

13/2/2018 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 také dvojice $[P, K]$, že pijan P ľúbi všetky alkoholy, ktoré sa v krčme K čapujú; a niektorý z nich vypil v každej krčme, ktorú navštívil.

Datalog:

```
answer(P, K) ←  
  lubi(P, A),  
  capuje(K, A),  
  not nelubi_niektory(P, K),  
  not niekde_nevypil(P, A).
```

```
nelubi_niektory(P, K) ←  
  lubi(P, _),  
  capuje(K, A),  
  not lubi(P, A).
```

```
niekde_nevypil(P, A) ←  
  lubi(P, A),  
  navstivil(_, P, K),  
  not v(P, A, K).
```

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

Relačná algebra:

$v = \pi_{P, A, K} (\text{navstivil} \bowtie \text{vypil})$

$\text{niekde_nevypil} = \pi_{P, A} ((\text{lubi} \bowtie \text{navstivil}) \triangleright v)$

$\text{nelubi_niektory} = \pi_{P, K} ((\pi_P (\text{lubi}) \bowtie \text{capuje}) \triangleright \text{lubi})$

$((\text{lubi} \bowtie \text{capuje}) \triangleright \text{nelubi_niektory}) \triangleright \text{niekde_nevypil} /* \text{answer} */$

b) Sformulujte bezpečný dotaz v relačnom kalkule (6) a SQL (6) na také dvojice [P, K], že pijan P ľúbi aspoň 10 alkoholov, ktoré sa čapujú v krčme K.

Relačný kalkul:

```
{[P, K]:  
  ∃ C  
  ♥ C = count(A)  
    (ľubi(P, A) ∧ capuje(K, A)) ∧ L >= 10  
}
```

SQL:

```
select l.Pijan, c.Krcma  
from ľubi l, capuje c  
where l.Alkohol = c.Alkohol  
group by l.Pijan, c.Krcma  
having count(distinct l.Alkohol) >= 10
```

2. a) Definujte pojmy kľúča a nadkľúča relačnej schémy. (6)

Uvažujme relačnú schému $[r, F]$, kde r je množina atribútov a F je množina funkčných závislostí.

Nadkľúč danej relačnej schémy $[r, F]$ je množina atribútov s taká, že $s \subseteq r$ a $s^+ = r$, kde s^+ je uzáver množiny atribútov s vzhľadom na F .

Kľúč danej relačnej schémy $[r, F]$ je množina atribútov k taká, že k je nadkľúč $[r, F]$ a zároveň žiadna vlastná podmnožina k nie je nadkľúčom $[r, F]$.

b) Napíšte algoritmus (pseudokód), ktorý v polynomiálnom čase nájde nejaký kľúč danej relačnej schémy. (6)

Uvažujme relačnú schému $[r, F]$, kde r je množina atribútov a F je množina funkčných závislostí.

Uvažujme funkčnú závislosť $r \rightarrow r$ a skracujme postupne jej ľavú stranu (postupným vynechávaním atribútov) tak, aby skrátaná ľavá strana naďalej určovala r . Atribút možno vynechať, ak zostávajúce atribúty naďalej určujú r . Keďže pre každý atribút r sa uzáver zvyšnej množiny atribútov vzhľadom na F (s polynomiálnou časovou zložitou) počíta práve raz, tento algoritmus má polynomiálnu časovú zložitú vzhľadom na veľkosť $[r, F]$.

Input: $[r, F]$

$k = r$;

for each $A \in k$

 if $((k - A)^+ = r)$

$k = k - A$;

Output: k

c) Existuje algoritmus, ktorý v polynomiálnom čase rozhodne, či daná relačná schéma má dva (rôzne) kľúče? Ak nie, zdôvodnite. Ak áno, napíšte pseudokód takého algoritmu. (6)

Áno, existuje. Najskôr nájdeme nejaký kľúč k algoritmom z úlohy b). Potom hľadáme ľubovoľný atribút $A \in k$ taký, že $(r - A)^+ = r$. Ak taký existuje, tak $r - A$ je nadkľúč danej schémy, ktorý neobsahuje kľúč k , teda obsahuje iný kľúč. Tento druhý kľúč vieme nájsť podobne ako ten prvý, minimalizáciou ľavej strany funkčnej závislosti $r - A \rightarrow r$

Ak pre všetky $A \in k$ platí $(r - A)^+ \neq r$, tak k je jediný kľúč.

Input: $[r, F]$

$k = r$;

for each $(A \in k)$

 if $((k - A)^+ = r)$

$k = k - A$;

$k2exists = FALSE$;

for each $(A \in k)$

 if $((r - A)^+ = r)$

$k2exists = TRUE$;

Output: $k2exists$

Aj tento algoritmus má zrejme polynomiálnu časovú zložitú.

3. Dané sú relácie $r(X, Y)$ a $s(X, Y)$, bez duplikátov a NULL hodnôt.

a) Zapište nasledujúci SQL dotaz ekvivalentne v relačnom kalkule **(6)**:

```
select distinct r1.Y
from r r1, r r2, s s1
where r1.X < 99 and (r1.X = r2.X or r1.X = s1.X)
```

{RY1:

$\exists RX1, RX2, RY2, SX, SY$

$r(RX1, RY1) \wedge r(RX2, RY2) \wedge s(SX, SY) \wedge RX1 < 99 \wedge (RX1 = RX2 \vee RX1 = SX1)$

}

b) Rozhodnite či je dotaz z úlohy a) ekvivalentný dotazu

```
select distinct r1.Y
from r r1, r r2
where r1.X < 99 and r1.X = r2.X
```

Odpoveď ÁNO resp. NIE zdôvodnite. **(6)**

Nie. Napríklad pre relácie $r(X, Y) = \{[1, 1]\}$, $s(X, Y) = \{\}$ je výsledkom tohto dotazu $\{1\}$, zatiaľ čo výsledkom dotazu z úlohy a) je prázdna množina.

4. Uvažujte rozvrh s_1, s_2, s_3 ,

$r_1(X), r_2(Z), r_1(Z), r_3(X), w_3(X), w_3(Y), c_3, c_1, w_1(X), r_2(Y), w_2(Z), w_2(Y), c_2$.

Rozhodnite, či tento rozvrh mohol vygenerovať scheduler systému, ktorý na izoláciu transakcií používa metódu časových pečiatok. Odpoveď ÁNO resp. NIE zdôvodnite. (6)

Nie. Tento rozvrh nie je konflikt-sériovateľný kvôli konfliktom $r_1(X) \rightarrow w_3(X) \rightarrow w_1(X)$. Metóda časových pečiatok garantuje, že výstupný rozvrh je vždy konflikt-sériovateľný.