

5/1/2022 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 krčmy K, ktoré čapujú borovičku, a zároveň každý borovičkový expert pil v K borovičku len pri jednej svojej návšteve. Borovičkový expert je pijan, ktorý borovičku ľúbi a ochutnal ju v každej krčme, ktorá ju čapuje.

Datalog:

$\text{answer}(\text{K}) \leftarrow$

$\text{capuje}(\text{K}, \text{borovicka}),$
 $\text{not nespravna_krcma}(\text{K}).$

$\text{nespravna_krcma}(\text{K}) \leftarrow$

$\text{navstivil}(\text{I1}, \text{P}, \text{K}), \text{vypil}(\text{I1}, \text{borovicka}, _),$
 $\text{navstivil}(\text{I2}, \text{P}, \text{K}), \text{vypil}(\text{I2}, \text{borovicka}, _),$
 $\text{lubi}(\text{P}, \text{borovicka}),$
 $\text{not neexpert}(\text{P}),$
 $\text{not I1} = \text{I2}.$

$\text{neexpert}(\text{P}) \leftarrow$

$\text{lubi}(\text{P}, \text{borovicka}),$
 $\text{capuje}(\text{K}, \text{borovicka}),$
 $\text{not niekedy_vypil}(\text{P}, \text{K}).$

$\text{niekedy_vypil}(\text{P}, \text{K}) \leftarrow$

$\text{navstivil}(\text{I}, \text{P}, \text{K}),$
 $\text{vypil}(\text{I}, \text{borovicka}, _).$

Relačná algebra:

$\text{niekedy_vypil} := \pi_{\text{Pijan}, \text{Krcma}} (\text{navstivil} \bowtie \sigma_{\text{Alkohol} = \text{'borovicka'}} (\text{vypil}))$

$\text{neexpert} := \pi_{\text{Pijan}} ($

$\pi_{\text{Pijan}, \text{Krcma}} (\sigma_{\text{Alkohol} = \text{'borovicka'}} (\text{lubi}) \times \sigma_{\text{Alkohol} = \text{'borovicka'}} (\text{capuje})) \triangleright$
 $\text{niekedy_vypil})$

$\text{nespravna_krcma} := \pi_{\text{Krcma}} ($

$(\pi_{\text{Pijan}, \text{Krcma}} (\text{navstivil} \bowtie \sigma_{\text{Alkohol} = \text{'borovicka'}} (\text{vypil})) \bowtie \pi_{\text{Pijan} = \text{P2} \wedge \text{Krcma} = \text{K2} \wedge \text{Idn} \neq \text{I2}}$
 $\rho_{\text{I2}, \text{P2}, \text{K2}} (\pi_{\text{Idn}, \text{Pijan}, \text{Krcma}} (\text{navstivil} \bowtie \sigma_{\text{Alkohol} = \text{'borovicka'}} (\text{vypil}))))$
 $\triangleright \text{neexpert})$

$/* \text{answer} */$

$\sigma_{\text{Alkohol} = \text{'borovicka'}} (\text{capuje}) \triangleright \text{nespravna_krcma}$

b) Sformulujte bezpečný dotaz v Datalogu **(6)** a SQL **(6)** na trojice [P, K, A] také, že pijan P vypil počas niektorej návštevy krčmy K väčšie množstvo alkoholu A než ktorýkoľvek iný pijan dokopy počas všetkých návštev krčmy K.

Datalog:

```
answer(P, K, A) ←  
  navstivil(I, P, K),  
  vypil(I, A, M),  
  not iny_dostihol(P, K, A, M).
```

```
iny_dostihol(P, K, A, M) ←  
  navstivil(I, P, K),  
  vypil(I, A, M),  
  subtotal(vypite(_, P2, K, A, M), [P2, K, A], [T = sum(M)]),  
  P ≠ P2,  
  T >= M.
```

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

```
SQL:  
with  
totals as  
(  
  select n.Pijan, n.Krcma, v.Alkohol, Total = sum(v.Mnozstvo)  
  from navstivil n, vypil v  
  where n.Idn = v.Idn  
  group by n.Pijan, n.Krcma, v.Alkohol  
)  
select n.Pijan, n.Krcma, v.Alkohol  
from navstivil n, vypil v  
where n.Idn = v.Idn and not exists ( /* iny_dostihol */  
  select *  
  from totals t  
  where t.Krcma = n.Krcma and t.Alkohol = v.Alkohol and t.Pijan <> n.Pijan and t.Total >= v.Mnozstvo  
)
```

2.

a) Definujte pojem *bezstratovosť spojenia dekompozície relačnej schémy*. Uvažujte reláciu $r(A, B, C, D, E)$ s funkčnými závislosťami $A \rightarrow CE$, $AD \rightarrow B$, $BC \rightarrow D$, $BE \rightarrow A$. Rozhodnite, či sa dekompozícia $r_1(A, C, D, E)$, $r_2(A, B, E)$ spája bezstratovo. Ak áno, zdôvodnite. Ak nie, uveďte príklad konkrétneho naplnenia relácií r , r_1 a r_2 , ktoré je v rozpore s definíciou. **(6)**

Definícia: Dekompozícia r do r_1, \dots, r_N je bezstratová (spája sa bezstratovo), keď pre každú populáciu r , ktorá spĺňa integritné obmedzenia schémy (v tomto prípade dané funkčné závislosti) platí

$$r = \pi_{r_1}(r) \bowtie \dots \bowtie \pi_{r_N}(r).$$

Pre test bezstratovosti použijeme algoritmus chase:

	A	B	C	D	E
ACDE	a1	b12	a3	a4	a5
ABE	a1	a2	b23	b24	a5

$A \rightarrow CE$

	A	B	C	D	E
ACDE	a1	b12	a3	a4	a5
ABE	a1	a2	a3	b24	a5

Tu chase končí, daná dekompozícia sa nespája bezstratovo. Kontrapríkladom je napr. populácia

$r = \{\{a1, b12, a3, a4, a5\}, \{a1, a2, b23, b24, a5\}\}$, ktorá spĺňa všetky funkčné závislosti platné v r , ale

$r_1(A, C, D, E) \bowtie r_2(A, B, E) =$

$\{\{a1, a3, a4, a5\}, \{a1, b23, b24, a5\}\} \bowtie \{\{a1, b12, a5\}, \{a1, a2, a5\}\}$

obsahuje n -ticu $\{a1, a2, a3, a4, a5\}$, ktorá nepatrí do r .

b) Pre relačnú schému z úlohy a) nájdite všetky dekompozície do dvoch relácií, ktoré sa spájajú bezstratovo a zároveň sú v Boyce-Coddovej normálnej forme (hľadaná dekompozícia nemusí zachovať všetky funkčné závislosti). Zdôvodnite. **(6)**

Enumerovať všetky dekompozície a pre každú testovať BCNF a bezstratovosť je pracné, skúsme to inak.

Lema. Žiadna relácia s aspoň 4 atribútmi (pre dané atribúty a funkčné závislosti) nie je v BCNF.

Dôkaz:

ABCDE nie je v BCNF, lebo platí $A \rightarrow C$ a A nie je nadkľúč ABCDE. Iné 5-atribútové relácie nie sú.

Uvažujme 4-atribútové relácie s atribútom A . K relácii AC nemôžeme pridať B ani D , lebo výsledná relácia by nebola v BCNF kvôli $A \rightarrow C$. Ani k relácii AE nemôžeme pridať B ani D , z rovnakého dôvodu. Z toho vyplýva, že atribút A nemôže byť v 4-atribútovej relácii, ktorá je v BCNF.

Jediná 4-atribútová relácia bez atribútu A je $BCDE$. Tá nie je v BCNF kvôli $BC \rightarrow D$. Tým je lema dokázaná.

Uvažujme 3-atribútové relácie, ktoré obsahujú A . ACE je v BCNF. Z dôkazu lemy vyplýva, že nepotrebujeme uvažovať iné rozšírenia AC a AE . Ostáva teda už len ABD , ktorá je BCNF.

Všetky 3-atribútové relácie, ktoré neobsahujú A , sú v BCNF: BCD , BCE , BDE , CDE .

Podme párovať 3-atribútové BCNF relácie do dekompozícií. Dekompozícia $\{ACE, ABD\}$ sa spája bezstratovo, lebo spoločný atribút A je nadkľúč v ACE . $\{ACE, BCD\}$ sa nespája bezstratovo, lebo spoločný atribút C nie je nadkľúč v žiadnej z nich. Z rovnakého dôvodu sa nespájajú bezstratovo $\{ACE, BDE\}$, $\{ABD, BCE\}$, $\{ABD, CDE\}$. Ostatné páry nie sú dekompozície.

Jediná dekompozícia s požadovanými vlastnosťami (BCNF, bezstratovosť) je $\{ACE, ABD\}$. Po vynechaní A z ľubovoľnej relácie sa nespojí bezstratovo. Po vynechaní ľubovoľného iného atribútu prestane byť dekompozíciou.

3. Uvažujte SQL dotaz nad reláciami $r(A, B)$, $s(X, Y, Z)$
 $\text{select } r.A, r.B, s.Z \text{ from } r \text{ full outer join } s \text{ on } (r.A = s.X \text{ and } r.B = s.Y).$

a) Vyjadrite daný dotaz ekvivalentne v relačnom kalkule. **(6)**

```
{[A, B, Z]:  
  (r(A, B)  $\wedge$  s(A, B, Z))  $\vee$   
  (r(A, B)  $\wedge$  Z = null  $\wedge$   $\neg$  ( $\exists Z$  s(A, B, Z)))  $\vee$   
  ( $\exists X \exists Y$  s(X, Y, Z)  $\wedge$  A = null  $\wedge$  B = null  $\wedge$   $\neg$  ( $\exists A \exists B$  r(A, B, Z)))  
}
```

b) Vyjadrite daný dotaz ekvivalentne v SQL bez použitia kľúčového slova *join*. **(6)**

```
(  
select r.A, r.B, s.Z  
from r, s  
where r.A = s.X and r.B = s.Y  
)  
union  
(  
select r.A, r.B, null as Z  
from r  
where not exists (  
  select *  
  from s  
  where r.A = s.X and r.B = s.Y)  
)  
union  
(  
select null as A, null as B, s.Z  
from s  
where not exists (  
  select *  
  from r  
  where r.A = s.X and r.B = s.Y)  
)
```

4. M je počet blokov rezervovaných v operačnej pamäti (RAM) pre výpočet dotazu

$\pi_{X,Y}(r \bowtie_{Y=W} s)$.

Relácia $r(X, Y)$ je uložená v $3M$ diskových blokoch, relácia $s(W, Z)$ je uložená v $2M$ diskových blokoch, v reláciách nie sú duplikáty ani *null* hodnoty. Bloky na disku a v RAM sú rovnako veľké.

a) Popíšte organizáciu RAM pri výpočte dotazu metódou nested-loop join. (6)

$M - 2$ blokov sa použije pre postupné čítanie relácie s , lebo je menšia ako r

1 blok sa použije pre postupné čítanie relácie r

1 blok sa použije pre zápis výstupných n -tíc

b) Vyjadrite ako funkciu M počet vstupných diskových operácií (z disku do RAM), ktoré nested-loop join urobí na danom dotaze. Vysvetlite (uved'te stručne momenty vo výpočte, kedy počet vstupných diskových operácií rastie).

$M - 2$ blokov relácie s sa prečíta do RAM.

Postupne sa prečíta $3M$ blokov relácie r do RAM (po jednom bloku, pričom záznamy r a s , ktoré sú v danej chvíli v RAM, sa joinujú; výsledné záznamy sa zapisujú do výstupného bloku; keď je výstupný blok plný, zapíše sa na disk).

Nasledujúcich $M - 2$ blokov relácie s sa prečíta do RAM.

Postupne sa prečíta $3M$ blokov r do RAM.

Posledné 4 bloky s sa prečítajú do RAM.

Postupne sa prečíta $3M$ blokov r do RAM.

To je dokopy $2 * (M - 2) + 3 * 3M + 4 = 11M$ vstupných operácií.

Výpočet daného dotazu metódou nested-loop join vyžaduje **11M vstupných diskových operácií**.

Tento počet sa dá trochu zredukovať „recykláciou“ 1 bloku r , ktorý je v RAM v momente, keď sa z disku do RAM číta nasledujúca časť relácie s (relácia r sa striedavo číta raz spredu, raz odzadu.) V tomto prípade sa týmto spôsobom ušetrí 2 diskové operácie.