

11/1/2023 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 trojice $[P, K, A]$, že krčma K čapuje alkohol A ; a zároveň P je jediný pijan, ktorý ľúbi A a ktorý alkohol A v krčme K nevypil.

Datalog:

```
answer(P, K, A) ←  
    capuje(K, A),  
    lubi(P, A),  
    not niekedy_vypil(P, K, A),  
    not iny_nevypil(P, K, A).
```

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

```
iny_nevypil(P, K, A) ←  
    capuje(K, A),  
    lubi(P, A),  
    lubi(P2, A),  
    not P2 = P,  
    not niekedy_vypil(P2, K, A).
```

Relačná algebra:

```
niekedy_vypil =  $\pi_{\text{Pijan}, \text{Krcma}, \text{Alkohol}}$  ( $\text{navstivil} \bowtie \text{vypil}$ );
```

```
iny_nevypil =  $\pi_{\text{Pijan}, \text{Krcma}, \text{Alkohol}}$  (( $\text{capuje} \bowtie \text{lubi} \bowtie_{\text{Alkohol} = A2 \wedge \text{Pijan} \neq P2} \rho_{P2, A2}(\text{lubi})$ )  $\triangleright$   $P2 = P \wedge \text{Krcma} = K \wedge \text{Alkohol} = A$   
 $\rho_{P, K, A}(\text{niekedy\_vypil})$ );
```

```
/* answer */
```

```
(( $\text{capuje} \bowtie \text{lubi}$ )  $\triangleright$  niekedy_vypil)  $\triangleright$  iny_nevypil;
```

b) Sformulujte bezpečný dotaz v Datalogu **(6)** a SQL **(6)** na dvojice krčiem [K1, K2] také, že krčmu K1 navštívilo apoň toľko (rôznych) pijanov ako K2; a každý alkohol vypitý v K2 sa vypil v K1 v aspoň takom celkovom množstve.

Datalog:

```
answer(K1, K2) ←  
    pocet(K1, C1),  
    pocet(K2, C2),  
    not niektery_alkohol_menej(K1, K2),  
    C1 >= C2.
```

```
pocet(K, C) ←  
    subtotal(n(K, P), [K], [C = count(P)]).
```

```
pocet(K, 0) ←  
    capuje(K, 0),  
    not n0(K).
```

```
n0(K) ←  
    navstivil(_, _, K).
```

```
n(K, P) ←  
    navstivil(_, P, K).
```

```
niektery_alkohol_menej(K1, K2) ←  
    mnozstvo(K1, A, S1),  
    mnozstvo(K2, A, S2),  
    S1 < S2.
```

```
mnozstvo(K, A, S) ←  
    subtotal(v(_, K, A, M), [K, A], [S = sum(M)]).
```

```
mnozstvo(K, A, 0) ←  
    navstivil(_, _, K),  
    vypil(_, A, _),  
    not v0(K, A).
```

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

```
v0(K, A) ←  
    navstivil(I, _, K),  
    vypil(I, A, _).
```

SQL:

with

pocet as

(

(

select n.Krcma, count(distinct n.Pijan) as C

from navstivil n

group by n.Krcma

)

union

(

select c.Krcma, 0 as C

from capuje c

where not exists (

select *

from navstivil n

where c.Krcma = n.Krcma)

)

),

mnozstvo as

(

(

select n.Krcma, v.Alkohol, sum(v.Mnozstvo) as S

from navstivil n, vypil v

where n.Idn = v.Idn

group by n.Krcma, v.Alkohol

)

union

(

select n.Krcma, v.Alkohol, 0 as S

from navstivil n, vypil v

where not exists (

select *

from navstivil n2, vypil v2

where n.Krcma = n2.Krcma and v.Alkohol = v2.Alkohol and n2.Idn = v2.Idn)

)

),

niektory_alkohol_menej as

(

select m1.Krcma as K1, m2.Krcma as K2

from mnozstvo m1, mnozstvo m2

where m1.Alkohol = m2.Alkohol and m1.S < m2.S

)

select p1.Krcma as K1, p2.Krcma as K2

from pocet p1, pocet p2

where p1.C >= p2.C and not exists (

select *

from niektory_alkohol_menej nam

where p1.Krcma = nam.K1 and p2.Krcma = nam.K2)

2.

a) Definujte pojem *bezstratovosť (spojenia) dekompozície relačnej schémy*. Uveďte algoritmus (zrozumiteľný pseudokód), ktorý testuje, či dekompozícia relačnej schémy do dvoch relačných schém je bezstratová. **(6)**

Dekompozícia relačnej schémy $[r, F]$ do $[r_1, F_1], \dots, [r_N, F_N]$ je bezstratová (spája sa bezstratovo), keď pre všetky naplnenia r (splňajúce integritné obmedzenia dané funkčnými závislosťami F^+) platí $r = r_1 \bowtie r_2 \bowtie \dots \bowtie r_N$.

Test bezstratovosti dekompozície so vstupom $[r, F], [r_1, F_1], [r_2, F_2]$, kde $[r_1, F_1], [r_2, F_2]$ tvoria dekompozíciu r :
if (množina spoločných atribútov r_1 a r_2 je nadkľúčom v r_1 alebo v r_2)

 výstup „dekompozícia je bezstratová“

else

 výstup „dekompozícia nie je bezstratová“

b) Dokážte korektnosť algoritmu z úlohy a), t.j. zdôvodnite prečo funguje. **(6)**

Predpokladajme, že ten test rozhodne, že daná dekompozícia je bezstratová. Bez ujmy na všeobecnosti predpokladajme, že množina spoločných atribútov relácií r_1, r_2 je nadkľúč v r_1 , označme túto množinu X . Ostatné atribúty r_1 označme $Y1$. Ostatné atribúty r_2 označme $Y2$.

Uvažujme ľubovoľné naplnenie r , splňajúce funkčné závislosti F .

Uvažujme ľubovoľnú n -ticu $t \in r$ (ak taká n -tica neexistuje, t.j. r je prázdna relácia, tak aj r_1 a r_2 sú prázdne, t.j. $r_1 \bowtie r_2$ je prázdna relácia). Táto n -tica je zložená z hodnôt atribútov X , ktoré označíme t_X a z hodnôt atribútov Y , ktoré označíme t_Y . Keďže r_1, r_2 tvoria dekompozíciu r , platí $Y = Y1 \cup Y2, [t_X, t_{Y1}] \in r_1, [t_X, t_{Y2}] \in r_2$. Teda $t \in r_1 \bowtie r_2$, t.j. $r \subseteq r_1 \bowtie r_2$.

Uvažujme ľubovoľnú n -ticu $t \in r_1 \bowtie r_2$, pričom t je zložená z t_X, t_{Y1}, t_{Y2} (ak taká n -tica neexistuje, tak aj r musí byť prázdna relácia, lebo $r_1 = \pi_{X, Y1}(r)$ a $r_2 = \pi_{X, Y2}(r)$). T.j. $[t_X, t_{Y1}] \in r_1$ a $[t_X, t_{Y2}] \in r_2$. V r musí existovať aspoň jeden záznam s hodnotami $X = t_X$ a $Y2 = t_{Y2}$, lebo $r_2 = \pi_{X, Y2}(r)$. Keďže X je nadkľúč v r_1 (teda platí $X \rightarrow Y$), v r_1 existuje práve jeden záznam s $X = t_X$, konkrétne záznam t_X, t_{Y1} . V každom zázname v s danou hodnotou $X = t_X$ je $Y = t_{Y1}$. Z toho vyplýva, že $t = [t_X, t_{Y1}, t_{Y2}] \in r$, t.j. $r_1 \bowtie r_2 \subseteq r$.

Dokázali sme, že ak test rozhodne, že daná dekompozícia je bezstratová, tak $r = r_1 \bowtie r_2$ pre každé prípustné naplnenie r .

Predpokladajme, že ten test rozhodne, že daná dekompozícia nie je bezstratová. To nastane v prípade, že spoločná množina atribútov r_1, r_2 , označíme ju X , nie je nadkľúčom v r_1 ani v r_2 . Vtedy r_1 obsahuje množinu atribútov $Z1$, že neplatí $X \rightarrow Z1$; r_2 obsahuje množinu atribútov $Z2$, že neplatí $X \rightarrow Z2$. Množinu atribútov r_1 resp. r_2 , ktoré funkčne závisia od X , označíme $Y1$ resp. $Y2$.

Nech r obsahuje nasledujúce 2 n -tice:

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[0, \dots, 0], Z2=[0, \dots, 0]$ a

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[1, \dots, 1], Z2=[1, \dots, 1]$.

Vtedy v r_1 sú 2 n -tice

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[0, \dots, 0]$ a

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[1, \dots, 1]$,

v r_2 sú dve n -tice

$X=[0, \dots, 0], Y2=[0, \dots, 0], Z2=[0, \dots, 0]$ a

$X=[0, \dots, 0], Y2=[0, \dots, 0], Z2=[1, \dots, 1]$.

Avšak v $r_1 \bowtie r_2$ sú 4 n -tice:

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[0, \dots, 0], Z2=[0, \dots, 0]$,

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[1, \dots, 1], Z2=[1, \dots, 1]$,

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[0, \dots, 0], Z2=[1, \dots, 1]$,

$X=[0, \dots, 0], Y1=[0, \dots, 0], Z1=[1, \dots, 1], Z2=[0, \dots, 0]$.

Napríklad pre toto naplnenie $r \neq r_1 \bowtie r_2$.

3. Uvažujte Datalogový program

$p(0, 1, 2).$

$p(Y, Z, X) \leftarrow p(Z, X, Y).$

$p(X, Y, Z) \leftarrow p(X, Z, Y), p(Y, X, Z).$

a) Zapište výpočet dotazu $? p(2, 1, W)$ naivnou evaluáciou v relačnej algebre. Uveďte výsledok dotazu. **(6)**

Naivná evaluácia:

$p = \{\};$ /* prázdna relácia $p(X, Y, Z)$ */

$\varphi($

$p = \{[0, 1, 2]\} \cup \pi_{Z, X, Y}(p) \cup (\pi_{X, Z, Y}(p) \bowtie \pi_{Y, X, Z}(p));$

$);$

/* result */

$\pi_Z(\sigma_{X=2 \wedge Y=1}(p))$

Simulácia naivnej evaluácie:

Inicializácia: $p = \{\};$

Po 1. iterácii φ : $p = \{[0, 1, 2]\}.$

Po 2. iterácii φ : $p = \{[0, 1, 2], [2, 0, 1]\}.$

Po 3. iterácii φ : $p = \{[0, 1, 2], [0, 2, 1], [1, 2, 0], [2, 0, 1]\}.$

Po 4. iterácii φ : $p = \{[0, 1, 2], [0, 2, 1], [1, 0, 2], [1, 2, 0], [2, 0, 1], [2, 1, 0]\}.$

Po 5. iterácii φ : $p = \{[0, 1, 2], [0, 2, 1], [1, 0, 2], [1, 2, 0], [2, 0, 1], [2, 1, 0]\}.$

Výsledok dotazu po finálnej selekcii a projekcii: **$\{0\}$** .

b) Rozhodnite, či pre daný program výpočet ľubovoľného dotazu naivnou evaluáciou skončí. Zdôvodnite. **(6)**

Program neobsahuje negáciu, takže naivná evaluácia skončí pre ľubovoľný dotaz.

4.

a) Vysvetlite metódu časových pečiatok pre izoláciu transakcií (uved'te kedy a aké akcie robí scheduler, akcie zapíšte v pseudokóde). **(6)**

Scheduler priradí každej transakcii T pri štarte časovú pečiatku $TS(T)$.

Každý objekt X má dve časové pečiatky, $TSR(X)$ a $TSW(X)$, obe sú inicializované na $-\infty$.

Pred vykonaním $r_T(X)$ vykoná scheduler toto:

if ($TS(T) < TSW(X)$)

 abort T ;

else

$TSR(X) := TS(T)$;

Pred vykonaním $w_T(X)$ vykoná scheduler toto:

if ($(TS(T) < TSR(X)) \vee (TS(T) < TSW(X))$)

 abort T ;

else

$TSW(X) := TS(T)$;

b) Uved'te príklad konflikt-sériovateľného a obnoviteľného rozvrhu, ktorý sa nedá generovať schedulerom, ktorý používa metódu časových pečiatok. Vysvetlite prečo má uvedený rozvrh požadované vlastnosti. **(6)**

$s1, s2, r2(X), w1(X), c1, c2$

Tento rozvrh je konflikt-sériovateľný, lebo obsahuje len jednu dvojicu konfliktných operácií; a obnoviteľný, lebo neobsahuje dirty read.

Scheduler urobí pre tento vstupný rozvrh nasledujúce akcie:

$s1$ pridelí $T1$ časovú pečiatku $TS(T1)$

$s2$ pridelí $T2$ časovú pečiatku $TS(T2)$, pričom $TS(T2) > TS(T1)$

$r2(X)$ $TSR(X) := TS(T2)$, lebo $TS(T2) > TSW(X)$

$w1(X)$ abort($T1$), lebo $TS(T1) < TSR(X)$