

12/12/2011 Úvod do databáz, zápočtový test, max 25 bodov, 90 min

0. Súhlasím so zverejnením výsledkov všetkých mojich testov z tohto kurzu vo forme [Meno, Výsledok] na webstránke prednášky. ÁNO (1), NIE (0).

ÁNO

1. Daná je databáza: $\text{capuje}(\text{Krcma}, \text{Alkohol}, \text{Cena})$, $\text{lubi}(\text{Pijan}, \text{Alkohol})$
 $\text{navstivil}(\text{Idn}, \text{Pijan}, \text{Krcma})$, $\text{vypil}(\text{Idn}, \text{Alkohol}, \text{Mnozstvo})$.

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

a) Sformulujte nasledujúci dotaz v relačnom kalkule (2), Datalogu (2) a SQL (2):
Nájdite dvojice pijanov [P1, P2], o ktorých platí, že P1 ľúbi niektorý z alkoholov,
ktorý P2 vypil v aspoň troch rôznych krčmách; a súčasne P1 nikdy nevypil žiaden z
alkoholov, ktoré vypil P2.

Relačný kalkul:

{[P1, P2]:

$\exists A$

$\text{lubi}(P1, A) \wedge /* P1 \text{ lubi nejaky alkohol, ktory } P2 \text{ vypil v aspon 3 krcmach */}$

(

$\exists K1 \exists K2 \exists K3$

$K1 \neq K2 \wedge K1 \neq K3 \wedge K2 \neq K3 \wedge$

$(\exists I \exists M \text{ navstivil}(I, P2, K1) \wedge \text{vypil}(I, A, M)) \wedge$

$(\exists I \exists M \text{ navstivil}(I, P2, K2) \wedge \text{vypil}(I, A, M)) \wedge$

$(\exists I \exists M \text{ navstivil}(I, P2, K3) \wedge \text{vypil}(I, A, M))$

) \wedge

$\neg /* P1 \text{ nevypil ziaden alkohol, ktory vypil } P2 */$

(

$\exists A2$

$(\exists I \exists K \exists M \text{ navstivil}(I, P1, K) \wedge \text{vypil}(I, A2, M)) \wedge$

$(\exists I \exists K \exists M \text{ navstivil}(I, P2, K) \wedge \text{vypil}(I, A2, M))$

)

}

Datalog:

```
answer(P1, P2) ←  
  lubi(P1, A),  
  navstivil(I1, P2, K1),  
  vypil(I1, A, _),  
  navstivil(I2, P2, K2),  
  vypil(I2, A, _),  
  navstivil(I3, P2, K3),  
  vypil(I3, A, _),  
  not K1 = K2,  
  not K1 = K3,  
  not K2 = K3,  
  not vypil_zle(P1, P2).
```

```
vypil_zle(P1, P2) ←  
  navstivil(I1, P1, _),  
  vypil(I1, A, _),  
  navstivil(I2, P2, _),  
  vypil(I2, A, _).
```

SQL:

```
create temporary table vypil_zle as  
select n1.Pijan as P1, n2.Pijan as P2  
from navstivil n1, vypil v1, navstivil n2, vypil v2  
where n1.Idn = v1.Idn and n2.Idn = v2.Idn and v1.Alkohol = v2.Alkohol
```

```
/* answer */
```

```
select l.Pijan as P1, n1.Pijan as P2  
from lubi l, navstivil n1, vypil v1, navstivil n2, vypil v2, navstivil n3, vypil v3  
where l.Alkohol = v1.Alkohol and n1.Idn = v1.Idn and n2.Idn = v2.Idn and  
n3.Idn = v3.Idn and n1.Pijan = n2.Pijan and n1.Pijan = n3.Pijan and  
v1.Alkohol = v2.Alkohol and v1.Alkohol = v3.Alkohol and  
n1.Krcma <> n2.Krcma and n1.Krcma <> n3.Krcma and n2.Krcma <> n3.Krcma and  
not exists (  
  select *  
  from vypil_zle vz  
  where l.Pijan = vz.P1 and n1.Pijan = vz.P2)
```

b) Nájdite dvojice [A, Total] ktoré hovoria o celkovom množstve alkoholu A, ktoré sa vypilo za vyššiu cenu než je priemerná cena alkoholu A. (Myslí sa priemerná jednotková cena cez krčmy, ktoré A čapujú.) Sformulujte tento dotaz v Datalogu. (2)

```
answer(A, Total) ←  
  subtotal(vypite_nad_priemer(_, A, M), [A], [Total = sum(M)]).
```

```
vypite_nad_priemer(I, A, M) ←  
  navstivil(I, _, K),  
  vypil(I, A, M),  
  capuje(K, A, C),  
  subtotal(capuje(_, A, C), [A], [Priemer = avg(C)]),  
  C > Priemer.
```

Tento program sa dá intuitívne optimalizovať tak, že subtotal v pravidle pre vypite_nad_priemer sa presunie do nového pravidla:

```
priemerna_cena(A, Priemer) ← subtotal(capuje(_, A, C), [A], [Priemer = avg(C)]).
```

Ideou tejto optimalizácie je zabrániť viacnásobnému výpočtu priemernej ceny pre ten istý alkohol A. Toto však nie je potrebné robiť, lebo dobrý optimalizátor sa o to postará automaticky. Ak sa chcete dozvedieť ako, zapíšte si kurz Relačné a logické databázové systémy, ktorý beží v letnom semestri.

2. Daná je relácia $r(A, B, C, D, E, F, G, H)$ s funkčnými závislosťami $BF \rightarrow ACG$, $C \rightarrow AE$, $AH \rightarrow F$, $AF \rightarrow H$, $BG \rightarrow F$, $E \rightarrow G$, $BCE \rightarrow F$, $GH \rightarrow AF$, $H \rightarrow D$.
a) Nájdite všetky kľúče relácie r . (2)

Atribút B nie je na pravej strane žiadnej funkčnej závislosti, teda musí byť v každom kľúči. Atribút D nie je na ľavej strane žiadnej funkčnej závislosti, teda nesmie byť v žiadnom kľúči.

ABCEFGH

-A: BCEFGH

-C: BEFGH

-E: BFGH

-F: BGH

-G: BH

+G: BG

+F: BF

+E: BE

+C: BC

+A: ABH

Kľúče v r sú ABH , BC , BE , BF , BG . Iné kľúče nie sú.

b) Nájdite minimálne pokrytie funkčných závislostí. (2)

Po minimalizácii ľavých strán funkčných závislostí:

$BF \rightarrow A$, $BF \rightarrow C$, $BF \rightarrow G$, $C \rightarrow A$, $C \rightarrow E$, $AH \rightarrow F$, $AF \rightarrow H$, $BG \rightarrow F$, $E \rightarrow G$, $BC \rightarrow F$,
 $GH \rightarrow A$, $GH \rightarrow F$, $H \rightarrow D$

Po odstránení redundantných funkčných závislostí (minimálne pokrytie):

$BF \rightarrow C$, $C \rightarrow A$, $C \rightarrow E$, $AH \rightarrow F$, $AF \rightarrow H$, $BG \rightarrow F$, $E \rightarrow G$, $GH \rightarrow A$, $H \rightarrow D$

Toto je jedno z minimálnych pokrytí. Iné minimálne pokrytia získame tak, že redundantné atribúty na ľavých stranách funkčných závislostí alebo celé redundantné funkčné závislosti odstránime v inom poradí.

c) Dekomponujte r do tretej normálnej formy, bezstratovo a so zachovaním všetkých funkčných závislostí. (2)

3NF dekompozícia z minimálneho pokrytia:

BCF

AC

CE

AFH

BGF

EG

AGH

DH

d) Rozhodnite a zdôvodnite, či rozklad $r_1(A, B, C, E, F, G, H)$, $r_2(D, H)$ je bezstratový (1),

Spoločný atribút r_1 a r_2 je H. Keďže H je nadkľúčom v r_2 , tento rozklad je bezstratový.

či je v tretej normálnej forme (1)

Relácia r_1 obsahuje len kľúčové atribúty, preto žiadna funkčná závislosť, ktorá v nej platí, nemôže porušiť 3NF. Relácia DH je binárna, preto je v 3NF. Takže tento rozklad je v tretej normálnej forme.

a či je v Boyce-Coddovej normálnej forme. (1)

V r_1 platí funkčná závislosť $E \rightarrow G$, pričom E nie je nadkľúčom v r_1 . Preto tento rozklad nie je v BCNF.

V r_1 platia aj iné závislosti, ktoré porušujú BCNF. Ale stačí nájsť jednu.

3. Daný je SQL dotaz nad databázou $r(X, Y)$, $s(X, Y)$, $t(X, Y)$:

```
select distinct r.X, count(s.Y) as S
```

```
from r, s
```

```
where r.Y = s.X and not exists (select * from s s2, t where t.X = 0 or s2.Y = 0)
```

```
group by r.X, s.X;
```

a) Zapište dotaz ekvivalentne v Datalogu. (3)

```
answer(RX, S) ←
```

```
  subtotal(rst(RX, _, SX, SY), [RX, SX], [S = count(SY)]).
```

```
rst(RX, RY, SX, SY) ←
```

```
  r(RX, RY),
```

```
  s(SX, SY),
```

```
  RY = SX,
```

```
  not st.
```

```
st ←
```

```
  s(, ),
```

```
  t(0, ).
```

```
st ←
```

```
  s(, 0),
```

```
  t(, ).
```

b) Vypočítajte výsledok dotazu (2) pre relácie

$r(X, Y) = s(X, Y) = \{[0, 0], [0, 1], [1, 0], [1, 1]\}$, $t(X, Y) = \emptyset$.

Pre danú databázu je relácia st prázdna (lebo t je prázdna), teda platí *not st*.

Vypočítajme reláciu rst :

```
rst(RX, RY, SX, SY) =
```

```
{[0,0,0,0], [0,0,0,1], [0,1,1,0], [0,1,1,1],
```

```
 [1,0,0,0], [1,0,0,1], [1,1,1,0], [1,1,1,1]}.
```

(Hodnoty grupovacích atribútov sú zvýraznené.)

Sú 4 grupy, $\{[RX, SX]\} = \{[0, 0], [0, 1], [1, 0], [1, 1]\}$.

Pre každú grupu je $\text{count}(SY) = 2$.

Výsledkom je množina $\{[X, S]\} = \{[0, 2], [1, 2]\}$. (V SQL vzniká po grupovaní, agregácii a následnom odprojektovaní grupovacieho atribútu SX multimnožina so 4 prvkami: $\{[0, 2], [0, 2], [1, 2], [1, 2]\}$. Finálny *distinct* tie duplikáty odstráni.)

4. Rozhodnite a zdôvodnite, či je rozvrh

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

a) obnoviteľný (1);

Rozvrh obsahuje dirty read $r_2(Z)$ (T2 číta nepotvrdené dáta od T1), ale T1 commituje skôr než T2.

Rozvrh obsahuje dirty read $r_3(X)$ (T3 číta nepotvrdené dáta od T2), ale T2 commituje skôr než T3.

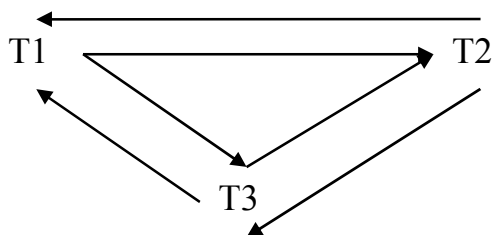
Iné dirty reads rozvrh neobsahuje, teda **rozvrh je obnoviteľný**.

b) vyhýbajúci sa kaskádovým abortom (1);

Tento rozvrh sa nevyhýba kaskádovým abortom, lebo obsahuje dirty read.

c) konflikt-sériovateľný (1)

Graf konfliktov:



Graf konfliktov je cyklický, preto **tento rozvrh nie je konflikt-sériovateľný**.