

11/1/2012 Úvod do databáz, **zápočtový** test, max 25 bodov, 90 min

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 trojice $[P, A1, A2]$ také, že pijan P ľúbi aspoň jeden z alkoholov A1 a A2 a pri každej svojej návšteve krčmy Feuerstein oba tie alkoholy (A1 aj A2) vypil.

Relačný kalkul:

$\{[P, A1, A2]:$

/* P lubi A1 alebo A2 */

$(\text{lubi}(P, A1) \vee \text{lubi}(P, A2)) \wedge$

/* A1 aj A2 su alkoholy */

$((\exists P1 \text{lubi}(P1, A1)) \vee (\exists K \exists C \text{capuje}(K, A1, C))) \wedge$

$((\exists P1 \text{lubi}(P1, A2)) \vee (\exists K \exists C \text{capuje}(K, A2, C))) \wedge$

\neg

/* pri niektorej navsteve feuerstein P nevypil A1 alebo nevypil A2 */

$(\exists I$

$\text{navstivil}(I, P, \text{feuerstein}) \wedge$

\neg

/* pri navsteve I pijan P vypil A1 aj A2 */

$(\exists M \text{vypil}(I, A1, M)) \wedge (\exists M \text{vypil}(I, A2, M))$

)

)

}

Datalog:

```
answer(P, A1, A2) ←  
    alkohol(A1), /* safety */  
    alkohol(A2), /* safety */  
    lubi(P, A1),  
    not niekedy_niektory_nevypil(P, A1, A2).
```

```
answer(P, A1, A2) ←  
    alkohol(A1),  
    alkohol(A2),  
  
    lubi(P, A2),  
    not niekedy_niektory_nevypil(P, A1, A2).
```

```
alkohol(A) ←  
    lubi(_, A).
```

```
alkohol(A) ←  
    capuje(_, A, _).
```

```
niekedy_niektory_nevypil(P, A1, A2) ←  
    navstivil(I, P, feuerstein),  
    alkohol(A1), /* safety */  
    alkohol(A2), /* safety */  
    not vypil_oba(I, A1, A2).
```

```
vypil_oba(I, A1, A2) ←  
    vypil(I, A1, _),  
    vypil(I, A2, _).
```

SQL:

```
create temporary table alkohol as
select l.Alkohol
from lubi l
union
select c.Alkohol
from capuje c
```

```
create temporary table vypil_obo as
select v1.Idn, v1.Alkohol as A1, v2.Alkohol as A2
from vypil v1, vypil v2
where v1.Idn = v2.Idn
```

```
create temporary table niekedy_niektory_nevypil as
select n.Pijan, a1.Alkohol as A1, a2.Alkohol as A2
from navstivil n, alkohol a1, alkohol a2
where n.Krcma = 'feuerstein' and not exists (
    select *
    from vypil_obo vo
    where n.Idn = vo.Idn and vo.A1 = a1.Alkohol and vo.A2 = a2.Alkohol)
```

```
/* main */
select l.Pijan, l.Alkohol
from lubi l, alkohol a1, alkohol a2
where (l.Alkohol = a1.Alkohol or l.Alkohol = a2.Alkohol) and not exists (
    select *
    from niekedy_niektory_nevypil nnn
    where nnn.Pijan = l.Pijan and nnn.A1 = a1.Alkohol and nnn.A2 = a2.Alkohol)
```

b) Nájdite dvojice [P, A] také, že pijan P vypil alkohol A vo väčšom celkovom množstve než ktorýkoľvek iný pijan; a zároveň P vypil alkohol A vo väčšom počte rôznych krčiem než ktorýkoľvek iný pijan. Sformulujte tento dotaz v Datalogu. (2)

```
answer(P, A) ←  
  navstivil(I, P, _), /* safety */  
  vypil(I, A, _), /* safety */  
  not iny_tolke_mnozstvo(P, A),  
  not iny_v_tolkych_krcmach(P, A).
```

```
iny_tolke_mnozstvo(P, A) ←  
  subtotal(nv_mnozstvo(_, P, A, M), [P, A], [T = sum(M)]),  
  subtotal(nv_mnozstvo(_, P2, A, M), [P2, A], [T2 = sum(M)]),  
  not P = P2,  
  T <= T2.
```

```
iny_v_tolkych_krcmach(P, A) ←  
  subtotal(nv_pocet(P, K, A), [P, A], [C = count(K)]),  
  subtotal(nv_pocet(P2, K, A), [P2, A], [C2 = count(K)]),  
  not P = P2,  
  C <= C2.
```

```
nv_mnozstvo(I, P, A, M) ←  
  navstivil(I, P, _),  
  vypil(I, A, M).
```

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

2. Daná je relácia $r(A, B, C, D, E, F, G, H)$ s funkčnými závislosťami
 $AH \rightarrow GE$, $AD \rightarrow E$, $B \rightarrow CEH$, $CD \rightarrow F$, $DE \rightarrow AB$

a) Nájdite všetky kľúče relácie r . (2)

Atribút D nie je na pravej strane žiadnej funkčnej závislosti, takže je v každom kľúči. F nie je na ľavej strane žiadnej funkčnej závislosti (a je na niektorej pravej), takže nie je v žiadnom kľúči.

AD je kľúč. BD je kľúč. DE je kľúč. Tým pádom atribúty A, B, E nie sú v žiadnom inom kľúči. Ostávajú atribúty C, D, G, H , lenže $CDGH$ nie je nadkľúč.

Kľúče sú AD, BD, DE . Iné kľúče nie sú.

b) Nájdite minimálne pokrytie funkčných závislostí. (2)

$AH \rightarrow G$, $AH \rightarrow E$, $AD \rightarrow E$, $B \rightarrow C$, $B \rightarrow E$, $B \rightarrow H$, $CD \rightarrow F$, $DE \rightarrow A$, $DE \rightarrow B$

Toto je minimálne pokrytie. Žiadna ľavá strana sa nedá zredukovať, ani žiadna funkčná závislosť nie je redundantná.

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

3NF dekompozícia z minimálneho pokrytia:

$AGH, AEH, ADE, BC, BH, CDF, BDE$

Alternatívna 3NF dekompozícia (A, B, D, E sú kľúčové atribúty):

$ABDE, AGH, AEH, BC, BH, CDF$

d) Dekomponujte r do Boyce-Coddovej normálnej formy, bezstratovo. Snažte sa vyhnúť zbytočnému rozbitiu funkčných závislostí. (2)

Začnime s 3NF dekompozíciou z minimálneho pokrytia:

$AGH, AEH, ADE, BC, BH, CDF, BDE$.

V relácii BDE platí $B \rightarrow E$, ale neplatí $B \rightarrow D$. Ostatné relácie sú v BCNF.

Dekompozícia do BCNF:

$AGH, AEH, ADE, BC, BH, CDF, BD, BE$

3. a) Rozhodnite a zdôvodnite (3), či sú programy P1 a P2 s dotazom „?- answer(Z)“ ekvivalentné (predikáty a, b tvoria extenzionálnu databázu spoločnú pre P1 aj P2):

P1: answer(X) ← a(X, X), not p(X, X).
 p(X, Y) ← a(X, Y), a(Y, X), not b(Y, X).

P2: answer(U) ← a(U, U), b(U, U).

Existuje test (Levy-Sagivov), ktorý ekvivalencie tohto typu rieši všeobecne. Lenže nie je potrebné poznať či vymýšľať všeobecný test. Stačí najskôr začať s úlohou b). Z riešenia úlohy b) vyplýva, že **P1 a P2 sú ekvivalentné**.

b) Zapište programy P1 a P2 s dotazom „?- answer(Z)“ v relačnom kalkule. (2)

P1:
 {Z:
 a(Z, Z) ∧
 ¬
 /* p(Z, Z) */
 (
 a(Z, Z) ∧ a(Z, Z) ∧ ¬ b(Z, Z)
)
 }

P2:
 {Z:
 a(Z, Z) ∧ b(Z, Z)
 }

Zjednodušte formulu pre P1:

$a(Z, Z) \wedge \neg (a(Z, Z) \wedge a(Z, Z) \wedge \neg b(Z, Z))$
 \equiv /* a(Z, Z) ∧ a(Z, Z) je ekvivalentné s a(Z, Z) */
 $a(Z, Z) \wedge \neg (a(Z, Z) \wedge \neg b(Z, Z))$
 \equiv /* de-Morganove pravidlá */
 $(a(Z, Z) \wedge \neg a(Z, Z)) \vee (a(Z, Z) \wedge b(Z, Z))$
 \equiv /* prvý disjunkt nie je TRUE pre žiadne Z */
 $a(Z, Z) \wedge b(Z, Z)$

Formula pre P1 je ekvivalentná formuli pre P2.

4. Uvažujte rozvrh $r_1(X)$, $w_1(X)$, $r_3(Y)$, $r_2(X)$, $r_1(Y)$, $w_2(Y)$, $w_3(X)$.

Vložte do rozvrhu commity všetkých troch transakcií (operácie c_1 , c_2 , c_3) tak, aby výsledný rozvrh bol

Po akomkoľvek doplnení commitov výsledný rozvrh nie je konflikt-sériovateľný kvôli dvojiciam operácií $r_3(Y)$, ..., $w_2(Y)$ a $r_2(X)$, ..., $w_3(X)$. Preto sa podmienky v úlohách b), c), d) nedajú splniť.

a) nie konflikt-sériovateľný a zároveň nie obnoviteľný; **(1)**

Keďže výsledný rozvrh nebude konflikt-sériovateľný, stačí splniť podmienku, aby nebol obnoviteľný. Napríklad takto:

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

T2 číta nepotvrdené data od T1 a T2 commituje skôr ako T1.

b) konflikt-sériovateľný, ale nie obnoviteľný; **(1)**

Taký rozvrh neexistuje.

c) konflikt-sériovateľný, obnoviteľný, ale nie vyhýbajúci sa kaskádovým abortom; **(1)**

Taký rozvrh neexistuje.

d) konflikt-sériovateľný, vyhýbajúci sa kaskádovým abortom, ale nie striktný. **(1)**

Taký rozvrh neexistuje.

V riešeniach zdôvodnite, prečo má rozvrh doplnený o commity požadovanú vlastnosť, resp. prečo sa commity nedajú doplniť tak, aby požadovanú vlastnosť mal.