

5/1/2012 Úvod do databáz, **zápočtový** test, max 25 bodov, 90 min

1. Daná je databáza: capuje(Krcma, Alkohol, Cena), lubi(Pijan, Alkohol)
navstivil(Idn, Pijan, Krcma), vypil(Idn, Alkohol, Mnozstvo).

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

a) Sformulujte nasledujúci dotaz v relačnom kalkule (2), Datalogu (2) a SQL (2):
Nájdite dvojice [P, K] o ktorých platí, že pijan P navštívil krčmu K práve trikrát,
pričom pri každej z tých troch návštev vypil 1 pivo a 1 borovičku a nič iné.

Relačný kalkul:

{[P, K]:

/* pijan P navstivil krcmu K prave trikrat (navstevy I1, I2, I3) */

$\exists I1 \exists I2 \exists I3$

$\text{navstivil}(I1, P, K) \wedge \text{navstivil}(I2, P, K) \wedge \text{navstivil}(I3, P, K) \wedge$

$I1 \neq I2 \wedge I1 \neq I3 \wedge I2 \neq I3 \wedge$

$\neg (\exists I \text{navstivil}(I, P, K) \wedge I \neq I1 \wedge I \neq I2 \wedge I \neq I3) \wedge$

/* pijan P vypil pri navstevach I1, I2, I3 jedno pivo a jednu borovicku */

$\text{vypil}(I1, \text{pivo}, 1) \wedge$

$\text{vypil}(I1, \text{borovicka}, 1) \wedge$

$\text{vypil}(I2, \text{pivo}, 1) \wedge$

$\text{vypil}(I2, \text{borovicka}, 1) \wedge$

$\text{vypil}(I3, \text{pivo}, 1) \wedge$

$\text{vypil}(I3, \text{borovicka}, 1) \wedge$

/* pijan P nevypil pri navstevach I1, I2, I3 nic okrem piva a borovicky */

$\neg (\exists A \exists M \text{vypil}(I1, A, M) \wedge A \neq \text{pivo} \wedge A \neq \text{borovicka}) \wedge$

$\neg (\exists A \exists M \text{vypil}(I2, A, M) \wedge A \neq \text{pivo} \wedge A \neq \text{borovicka}) \wedge$

$\neg (\exists A \exists M \text{vypil}(I3, A, M) \wedge A \neq \text{pivo} \wedge A \neq \text{borovicka})$

}

Datalog:

```
answer(P, K) ←  
  navstivil(I1, P, K),  
  navstivil(I2, P, K),  
  navstivil(I3, P, K),  
  not I1 = I2,  
  not I1 = I3,  
  not I2 = I3,  
  not navstivil_inokedy(P, K, I1, I2, I3),  
  vypil(I1, pivo, 1),  
  vypil(I1, borovicka, 1),  
  not vypil_ine(I1),  
  vypil(I2, pivo, 1),  
  vypil(I2, borovicka, 1),  
  not vypil_ine(I2),  
  vypil(I3, pivo, 1),  
  vypil(I3, borovicka, 1),  
  not vypil_ine(I3).
```

```
navstivil_inokedy(P, K, I1, I2, I3) ←  
  navstivil(I1, _, _), /* safety */  
  navstivil(I2, _, _), /* safety */  
  navstivil(I3, _, _), /* safety */  
  navstivil(I, P, K),  
  not I = I1,  
  not I = I2,  
  not I = I3.
```

```
vypil_ine(I) ←  
  vypil(I, A, _),  
  not A = pivo,  
  not A = borovicka.
```

SQL:

```
create temporary table navstivil_inokedy as
select n.Pijan, n.Krcma, n1.Idn as I1, n2.Idn as I2, n3.Idn as I3
from navstivil n1, navstivil n2, navstivil n3, navstivil n
where n.Idn <> n1.Idn and n.Idn <> n2.Idn and n.Idn <> n3.Idn
```

```
create temporary table vypil_ine as
select v.Idn
from vypil v
where v.Alkohol <> pivo and v.Alkohol <> borovicka
```

```
/* main */
select
from navstivil n1, navstivil n2, navstivil n3, vypil v1p, vypil v2p, vypil v3p,
vypil v1b, v2b, v3b
where n1.Pijan = n2.Pijan and n1.Pijan = n3.Pijan and
n1.Krcma = n2.Krcma and n1.Krcma = n3.Krcma and
n1.Idn <> n2.Idn and n1.Idn <> n3.Idn and n2.Idn <> n3.Idn and
n1.Idn = v1p.Idn and v1p.Alkohol = ,pivo‘ and v1p.Mnozstvo = 1 and
n1.Idn = v1b.Idn and v1b.Alkohol = ,borovicka‘ and v1b.Mnozstvo = 1 and
n2.Idn = v2p.Idn and v2p.Alkohol = ,pivo‘ and v2p.Mnozstvo = 1 and
n2.Idn = v2b.Idn and v2b.Alkohol = ,borovicka‘ and v2b.Mnozstvo = 1 and
n3.Idn = v3p.Idn and v3p.Alkohol = ,pivo‘ and v3p.Mnozstvo = 1 and
n3.Idn = v3b.Idn and v3b.Alkohol = ,borovicka‘ and v3b.Mnozstvo = 1 and
not exists (
select *
from navstivil_inokedy ni
where ni.Pijan = n1.Pijan and ni.Krcma = n1.Krcma and
ni.I1 = n1.Idn and ni.I2 = n2.Idn and ni.I3 = n3.Idn) and
not exists (
select * from vypil_ine vi
where vi.Idn = n1.Idn) and
not exists (
select * from vypil_ine vi
where vi.Idn = n2.Idn) and
not exists (
select * from vypil_ine vi
where vi.Idn = n3.Idn)
```

b) Nájdite trojice [K, A, Pocet] ktoré hovoria o počte (rôznych) pijanov, ktorí aspoň raz navštívili krčmu K, ľúbia alkohol A a vypili ho pri každej návšteve krčmy K. Vo výsledku nemajú byť trojice, kde Pocet = 0. Sformulujte tento dotaz v Datalogu. (2)

```
answer(K, A, Pocet) ←  
    subtotal(data(P, K, A), [K, A], [Pocet = count(P)]).
```

```
data(P, K, A) ←  
    lubi(P, A),  
    navstivil(_, P, K),  
    not niekedy_nevypil(P, K, A).
```

```
niekedy_nevypil(P, K, A) ←  
    /* safety: lubi(P, A) */  
    navstivil(I, P, K),  
    not vypil(I, A, _).
```

2. Daná je relácia $r(A, B, C, D, E, F, G, H)$ s funkčnými závislosťami
 $ABH \rightarrow G, AC \rightarrow E, AD \rightarrow F, AF \rightarrow BH, BG \rightarrow CF, EF \rightarrow AG, G \rightarrow F, GH \rightarrow AF, H \rightarrow D$
 a) Nájdite všetky kľúče relácie r . (2)

Skúsme uhádnuť nejaké krátke kľúče. Jednoatribútové kľúče nie sú. $AD, AF, AG, AH, EF, EG, GH$ sú všetky dvoatribútové kľúče. Dlhšie kľúče preskúmame kompletným prehľadávaním (keďže hľadáme aspoň 3-atribútové kľúče, prehľadávanie sa dá značne urýchliť úvahami typu „ak je v kľúči A, tak nie je v kľúči D, F, G, ani H“).

ABCDEFGH

-A: BCDEFGH

-E: BCDFGH

-G: BCDFH

+G: BCDFG

+E: BCDEH

+A: ABCE

Kľúče sú AD, AF, AG, AH, EF, EG, GH. Iné kľúče nie sú.

b) Nájdite minimálne pokrytie funkčných závislostí. (2)

Po rozbití pravých strán funkčných závislostí:

$ABH \rightarrow G, AC \rightarrow E, AD \rightarrow F, AF \rightarrow B, AF \rightarrow H, BG \rightarrow C, BG \rightarrow F, EF \rightarrow A, EF \rightarrow G, G \rightarrow F,$
 $GH \rightarrow A, GH \rightarrow F, H \rightarrow D$

Po minimalizácii ľavých strán funkčných závislostí:

$AH \rightarrow G, AC \rightarrow E, AD \rightarrow F, AF \rightarrow B, AF \rightarrow H, BG \rightarrow C, BG \rightarrow F, EF \rightarrow A, EF \rightarrow G, G \rightarrow F,$
 $GH \rightarrow A, GH \rightarrow F, H \rightarrow D$

Po odstránení redundantných funkčných závislostí získame (niektoré) minimálne pokrytie:

$AH \rightarrow G, AC \rightarrow E, AD \rightarrow F, AF \rightarrow B, AF \rightarrow H, BG \rightarrow C, EF \rightarrow A, G \rightarrow F, GH \rightarrow A, H \rightarrow D$

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

3NF dekompozícia z minimálneho pokrytia:

ACE

ABF

ADF

AEF

AFH

AGH

BCG

DH

FG

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

V horeuvedenej 3NF dekompozícii potrebujeme skúmať len relácie s 3 atribútmi. V r platia len dve netriviálne závislosti s jedným atribútom na ľavej strane (len také môžu ohroziť BCNF v ternárnych reláciách): $G \rightarrow F$, $H \rightarrow D$. Keďže žiadna z týchto závislostí neplatí v žiadnej ternárnej relácii horeuvedenej 3NF dekompozície, **tá 3NF dekompozícia je zároveň v BCNF.**

3. Daná je EDB relácia $a(X, Y)$ a datalogový program

$f(X, Y) \leftarrow a(X, _), a(Y, _), \text{not } g(X, Y).$

$g(X, Y) \leftarrow a(X, Z), a(Y, _), \text{not } a(Y, Z).$

$g(Y, X) \leftarrow a(X, Z), a(Y, _), \text{not } a(Y, Z).$

a) Zapište dotaz „? $f(X, Y)$ “ ekvivalentne v relačnom kalkule. (2)

$\{[X, Y]: (\exists W a(X, W)) \wedge (\exists W a(Y, W)) \wedge \neg$
 $((\exists Z \exists W a(X, Z) \wedge a(Y, W) \wedge \neg a(Y, Z)) \vee (\exists Z \exists W a(Y, Z) \wedge a(X, W) \wedge \neg a(X, Z)))\}$

To je ekvivalentné

$\{[X, Y]: (\exists W a(X, W)) \wedge (\exists W a(Y, W)) \wedge$
 $\neg (\exists Z \exists W a(X, Z) \wedge a(Y, W) \wedge \neg a(Y, Z)) \wedge$
 $\neg (\exists Z \exists W a(Y, Z) \wedge a(X, W) \wedge \neg a(X, Z))\}$

b) Zapište dotaz „? $f(X, Y)$ “ ekvivalentne v SQL. (2)

create temporary table g as

(select a1.X, a2.X as Y

from a a1, a a2

where not exists (select * from a a3 where a3.X = a2.X and a3.Y = a1.Y))

union

((select a2.X as Y, a1.X

from a a1, a a2

where not exists (select * from a a3 where a3.X = a2.X and a3.Y = a1.Y))

select a1.X, a2.X as Y

from a a1, a a2

where not exists (select * from g g1 where g1.X = a1.X and g1.Y = a2.X)

c) Vypočítajte výsledok dotazu „? f(X, Y)“ pre reláciu
 $a(X, Y) = \{(1,1), (1,4), (2,1), (2,3), (2,4), (3,1), (4,1), (4,4)\}$. (2)

Môžeme vypočítať najskôr

$g(X, Y) = \{[1, 2], [1, 3], [2, 1], [2, 3], [2, 4], [3, 1], [3, 2], [3, 4], [4, 2], [4, 3]\}$.

Ako? Pre človeka je snád' najjednoduchšie všimnúť si, že datalogové pravidlá pre predikát g sa líšia len v hlave, kde sú prehodené premenné X a Y. Pravá strana je rovnaká. Prvé pravidlo neformálne hovorí: "Ak na niektorých prvých miestach v a (to môže byť aj ten istý záznam) sú hodnoty X a Y, pričom hodnotu Z pri X nevieš nikde v a nájsť pri Y, tak do g pridaj dvojicu [X, Y]." Dokopy tie dve pravidlá hovoria: "Ak na niektorých prvých miestach v a sú hodnoty X a Y, pričom hodnotu Z pri X nevieš nikde v a nájsť pri Y, tak do g pridaj dvojice [X, Y] aj [Y, X]."

Potom vypočítame $f(X, Y) = \{[1, 1], [1, 4], [2, 2], [3, 3], [4, 1], [4, 4]\}$.

Ako? Datalogové pravidlo pre f neformálne hovorí: "Ak na niektorých prvých miestach v a sú hodnoty X a Y, pričom dvojica [X, Y] nie je v g, tak [X, Y] pridaj do f."

4. Uved'te príklad rozvrhu, v ktorom všetky transakcie končia commitom, a ktorý je:
a) konflikt-sériovateľný, obnoviteľný a nevyhýbajúci sa kaskádovým abortom; **(1)**

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

b) striktný a zároveň nie konflikt-sériovateľný; **(1)**

$r1(X), w2(X), r2(Y), w1(Y), c1, c2$

c) konflikt-sériovateľný, vyhýbajúci sa kaskádovým abortom a zároveň taký, že sa nedá generovať schedulerom, ktorý používa dvojfázové zamykanie. **(1)**

$r1(X), r2(X), w1(X), c1, c2$