

25/1/2021 Úvod do databáz, skúškový test, max 60 bodov

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) Sformulujte bezpečný dotaz v Datalogu (6) a relačnej algebre (6) na dvojice  $[P, K]$  také, že pijan P pri niektorej návšteve krčmy K vypil každý alkohol, ktorý K čapuje; a pri každej svojej návšteve K vypil niektorý z alkoholov, ktoré ľúbi.

Datalog:

```
answer(P, K) ←  
  navstivil(I, P, K),  
  not capuje_nevypil(I),  
  not niekedy_nevypil_nic_oblubene(P, K).
```

```
capuje_nevypil(I) ←  
  navstivil(I, _, K),  
  capuje(K, A),  
  not v(I, A).
```

```
v(I, A) ←  
  vypil(I, A, _).
```

```
niekedy_nevypil_nic_oblubene(P, K) ←  
  navstivil(I, P, K),  
  not vypil_nieco_oblubene(I).
```

```
vypil_nieco_oblubene(I) ←  
  navstivil(I, P, K),  
  lubi(P, A),  
  vypil(I, A, _).
```

Relačná algebra:

$v = \pi_{\text{Idn}, \text{Alkohol}}(\text{vypil})$

$\text{capuje\_nevypil} = \pi_{\text{Idn}}(\pi_{\text{Idn}, \text{Alkohol}}(\pi_{\text{Idn}, \text{Krcma}}(\text{navstivil}) \bowtie \text{capuje}) - v)$

$\text{vypil\_nieco\_oblubene} = \pi_{\text{Idn}}(\text{navstivil} \bowtie \text{lubi} \bowtie \text{vypil})$

$\text{niekedy\_nevypil\_nic\_oblubene} = \pi_{\text{Pijan}, \text{Krcma}}(\text{navstivil} \triangleright \text{vypil\_nieco\_oblubene})$

/\* answer \*/

$\pi_{\text{Pijan}, \text{Krcma}}(\text{navstivil} \triangleright \text{capuje\_nevypil}) \triangleright \text{niekedy\_nevypil\_nic\_oblubene}$

b) Sformulujte bezpečný dotaz v Datalogu **(6)** a SQL **(6)** na dvojice [P, K] také, že pijan P ľúbi aspoň 7 alkoholov, ktoré K čapuje; a žiaden zo svojich obľúbených alkoholov nevypil v K celkovom množstve väčšom než 100 (čo zahŕňa aj prípad, že P v K nikdy nevypil alkohol, ktorý ľúbi).

Datalog:

```
answer(P, K) ←  
  subtotal(lubi_capuje(P, K, A), [P, K], [C = count(A)]),  
  C >= 7,  
  not niecoho_privela(P, K).
```

```
lubi_capuje(P, K, A) ←  
  lubi(P, A),  
  capuje(K, A).
```

```
niecoho_privela(P, K) ←  
  subtotal(nvl(_, P, K, A, M), [P, K, A], [S = sum(M)]),  
  S >= 100.
```

```
nvl(I, P, K, A, M) ←  
  navstivil(I, P, K),  
  lubi(P, A),  
  vypil(I, A, M).
```

```
SQL:  
with lubi_aspon7 as  
(  
  select l.Pijan, l.Krcma  
  from lubi l, capuje c  
  where l.Alkohol = c.Alkohol  
  group by l.Pijan, l.Krcma  
  having count(l.Alkohol) >= 7  
)  
select la7.Pijan, la7.Krcma  
from lubi_aspon7 la7  
where not exists (  
  select *  
  from navstivil n, lubi l, vypil v  
  where la7.Pijan = n.Pijan and la7.Krcma = n.Krcma and la7.Alkohol = l.Alkohol and n.Idn= v.Idn and  
    n.Pijan = l.Pijan and l.Alkohol = v.Alkohol  
  group by n.Pijan, n.Krcma, l.Alkohol  
  having sum(v.Mnozstvo) >= 100  
)
```

2. Daná je relácia  $r(A, B, C, D, E, F, G, H)$  s funkčnými závislosťami  
 $A \rightarrow G$ ,  $AD \rightarrow BG$ ,  $AC \rightarrow BDE$ ,  $E \rightarrow B$ ,  $BG \rightarrow CF$ ,  $BE \rightarrow D$ ,  $BH \rightarrow E$ ,  $BCD \rightarrow GH$ ,  $BDG \rightarrow A$ .

Uvažujte dekompozíciu  $r$ :

$(A, B, D)$ ,  $(A, G)$ ,  $(A, C, E)$ ,  $(B, C, F, G)$ ,  $(B, D, G, H)$ ,  $(B, E)$ ,  $(D, E)$ ,  $(E, H)$ .

a) Rozhodnite či je daná dekompozícia bezstratová (t.j. či sa spája bezstratovo). Ak ÁNO, zdôvodnite. Ak NIE, vysvetlite prečo na konkrétnom kontrapríklade. **(6)**

Použijeme algoritmus chase.

	A	B	C	D	E	F	G	H
<b>ABD</b>	a1	a2		a4			a7 <sub>1: A→G</sub>	
<b>AG</b>	a1						a7	
<b>ACE</b>	a1	a2 <sub>2: E→B</sub>	a3	a4 <sub>4: BE→D</sub>	a5	a6 <sub>3: BG→F</sub>	a7 <sub>1: A→G</sub>	a8 <sub>5: BCD→H</sub>
<b>BCFG</b>		a2	a3	a4 <sub>4: BE→D</sub>		a6	a7	a8 <sub>5: BCD→H</sub>
<b>BDGH</b>		a2	a2 <sub>3: BG→C</sub>	a4		a6 <sub>3: BG→F</sub>	a7	a8
<b>BE</b>		a2			a5			
<b>DE</b>		a2 <sub>2: E→B</sub>		a4	a5			
<b>EH</b>		a2 <sub>2: E→B</sub>			a5			a8

Riadok ACE obsahuje len symboly  $a^*$ . **ÁNO, daná dekompozícia je bezstratová.**

b) Rozhodnite či daná dekompozícia je v Boyce-Coddovej normálnej forme. Ak ÁNO, zdôvodnite. Ak NIE, vysvetlite prečo na konkrétnom kontrapríklade. **(6)**

Definícia. Relačná schéma  $[r, F]$  je v BCNF, keď pre každú platnú kánonickú funkčnú závislosť  $X \rightarrow Y$  platí, že  $X$  je nadkľúč v  $r$ .

V BDGH platí  $BH \rightarrow D$ , ale neplatí  $BH \rightarrow G$ . **Daná dekompozícia NIE JE v BCNF.**

3. Uvažujte dotaz v relačnom kalkule (konečnú reláciu  $r(X, Y, Z)$  chápajte ako EDB predikát):

$\{[X1, X2]: (\exists Y1 \exists Z1 r(X1, Y1, Z1)) \wedge (\exists Y2 \exists Z2 r(X2, Y2, Z2)) \wedge$   
 $((\forall Y3 \forall Z3 \forall Z4 (r(X1, Y3, Z3) \Leftrightarrow r(X2, Y3, Z4)) \vee$   
 $\neg (\exists Y4 \exists Z5 \exists Z6 r(X1, Y4, Z5) \wedge r(X2, Y4, Z6)))\}$

a) Definujte v Datalogu predikát  $answer(., .)$  ekvivalentný danému dotazu. **(6)**

```
answer(X1, X2) ←  
  r(X1, _, _),  
  r(X2, _, _),  
  not neq(X1, X2).
```

```
answer(X1, X2) ←  
  r(X1, _, _),  
  r(X2, _, _),  
  not and(X1, X2).
```

```
neq(X1, X2) ←  
  r(X1, Y3, _),  
  r(X2, _, _),  
  not r(X2, Y3, _).
```

```
neq(X1, X2) ←  
  r(X2, Y3, _),  
  r(X1, _, _),  
  not r(X1, Y3, _).
```

```
and(X1, X2) ←  
  r(X1, Y4, _),  
  r(X2, Y4, _).
```

b) Vypočítajte výsledok daného dotazu pre

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

$neq(X1, X2) = \{[0, 1], [0, 3], [1, 0], [1, 3], [3, 0], [3, 1]\}$

$and(X1, X2) = \{[0, 0], [0, 1], [1, 0], [1, 1], [1, 3], [3, 1], [3, 3]\}$

$answer(X1, X2) = \{[0, 0], [0, 3], [1, 1], [3, 0], [3, 3]\}$

Výsledkom dotazu je relácia  **$\{[0, 0], [0, 3], [1, 1], [3, 0], [3, 3]\}$** .

4. Predpokladajte, že v operačnej pamäti sú voľné presne 4 bloky. Bloky na disku a v operačnej pamäti sú rovnako veľké, v každom bloku relácie  $r$  sú uložené 2 záznamy relácie  $r(X, Y) = \{[7, 8], [4, 9] \mid [2, 1], [7, 6] \mid [3, 9], [5, 7] \mid [8, 1], [7, 9] \mid [3, 4], [2, 2] \mid [2, 1], [3, 2] \mid [1, 6], [1, 5] \mid [4, 9], [4, 2] \mid [3, 5], [1, 7] \mid [4, 6], [1, 8]\}$ . (Znak  $\mid$  slúži ako oddeľovač blokov, v ktorých je uložená relácia  $r$ .) Na výpočet SQL dotazu  $\text{select } r.X, r.Y \text{ order by } r.X, r.Y$  sa používa fyzický operátor merge-sort (externé triedenie).

a) Uved'te význačné medzivýsledky výpočtu (stav operačnej pamäte a disku) pre danú reláciu a dotaz. **(6)**

1. fáza, vytváranie utriedených behov:

Prvé 4 bloky sa prečítajú do RAM:

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

Po internom utriedení záznamov sa na disk zapíše beh

$\text{run1} = [[2, 1], [3, 9] \mid [4, 9], [5, 7] \mid [7, 6], [7, 8] \mid [7, 9], [8, 1]]$ .

Ďalšie 4 bloky sa prečítajú do RAM:

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

Po internom utriedení záznamov sa na disk zapíše beh

$\text{run2} = [[1, 5], [1, 6] \mid [2, 1], [2, 2] \mid [3, 2], [3, 4] \mid [4, 2], [4, 9]]$ .

Posledné 2 bloky sa prečítajú do RAM:

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

Po internom utriedení záznamov sa na disk zapíše beh

$\text{run3} = [[1, 7], [1, 8] \mid [3, 5], [4, 6]]$ .

2. fáza, spájanie utriedených behov do dlhších behov:

Behy  $\text{run1}$ ,  $\text{run2}$ ,  $\text{run3}$  sa spoja do jedného behu v jedinom prechode. Pre čítanie každého z behov  $\text{run1}$ ,  $\text{run2}$ ,  $\text{run3}$  sa rezervuje v RAM 1 blok. Pre výstup sa rezervuje v RAM 1 blok. Výsledný beh:

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

b) Uved'te počet vstupných a počet výstupných diskových operácií pre danú reláciu a dotaz (predpokladajte, že výsledný beh sa zapíše na disk). **(6)**

V 1. fáze sa každý blok  $r$  raz prečíta (10 čítaní z disku) a raz zapíše (10 zápisov na disk). V druhej fáze sa každý blok raz prečíta (10 čítaní z disku) a rovnaký počet blokov sa zapíše (10 zápisov na disk).

**Celkovo sa urobí 20 vstupných a 20 výstupných diskových operácií.**

*(Jedna vstupná operácia sa dá ušetriť tým, že prvý blok z behu  $\text{run3}$ , ktorý je po prvej fáze v RAM, sa použije v druhej fáze bez opätovného čítania z disku.)*