

9/2/2016 Úvod do databáz, skúškový test, max 25 bodov, 90 min

1. Uvažujte databázu bez duplikátov a null hodnôt: $\text{capuje}(\text{Krcma}, \text{Alkohol})$, $\text{lubi}(\text{Pijan}, \text{Alkohol})$, $\text{navstivil}(\text{Idn}, \text{Pijan}, \text{Krcma})$, $\text{vypil}(\text{Idn}, \text{Alkohol}, \text{Mnozstvo})$.

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

a) Nájdite alkoholy, ktoré sa čapujú v krčme Riviera, ľúbi ich každý pijan, ale pijú ich len takí pijani, ktorí krčmu Riviera nikdy nenavštívili (pijan je niekto, kto ľúbi nejaký alkohol alebo navštívil nejakú krčmu).

Sformulujte dotaz v Datalogu (2) a relačnom kalkule (2).

Datalog:

```
answer(A) ←  
  capuje(riviera, A),  
  not niekto_nelubi(A),  
  not navstivilr_vypil(A).
```

```
niekto_nelubi(A) ←  
  capuje(_, A),  
  pijan(P),  
  not lubi(P, A).
```

```
navstivilr_vypil(A) ←  
  navstivil(_, P, riviera),  
  navstivil(I, P, _),  
  vypil(I, A, _).
```

```
pijan(P) ←  
  navstivil(_, P, _).
```

```
pijan(P) ←  
  lubi(P, _).
```

Relačný kalkul:

```
{A:  
  capuje(riviera, A) ∧  
  ¬  
  (  
    ∃P ∃I ∃K ∃A2  
    (navstivil(I, P, K) ∨ lubi(P, A2)) ∧  
    ¬ lubi(P, A)  
  )  
  ∧  
  ¬  
  (  
    ∃P ∃I1 ∃I2 ∃K2 ∃M  
    navstivil(I1, P, riviera) ∧ navstivil(I2, P, K2) ∧ vypil(I2, A, M)  
  )  
}
```

b) Nájďte dvojice [K, P] také, že pijan P navštívil krčmu K vícekrát než kterýkoliv jiný pijan a zároveň vypil v K méně borovičky než kterýkoliv jiný návštěvník K. Sformulujte dotaz v Datalogu (2) a SQL (2).

Datalog:

answer(K, P) ←

navstivil(_, P, K),
not iny_navstivil_viacr(P, K),
not iny_vypil_menejr(P, K).

iny_navstivil_viacr(P, K) ←

pocet_navstev(P, K, N1),
pocet_navstev(P2, K, N2),
not P = P2,
N2 >= N1.

pocet_navstev(P, K, N) ←

subtotal(navstivil(I, P, K), [P, K], [N = count(I)]).

iny_vypil_menejr(P, K) ←

mnozstvob(P, K, M1),
mnozstvob(P2, K, M2),
not P = P2,
M2 <= M1.

mnozstvob(P, K, M) ←

subtotal(b(_, P, K, X), [P, K], [M = sum(X)]).

mnozstvob(P, K, 0) ←

navstivil(_, P, K),
not b2(P, K).

b(I, P, K, X) ←

navstivil(I, P, K),
vypil(I, borovicka, X).

b2(P, K) ←

b(_, P, K, _).

```

SQL:
with pocetn as
(
select n.Pijan, n.Krcma, count(n.Idn) as N
from navstivil n
group by n.Pijan, n.Krcma
),

with mnozstvob as
(
(
select n.Pijan, n.Krcma, sum(v.Mnozstvo) as M
from navstivil n, vypil v
where n.Idn = v.Idn and v.Alkohol = 'borovicka'
group by n.Pijan, n.Krcma
)
union
(
select n.Pijan, n.Krcma, 0 as M
from navstivil n
where not exists (
select *
from navstivil n2, vypil v
where n.Pijan = n2.Pijan and n.Krcma = n2.Krcma and n2.Idn = v.Idn and v.Alkohol = 'borovicka'
)
)
),

select pn.Krcma as K, pn.Pijan as P
from pocetn pn, mnozstvob mb
where pn.Krcma = mb.Krcma and pn.Pijan = mb.Pijan and not exists (
select *
from pocetn pn2
where pn2.Krcma = pn.Krcma and pn2.Pijan <> pn.Pijan and pn2.N >= pn.N
) and not exists (
select *
from mnozstvob mb2
where mb2.Krcma = mb.Krcma and mb2.Pijan <> mb.Pijan and mb2.M <= mb.M
)

```

2. Uvažujte SQL dotaz nad reláciou $r(X, Y)$:

*select r1.Y as X, r2.Y as Y, sum(r2.X) as S from r r1, r r2 where r1.X < r2.Y
group by r1.Y, r2.Y having sum(r1.X) > r1.Y*

a) Vyjadrite daný dotaz v Datalogu. (2)

$\text{answer}(Y1, Y2, SX2) \leftarrow$
 $\text{subtotal}(\text{join}(X1, Y1, X2, Y2), [Y1, Y2], [SX1 = \text{sum}(X1), SY1 = \text{sum}(Y1), SX2 = \text{sum}(X2)]),$
 $SX1 > Y1.$

$\text{join}(X1, Y1, X2, Y2) \leftarrow$
 $r1(X1, Y1),$
 $r2(X2, Y2),$
 $X1 < Y2.$

b) Vyjadrite daný dotaz v relačnej algebre. (2)

$\pi_{X=r1.Y, Y=r2.Y, S} (\sigma_{SX1 > r1.Y} (\Gamma_{r1.Y, r2.Y, SX1 = \text{sum}(r1.X), SY1 = \text{sum}(r1.Y), S = \text{sum}(r2.X)} (P_{r1}(r) \bowtie_{r1.X < r2.Y} P_{r2}(r))))$

c) Uved'te medzivýsledky po vykonaní jednotlivých operátorov pri výpočte daného dotazu podľa plánu z úlohy

b) pre reláciu

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

$\text{join}(r1.X, r1.Y, r2.X, r2.Y) =$

$P_{r1}(r) \bowtie_{r1.X < r2.Y} P_{r2}(r) =$

{
 $[1, 1, 1, 4], [1, 1, 2, 3], [1, 1, 2, 4], [1, 1, 4, 2], [1, 1, 4, 3], [1, 1, 4, 4],$
 $[1, 4, 1, 4], [1, 4, 2, 3], [1, 4, 2, 4], [1, 4, 4, 2], [1, 4, 4, 3], [1, 4, 4, 4],$
 $[2, 1, 1, 4], [2, 1, 2, 3], [2, 1, 2, 4], [2, 1, 4, 3], [2, 1, 4, 4],$
 $[2, 3, 1, 4], [2, 3, 2, 3], [2, 3, 2, 4], [2, 3, 4, 3], [2, 3, 4, 4],$
 $[2, 4, 1, 4], [2, 4, 2, 3], [2, 4, 2, 4], [2, 4, 4, 3], [2, 4, 4, 4],$
 $[3, 1, 1, 4], [3, 1, 2, 4], [3, 1, 4, 4]$
 }

$\text{agg}(r1.Y, r2.Y, SX1, SY1, S) =$

$\Gamma_{r1.Y, r2.Y, SX1 = \text{sum}(r1.X), SY1 = \text{sum}(r1.Y), S = \text{sum}(r2.X)} (P_{r1}(r) \bowtie_{r1.X < r2.Y} P_{r2}(r)) =$

{
 $[1, 2, 1, 1, 4], [1, 3, 6, 4, 12], [1, 4, 18, 9, 21], [3, 3, 4, 6, 6], [3, 4, 6, 9, 7],$
 $[4, 2, 1, 4, 4], [4, 3, 6, 16, 12], [4, 4, 9, 24, 14]$
 }

$\text{sel}(r1.Y, r2.Y, SX1, SY1, S) =$

$\sigma_{SX1 > r1.Y} (\Gamma_{r1.Y, r2.Y, SX1 = \text{sum}(r1.X), SY1 = \text{sum}(r1.Y), S = \text{sum}(r2.X)} (P_{r1}(r) \bowtie_{r1.X < r2.Y} P_{r2}(r))) =$

{ $[1, 3, 6, 4, 12], [1, 4, 18, 9, 21], [3, 3, 4, 6, 6], [3, 4, 6, 9, 7], [4, 3, 6, 16, 12], [4, 4, 9, 24, 14]$ }

$\text{proj}(X, Y, S) =$

$\pi_{X=r1.Y, Y=r2.Y, S} (\sigma_{SX1 > r1.Y} (\Gamma_{r1.Y, r2.Y, SX1 = \text{sum}(r1.X), SY1 = \text{sum}(r1.Y), S = \text{sum}(r2.X)} (P_{r1}(r) \bowtie_{r1.X < r2.Y} P_{r2}(r)))) =$
 $\{[1, 3, 12], [1, 4, 21], [3, 3, 6], [3, 4, 7], [4, 3, 12], [4, 4, 14]\}$

3. Uvažujte vstupný rozvrh (rl resp. wl sú žiadosti o read-lock resp. write-lock)

start1, start2, rl1(X), start3, wl2(Y), rl1(Y), rl3(X).

a) Uveďte a vysvetlite po každej prečítanej operácii akciu, ktorú vykoná systém používajúci dvojfázové zamykanie bez prevencie deadlocku. (2)

Uvedieme len akcie, ktoré súvisia s 2PL:

start1

start2

rl1(X) pridelí read-lock na X transakcii T1

start3

wl2(Y) pridelí write-lock na Y transakcii T2

rl1(Y) nechá transakciu T1 čakať na pridelenie read-lock na Y

rl3(X) pridelí read-lock na X transakcii T3

b) Uveďte a vysvetlite po každej prečítanej operácii akciu, ktorú vykoná systém používajúci dvojfázové zamykanie a metódu wait-or-die. (2)

Uvedieme len akcie, ktoré súvisia s 2PL a wait-or-die:

start1 pridelí časovú pečiatku TS(T1)

start2 pridelí časovú pečiatku TS(T2)

rl1(X) pridelí read-lock na X transakcii T1

start3 pridelí časovú pečiatku TS(T3)

wl2(Y) pridelí write-lock na Y transakcii T2

rl1(Y) keďže $TS(T1) < TS(T2)$, nechá transakciu T1 čakať na pridelenie read-lock na Y

rl3(X) pridelí read-lock na X transakcii T3

c) Uveďte a vysvetlite po každej prečítanej operácii akciu, ktorú vykoná systém používajúci dvojfázové zamykanie a metódu kill-or-wait. (2)

Uvedieme len akcie, ktoré súvisia s 2PL a kill-or-wait:

start1 pridelí časovú pečiatku TS(T1)

start2 pridelí časovú pečiatku TS(T2)

rl1(X) pridelí read-lock na X transakcii T1

start3 pridelí časovú pečiatku TS(T3)

wl2(Y) pridelí write-lock na Y transakcii T2

rl1(Y) keďže $TS(T1) < TS(T2)$, abortuje T2 a následne pridelí read-lock na Y transakcii T1

rl3(X) pridelí read-lock na X transakcii T3

4. Na výpočet SQL dotazu *select r.X, r.Y order by r.X, r.Y* sa použije merge-sort, ktorý má k dispozícii 4 bloky operačnej pamäte. Bloky na disku a v operačnej pamäti sú rovnako veľké, do jedného bloku sa zmestia 2 záznamy relácie r.

a) Popíšte organizáciu pamäte a znázornite priebeh (uved'te význačné medzivýsledky) algoritmu merge-sort pre reláciu

$r(X, Y) = \{[7, 8], [4, 9] \mid [2, 1], [7, 6] \mid [3, 9], [5, 7] \mid [8, 1], [7, 9] \mid [3, 4], [2, 2] \mid [2, 1], [3, 2] \mid [1, 6], [1, 5] \mid [4, 9], [4, 2] \mid [3, 5], [1, 7] \mid [4, 6], [1, 8]\}$.

(Znak \mid slúži ako oddeľovač blokov.) (2)

V prvej fáze sa rezervujú všetky 4 bloky operačnej pamäte pre vstup r. Po internom triedení prečítaných blokov sa všetky použité bloky zapisujú do utriedeného súboru BEH1 (predpokladáme, že bloky r sa čítajú v takom poradí ako sú napísané):

$[2, 1], [3, 9] \mid [4, 9], [5, 7] \mid [7, 6], [7, 8] \mid [7, 9], [8, 1]$

Pokračuje sa čítaním nasledujúcich blokov r. Výstupom je utriedený súbor BEH2:

$[1, 5], [1, 6] \mid [2, 1], [2, 2] \mid [3, 2], [3, 4] \mid [4, 2], [4, 9]$

Pokračuje sa čítaním nasledujúcich blokov r. Výstupom je utriedený súbor BEH3:

$[1, 7], [1, 8] \mid [3, 5], [4, 6]$

Relácia r sa dočítala. V druhej fáze sa zlučujú utriedené behy do väčších behov. V operačnej pamäti sa rezervujú 3 bloky pre vstup behov (1 blok pre 1 beh), 1 blok pre výstup.

Výstupom je utriedený beh:

$[1, 5], [1, 6] \mid [1, 7], [1, 8] \mid [2, 1], [2, 1] \mid [2, 2], [3, 2] \mid [3, 4], [3, 5] \mid [3, 9], [4, 2] \mid [4, 6], [4, 9] \mid [4, 9], [5, 7] \mid [7, 6], [7, 8] \mid [7, 9], [8, 1]$

b) Uved'te počet diskových operácií (vrátane výstupných operácií) pri vykonávaní daného dotazu pre reláciu z úlohy a). Vysvetlite. (2)

V prvej fáze sa každý blok relácie r prečíta a rovnaký počet blokov sa zapíše: 10 vstupných a 10 výstupných operácií.

V druhej fáze sa z utriedených behov prečíta 10 blokov, zapíše sa 10 blokov.

Celkovo sa urobí 40 diskových operácií (20 vstupných a 20 výstupných).