu $s.W$ v relácii s existuje len jeden záznam s s tou hodnotou. Preto nezáleží na tom, či vo výsledku uvažujeme aj duplikované záznamy alebo nie—duplikáty tam nie sú.