

30/1/2019 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) Nájdite dvojice $[K, A]$ také, že krčma K čapuje alkohol A ; a zároveň každý pijan, ktorý navštívil K a ľúbi A , vypil A pri niektorej návšteve krčmy K . Sformulujte tento dotaz v relačnom kalkule (6) a Datalogu (6).

Relačný kalkul:

```
{[K, A]:
  capuje(K, A) ^
  ¬ /* nespravny_pijan(K, A) */
  (
    ∃P
    (∃I navstivil(I, P, K)) ^ lubi(P, A) ^
    ¬ /* niekedy_vypil(P, K, A) */
    (
      ∃I ∃M
      navstivil(I, P, K) ^ vypil(I, A, M)
    )
  )
}
```

Datalog:

```
answer(K, A) ←
  capuje(K, A),
  not nespravny_pijan(K, A).
```

```
nespravny_pijan(K, A) ←
  navstivil(_, P, K),
  lubi(P, A),
  not niekedy_vypil(P, K, A).
```

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

b) Nájdiť krémy, ktoré čapujú alkohol, ktorý ľúbi aspoň 80% návštevníkov tej krčmy a ktorý sa vypil počas aspoň 70% návštev tej krčmy. Sformulujte tento dotaz v Datalogu (6) a SQL (6).

Datalog:

```
answer(K) ←
  capuje(K, A),
  subtotal(lp(K, A, P), [K, A], [CLP = count(P)]),
  subtotal(p(K, P), [K], [CP = count(P)]),
  subtotal(nv(K, A, I), [K, A], [CNV = count(I)]),
  subtotal(n(K, I), [K], [CN = count(I)]),
  CLP >= 0.8 * CP,
  CNV >= 0.7 * CN.
```

```
lp(K, A, P) ←
  navstivil(_, P, K),
  lubi(P, A).
```

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

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

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

SQL:

```
with
agglp as (
  select n.Krcma, l.Alkohol, count(distinct n.Pijan) as CLP
  from navstivil n, lubi l
  where n.Pijan = l.Pijan
  group by n.Krcma, l.Alkohol
),
aggp as (
  select n.Krcma, count(distinct n.Pijan) as CP
  from navstivil n
  group by n.Krcma
),
aggnv as (
  select n.Krcma, v.Alkohol, count(n.Idn) as CNV
  from navstivil n, vypil v
  where n.Idn = v.Idn
  group by n.Krcma, v.Alkohol
),
aggn as (
  select n.Krcma, count(n.Idn) as CN
  from navstivil n
  group by n.Krcma
)
select c.Krcma
from capuje c, agglp alp, aggp ap, aggnv anv, aggn an
where c.Krcma = alp.Krcma and c.Krcma = ap.Krcma and c.Krcma = anv.Krcma and c.Krcma = an.Krcma and
alp.CLP >= 0.8 * ap.CP and anv.CNV >= 0.7 * CN
```

2. Daná je relácia $r(A, B, C, D, E, F, G)$ s funkčnými závislosťami
 $ABCD \rightarrow EF$; $ABE \rightarrow FG$; $ABDG \rightarrow CF$; $G \rightarrow BD$.

a) Nájdite nejaké minimálne pokrytie danej množiny funkčných závislostí. (6)

$ABCD \rightarrow E, ABE \rightarrow G, AG \rightarrow C, AG \rightarrow F, G \rightarrow B, G \rightarrow D$

b) Nájdite všetky kľúče r . (6)

Atribút A musí byť v každom kľúči, lebo nie je na pravej strane žiadnej funkčnej závislosti minimálneho pokrytia.

ABCDEFG
 -B: ACDEFG
 -C: ADEFG
 -D: AEEFG
 -E: AFG
 -F: AG
 +F: AF
 +E: AEF
 +D: ADEF
 +C: ACDEF
 +B: ABCDEF
 -C: ABDEF
 -D: ABEF
 -E: ABF
 +E: ABE
 +D: ABDF
 +C: ABCDF
 -D: ABCF
 +D: ABCD

Kľúče r sú ABCD, ABE, AG. Iné kľúče nie sú.

c) Rozhodnite či existuje bezstratová dekompozícia r do dvoch relácií, ktorá je v tretej normálnej forme a zachováva všetky funkčné závislosti danej relačnej schémy. Ak áno, uveďte takú dekompozíciu a zdôvodnite prečo spĺňa všetky tieto požiadavky. Ak nie, zdôvodnite prečo neexistuje. (6)

Definícia: Relačná schéma (r, U) je v 3NF, ak pre každú netriviálnu funkčnú závislosť $X \rightarrow Y$ z U^+ platí, že X je nadkľúč r , alebo Y patrí do niektorého kľúča.

Podme nájsť rozklad r do r_1 a r_2 s požadovanými vlastnosťami. Jediný atribút, ktorý nepatrí do žiadneho kľúča, je F. Teda všetky ostatné atribúty môžeme dať spolu do jednej relácie bez porušenia 3NF: $r_1(A, B, C, D, E, G)$. r_2 musí obsahovať F. Keďže vyžadujeme zachovanie všetkých funkčných závislostí, musia byť kvôli $AG \rightarrow F$ v r_2 tiež atribúty A, G: $r_2(A, F, G)$.

Áno, taká dekompozícia existuje, napr. $r_1(A, B, C, D, E, G)$, $r_2(A, F, G)$. Keďže $r_1 \cap r_2 = \{A, G\} \rightarrow \{F\}$, táto dekompozícia je bezstratová. V r_2 je zachovaná funkčná závislosť $AG \rightarrow F$, v r_1 sú zachované všetky ostatné funkčné závislosti minimálneho pokrytia.

3. Uvažujte relácie $r(X, Y, Z)$, $s(Z)$ bez duplikátov a NULL hodnôt. Relácia d sa počíta takto:

$$d = \Delta(\Pi_{X,Y}(r)) - \Pi_{X,Y}(\Delta(\Pi_{X,Y}(r) \times s) - r).$$

a) Zapište reláciu d v relačnom kalkule (relácie interpretujte ako predikáty). (6)

Predikát d je pravdivý pre usporiadané dvojice $[X, Y]$ z množiny

$\{[X, Y]\}$:

$$\begin{aligned} & \exists Z1 \\ & r(X, Y, Z1) \wedge \\ & \neg \\ & (\\ & \quad \exists Z2 \exists Z3 \\ & \quad r(X, Y, Z2) \wedge s(Z3) \wedge \\ & \quad \neg \\ & \quad r(X, Y, Z3) \\ &) \\ &) \end{aligned}$$

Definícia $d(., .)$ v relačnom kalkule bez použitia množín:

$\forall X \forall Y$

$$\begin{aligned} & (\\ & \quad \exists Z1 \\ & \quad r(X, Y, Z1) \wedge \\ & \quad \neg \\ & \quad (\\ & \quad \quad \exists Z2 \exists Z3 \\ & \quad \quad r(X, Y, Z2) \wedge s(Z3) \wedge \\ & \quad \quad \neg \\ & \quad \quad r(X, Y, Z3) \\ & \quad) \\ &) \\ & \Rightarrow d(X, Y) \end{aligned}$$

b) Vypočítajte reláciu d pre databázu

$$r(X, Y, Z) = \{[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 0, 1], [1, 1, 1]\},$$

$$s(Z) = \{[0], [1]\}. \quad (6)$$

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

$$\Delta(\Pi_{X,Y}(r)) = .(X, Y) = \{[0, 0], [0, 1], [1, 0], [1, 1]\}$$

$$\Delta(\Pi_{X,Y}(r) \times s) = .(X, Y, Z) = \{[0, 0, 0], [0, 1, 0], [1, 0, 0], [1, 1, 0], [0, 0, 1], [0, 1, 1], [1, 0, 1], [1, 1, 1]\}$$

$$\Delta(\Pi_{X,Y}(r) \times s) - r = .(X, Y, Z) = \{[1, 1, 0], [0, 1, 1]\}$$

$$\Pi_{X,Y}(\Delta(\Pi_{X,Y}(r) \times s) - r) = .(X, Y) = \{[1, 1], [0, 1]\}$$

$$\Delta(\Pi_{X,Y}(r)) - \Pi_{X,Y}(\Delta(\Pi_{X,Y}(r) \times s) - r) = d(X, Y) = \{[0, 0], [1, 0]\}$$

4. Uvažujte SQL dotaz *select r.X, s.X from r, s where r.X = s.X and r.Y > s.Y*. Relácia *r* je uložená v 1000 diskových blokoch, relácia *s* je uložená v 500 diskových blokoch. Bloky na disku a v operačnej pamäti sú rovnako veľké. Relácie *r, s* neobsahujú duplikáty ani null hodnoty. Dotaz sa počíta metódou nested-loop-join, k dispozícii je 200 voľných blokov v operačnej pamäti.

Vysvetlite spôsob použitia voľných blokov v operačnej pamäti pri výpočte dotazu, uveďte počet vstupných diskových operácií. (6)

198 blokov RAM sa použije pre vstup blokov relácie *s* (lebo je menšia než *r*).

1 blok RAM sa rezervuje pre vstup blokov relácie *r*.

1 blok RAM sa rezervuje pre výstup blokov joinu.

Priebeh výpočtu:

1. Prečíta sa 198 blokov *s* a potom postupne všetkých 1000 blokov *r* (pre každý prečítaný blok *r* sa vypočítajú výstupné *n*-tice).

2. Prečíta sa ďalších 198 blokov *s* a potom postupne všetkých 1000 blokov *r* (pre každý prečítaný blok *r* sa vypočítajú výstupné *n*-tice).

3. Prečíta sa zvyšných 104 blokov *s* a potom postupne všetkých 1000 blokov *r* (pre každý prečítaný blok *r* sa vypočítajú výstupné *n*-tice).

Dokopy sa urobí $198 + 1000 + 198 + 1000 + 104 + 1000 = 3500$ vstupných diskových operácií.

Ak na začiatku krokov 2 a 3 „recyklujeme“ naposledy prečítaný blok *r*, ktorý je vtedy uložený v RAM, vieme počet vstupných operácií zredukovať na 3498.