

I. Daná je databáza

dodava(firma, vyrobok, cena, lehota)	$D(f,v,c,l)$
firmy(firma, mesto)	$F(f,m)$
objednavky(klient, vyrobok)	$O(k,v)$

(Jedna firma môže dodávať ten istý výrobok za rôzne ceny v rôznych dodacích lehotách.)

1) Vypíšte žilinské firmy, ktoré dodávajú všetky výrobky, ktoré dodáva firma CRN. Sformulujte tento dotaz v relačnom kalkule (2), Datalogu (2) a SQL (2).

$\{f: F(f, \text{Žilina}) \wedge (\forall (v, x_1, y_1) D(\text{CRN}, v, x_1, y_1) \Rightarrow \exists (x_2, y_2) D(f, v, x_2, y_2))\}$

$NDV(f) \leftarrow F(f, \text{Žilina}), D(\text{CRN}, v, x_1, y_1), \neg D(f, v, x_2, y_2).$

$Answer(f) \leftarrow F(f, \text{Žilina}), \neg NDV(f).$

*select F.firma from Firmy F*

*where F.mesto = "Žilina" and not exists*

*(select D1.vyrobok from Dodava D1*

*where D1.Firma = "CRN" and D1.vyrobok not in*

*(select D2.vyrobok from Dodava D2 where D2.firma = F.firma));*

2) Pre každý výrobok, ktorý objednal klient Fero, vypíšte minimálnu cenu, za ktorú sa ten výrobok dodáva; a počet firiem, ktoré ten výrobok dodávajú lacnejšie ako za 100 Sk. Sformulujte tento dotaz v SQL (3).

*select O.vyrobok, min(D.cena), count(distinct D.firma) as Pocet\_firiem  
from Objednavk O, Dodava D*

*where O.klient = "Fero" and D.vyrobok = O.vyrobok and D.cena < 100  
group by O.vyrobok*

*union*

*select O.vyrobok, min(D.cena), 0 as Pocet\_firiem*

*from Objednavk O, Dodava D*

*where O.klient = "Fero" and D.vyrobok = O.vyrobok*

*group by O.vyrobok*

*having min(D.cena) >= 100;*

3) Vypíšte žilinské firmy, ktoré dodávajú do 2 dní všetky výrobky, ktoré si objednal klient Fero. Sformulujte tento dotaz v relačnom kalkule (2), Datalogu (2) a SQL (2).

$\{f: F(f, \text{Žilina}) \wedge (\forall (v) O(\text{Fero}, v) \Rightarrow \exists (x_1, y_1) D(f, v, x_1, y_1) \wedge y_1 \leq 2 \text{ dni})\}$

$DZ2d(f, v) \leftarrow F(f, \text{Žilina}), D(f, v, x_1, y_1), y_1 \leq 2 \text{ dni}$

$NDV(f) \leftarrow F(f, \text{Žilina}), O(\text{Fero}, v), \neg DZ2d(f, v).$

$Answer(f) \leftarrow F(f, \text{Žilina}), \neg NDV(f).$

*select F.firma from Firmy F*

*where F.mesto = "Žilina" and not exists*

*(select O.vyrobok from Objednavky O*

*where O.klient = "Fero" and O.vyrobok not in*

*(select D.vyrobok from Dodava D  
where D.firma = F.firma and D.lehota ≤ 2 dni ));*

4) Pre každý výrobok, ktorý dodáva firma CLR, vypíšte počet bratislavských firiem, ktoré dodávajú ten výrobok lacnejšie ako CLR; a tiež vypíšte minimálnu cenu pre ten výrobok (spomedzi bratislavských firiem). Sformulujte tento dotaz v SQL. (3)

```
select D1.vyrobok, min(D2.cena), count(distinct F.firma) as Pocet_firiem
from Firmy F, Dodava D1 D2 /*Dodava D1, Dodava D2*/
where D1.firma = "CLR" and F.mesto="Bratislava" and D2.firma = F.firma
and D1.vyrobok = D2.vyrobok and D2.cena < D1.cena
group by D1.vyrobok
union
select D1.vyrobok, min(D.cena), 0 as Pocet_firiem
from Firmy F, Dodava D1 D2,
where D1.firma = "CLR" and F.mesto="Bratislava" and D2.firma = F.firma
and D1.vyrobok = D2.vyrobok and all D2.cena >= D1.cena
group by D1.vyrobok;
```

5) Vypíšte klientov, z ktorých každý si objednal iba tie výrobky, ktoré dodávajú firmy zo Zvolena. Sformulujte tento dotaz v relačnom kalkule (2), Datalogu (2) a SQL (2).

$\{k: (\forall v)(O(k,v) \Rightarrow (\exists f)(\exists c)(\exists l)(F(f, Zvolen) \wedge D(f,v,c,l)))\}$  alebo  
 $\{k: \neg((\exists v)O(k,v) \wedge \neg(\exists f)(\exists c)(\exists l)(F(f, Zvolen) \wedge D(f,v,c,l)))\}$  alebo  
 $\{k: (\forall v)(\exists f)(\exists c)(\exists l)(O(k,v) \wedge F(f, Zvolen) \wedge D(f,v,c,l))\}$  Nie je to ekvivalentný dotaz, nezahrnuje klientov, čo si neobjednali nič.

AllV(v) ← O(x,v).

AllV(v) ← D(y,v,z,t). /\* všetky výrobky \*/

DZF(v) ← F(f, Zvolen), D(f,v,c,l).

/\* výrobky dodávané aj firmami zo Zvolena \*/

VNZF(v) ← AllV(v), ¬ DZF(v).

/\* výrobky, ktoré nedodáva žiadna firma zo Zvolena \*/

Answer(k) ← O(k,v), ¬ VNZF(v).

**select O.k from Objednavky O**

**where all O.v in**

**(select D.v from Firmy F Dodava D**

**where (F.mesto = "Zvolen") and (D.firma = F.firma)); alebo bez all**

**select O.k from Objednavky O1**

**where not exists select O2.v from Objednavky O2**

**where (O2.klient = O.klient) and (O2.vyrobok not in**

**(select D.v from Firmy F Dodava D**

**where (F.mesto = "Zvolen") and (D.firma = F.firma));**

6) Koľko výrobkov, z ktorých každý je dodávaný aspoň 5 rôznymi dodávateľmi, si objednali (zároveň) obaja klienti Hogo aj Fogo? Sformulujte tento dotaz v SQL. (3)

```
select count(vyrobok) from Dodava D, Objednavky O1 O2
where O1.klient = "Hogo" and O2.klient="Fogo" and
    O1.vyrobok = O2.vyrobok and D.vyrobok = O1.vyrobok
having count(distinct firma) >= 5;
```

7) Sformulujte nasledujúci dotaz v relačnom kalkule (2), relačnej algebre (2), Datalogu (2) a SQL (2): Nájdite dvojice [firma, počet], ktoré hovoria, koľko výrobkov z objednávky klienta Lojzo vie tá firma dodať do 14 dní. Pritom treba nájsť len tie firmy, ktoré Lojzovi vedia dodať aspoň 6 výrobkov.

$$\{ \langle F, P \rangle \mid \neg \exists (P = \text{count}(V)) (\exists (C, L) (d(F, V, C, L) \wedge o(\text{lojzo}, V) \wedge L \leq 14)) \wedge P \geq 6 \}$$

$$\sigma_{P \geq 6}(\Gamma_{F, P = \text{count}(V)}(\prod_{F, V} ((\sigma_{L \leq 14} d) \bowtie (\sigma_{K = \text{lojzo}} o))))$$

$$h(F, V) \leftarrow d(F, V, C, L), o(\text{lojzo}, V), L \leq 14.$$

$$a(F, P) \leftarrow \text{Subtotal}(h(F, V); F; P = \text{count}(V)), P \geq 6.$$

(alternatívna syntax pre subtotal  $a(F, V) \leftarrow \text{Subtotal}(h, \text{group}, \text{count}, F, V), V \geq 6.$ )

```
select d.firma as F, count(distinct d.vyrobok) as P from dodava d, objednávky o
where d.lehota ≤ 14 and o.klient = "Lojzo" and o.vyrobok = d.vyrobok
group by d.firma having count(distinct d.vyrobok) ≥ 6;
```

8) Zapište nasledujúci dotaz v slovenskom jazyku (čo najpresnejšie) (1), v SQL (2) a v Datalogu (2):

$$\{ [F1, F2] : \exists M1 \text{ firma}(F1, M1) \wedge \exists M2 \text{ firma}(F2, M2) \wedge \\ \forall V (\text{objednavky}(\text{dezider}, V) \Rightarrow (\exists C1 \exists L1 \exists C2 \exists L2 \text{ dodava}(F1, V, C1, L1) \\ \wedge \text{dodava}(F2, V, C2, L2))) \}$$

Napište všetky dvojice firiem takých, že dodávajú všetky výrobky, čo si objednal Dezidér!

$$\text{ndv}(F) \leftarrow f(F, \_), o(\text{dezider}, V), \neg d(F, V, \_, \_).$$

$$\text{dv}(F) \leftarrow f(F, \_), \neg \text{ndv}(F).$$

$$a(F1, F2) \leftarrow \text{dv}(F1, F2).$$

```
select f1.firma, f2.firma from firmy f1, firmy f2
where f1.firma not in (select f.firma from firmy f, objednávky o
    where o.klient = "Dezider" and o.vyrobok not in
        (select d.vyrobok from dodava d where d.firma = f.firma)) and
    f2.firma not in (select f.firma from firmy f, objednávky o
    where o.klient = "Dezider" and o.vyrobok not in
        (select d.vyrobok from dodava d where d.firma = f.firma));
```

9) Sformulujte nasledujúci dotaz v relačnom kalkule (2), relačnej algebre (2), Datalogu (2) a SQL (2): Nájdite klientov, ktorí si objednali výrobky dodávané

firmami z Košíc alebo z Prešova a ku každému klientovi počet takých výrobkov. (Stačí, aby si každý z klientov objednal jeden výrobok, ktorý vie dodať aspoň jedna firma, ktorá je z Košíc alebo z Prešova.) Vypísať len tých klientov, ktorí si objednali aspoň 10 takých výrobkov

$\{\langle k, p \rangle : (\bigvee (p = \text{count}(v)) (\exists (f, m, x_1, x_2) (O(k, v) \wedge D(f, v, x_1, x_2) \wedge F(f, m) \wedge (m = \text{Košice} \vee m = \text{Prešov})))) \wedge (p \geq 10)\}$   
 $\sigma_{p \geq 10}(\Gamma_{k, p \leftarrow \text{count}(v)}(\Pi_{k, v}(O \bowtie D \bowtie (\sigma_{m = \text{Košice} \vee m = \text{Prešov}} F))))$   
 $DKP(k, v) \leftarrow O(k, v), D(f, v, x_1, x_2), F(f, m), (m = \text{Košice} \vee m = \text{Prešov}).$   
 $\text{Answer}(k, p) \leftarrow \text{Subtotal}(DKP(k, v); k; p = \text{count}(v)), p \geq 10.$

**select** O.klient, count(O.vyrobok) **as** pocet **from** Objednavky O, Dodava D firmy F **where** O.vyrobok = D.vyrobok **and** D.firma = F.firma **and** F.mesto **in** ("Košice", "Prešov")  
**group by** O.klient **having** count(O.vyrobok) >= 10;

10) Zapište nasledujúci dotaz v slovenskom jazyku (čo najpresnejšie) (1), v SQL (2) a v Datalogu (2):

$\{F: \forall V \forall F1 \forall C1 \forall L1 \exists C2 \exists L2 ((\text{dodava}(F1, V, C1, L1) \wedge \text{firmy}(F1, \text{bratislava})) \Rightarrow (\text{dodava}(F, V, C2, L2) \wedge (C2 < C1 \vee L2 < L1)))\}$

Všetky firmy, ktoré dodávajú všetky tovary dodávané Bratislavskými firmami a navyše každý tovar dokážu dodať buď lacnejšie alebo rýchlejšie ako akákoľvek firma z Bratislavy.

**select** F.firma **from** Firmy F  
**where not exists** (**select** D1.vyrobok **from** Dodava D1, Firmy F1  
**where** F1.firma=D1.firma **and** F.mesto = "Bratislava" **and** D1.vyrobok **not in** (**select** D2.vyrobok **from** Dodava D2 **where** D2.firma=F1.firma **and** (D2.cena < D1.cena **or** D2.lehota < D1.lehota));

$Ba(f, v, c, l) \leftarrow \text{Dodava}(f, v, c, l), \text{Firmy}(f, \text{Bratislava}).$   
 $Vba(v, c, l) \leftarrow \text{Subtotal}(Ba(f, v, c, l); v; c = \min(c), l = \min(l)).$   
 $\text{Nevyhovuje}(f) \leftarrow \text{Firmy}(f, x_1), Vba(v, c, l), \neg \text{Dodava}(f, v, x_2, x_3)$   
 $\text{Firmy}(f, x_1), Vba(v, c_1, l_1), \text{Dodava}(f, v, c_2, l_2), c_1 \leq c_2, l_1 \leq l_2.$   
 $\text{Answer}(f) \leftarrow \text{Firmy}(f, x_1), \neg \text{Nevyhovuje}(f).$

11) Sformulujte nasledujúci dotaz v relačnom kalkule (2), relačnej algebre (2), Datalogu (2) a SQL (2): Nájdite výrobky, ktoré si objednalo aspoň 1000 klientov a ktoré dodávajú firmy z Bratislavy alebo zo Senca. (Pre každý ten výrobok musia platiť obidve tieto podmienky.) Spolu s každým výrobkom vypíšte počet klientov, ktorí si ten výrobok objednali.

$\{\langle v, p \rangle : (\bigvee (p = \text{count}(k) O(k, v)) \wedge (p \geq 1000)) \wedge (\exists (f, m, x_1, x_2) (D(f, v, x_1, x_2) \wedge F(f, m) \wedge (m = \text{Bratislava} \vee m = \text{Senec})))\}$   
 $(\sigma_{p \geq 1000}(\Gamma_{v, p \leftarrow \text{count}(k)} O) \bowtie (\Pi_v (D \bowtie (\sigma_{m = \text{Bratislava} \vee m = \text{Senec}} F))))$

OVT(v, p) ← Subtotal(O(k,v);v;p=count(k)), p≥1000.  
 BaSe(v) ← D(f,v,x<sub>1</sub>,x<sub>2</sub>),F(f,m),(m=Bratislava ∨ m=Senec).  
 Answer(v,p) ← OVT(v, p),BaSe(v).

**select** O,vyrobok, count(O.klient) **as** p **from** Objednavky O, Firmy F, Dodava D  
**where** F.mesto **in** ("Bratislava", "Senec") **and**  
 D.firma = F.firma **and** D.vyrobok = O.vyrobok  
**group by** O.vyrobok **having** count(O.klient) ≥1000;

12) Zapište nasledujúci dotaz v slovenskom jazyku (čo najpresnejšie) (1), v SQL (2) a v Datalogu (2):

{K: ∃V1∀V2∀C∀L  
 (objednavky(K, V1) ∧ (¬ objednavky(K, V2) ∨ ¬ dodava(datasys, V2, C, L)))}

(Všetci) klienti, ktorí si neobjednali žiaden vyrobok dodávaný firmou Datasys.  
 Pozn.: Podľa princípu byrokracie, klient je iba ten, kto si niečo objednal.

ODS(k) ← O(k,v), D(Datasys,v.c,l).  
 Answer(k) ← O(k,v),¬ ODS(k).  
**select** O.klient **from** Objednavky O  
**where** O.vyrobok **not in** (select D.vyrobok) **from** Dodava D  
**where** D.firma="Datasys";

II. Daná je databáza

capuje(krcma, alkohol)	C(k, a)
lubi(pijan, alkohol)	L(p, a)
navstivil(idn, pijan, krcma, od, do)	N(i, p, k, o, d)
vypil(idn, alkohol, mnozstvo)	V(i, a, m)

1) Sformulujte nasledujúci dotaz v relačnom kalkule (2), relačnej algebre (2), Datalogu (2) a SQL (2): Nájdite dvojice [krcma, mnozstvo], ktoré hovoria, koľko alkoholu Curaçao sa v ktorej krčme vypilo. Pritom treba nájsť len tie krčmy, kde sa vypilo menej ako 10 jednotiek Curaçao (k týmto patria aj tie krčmy, kde sa alkohol Curaçao nečapuje; a tiež tie krčmy, ktoré Curaçao síce čapujú, ale vypilo sa ho 0 jednotiek).

{⟨k,q⟩: ((∃i, q=sum(m))(∃(p,o,d) N(i, p, k, o, d) ∧ V(i, Cu, m))) ∧ q<10 ∨  
 ∀(a,i,p,o,d) ((N(i, p, k, o, d) ∧ V(i, a, m)) ⇒ (a ≠ Cu)) ∧ q=0) ∨  
 (¬∃(a,i,p,o,d) C(k, a) ∧ N(i, p, k, o, d)) ∧ q=0 }

Buď sa tam vypilo menej ako 10l Curaçao, alebo keď sa tam niečo pilo nebolo to Curaçao, alebo krčmu vôbec nikto nenavštívil.

$\sigma_{q<10}(\Gamma_{k,q \leftarrow \text{sum}(m)}(\prod_{i,m,k} (N \bowtie (\sigma_{a=Cu} V))) \cup (\prod_k C) \times \langle i=-1, m=0 \rangle))$

NPC(k, i, m) ← N(i, \_, k, \_, \_), V(i, Cu, m). /\* Pilo sa Curaçao \*/  
 NPC(k, i, m) ← C(k, \_), i=-1, m=0. /\* Inicializácia nuly, pre všetky kčmy \*/  
 Answer(k,q) ← Subtotal(NPC(k,i,m);k;q=sum(m)), q < 10.

```

with NPC(k, i, m) as (select N.k, N.i, V.m from Navstivil N, Vypil V
                    where N.i=V.i and V.a = " Curaçao"
                    union select C.k, -1 as i, 0 as m from Capuju C)
do select k, sum(m) as q from NPC
/* do sa nepoužíva v SQL99, pridal som ho kvôli čitateľnosti. */
   group by k having sum(m) < 10;

```

Iné riešenie:

$$\{ \langle k, q \rangle : ((\exists i, q = \text{sum}(m))(\exists (p, o, d) N(i, p, k, o, d) \wedge V(i, Cu, m))) \wedge q < 10 \vee ((\exists a C(k, a)) \wedge \neg(\exists (i, p, o, d, m) (N(i, p, k, o, d) \wedge V(i, Cu, m))) \wedge q = 0) \}$$

Buď sa tam vypilo menej ako 10l Curaçao, alebo sa v krčme Curaçao nepilo.

$$\sigma_{q < 10}(\Gamma_{k, q \leftarrow \text{sum}(m)}(\prod_{i, m, k} (N \bowtie (\sigma_{a=Cu} V)))) \cup (\prod_k C - (\prod_k (N \bowtie (\sigma_{a=Cu} V)))) \times \langle q = 0 \rangle$$

$NPC(k, i, m) \leftarrow N(i, \_, k, \_, \_), V(i, Cu, m).$  /\* Pilo sa Curaçao \*/

$Answer(k, q) \leftarrow \text{Subtotal}(NPC(k, i, m); k; q = \text{sum}(m)), q < 10.$

$Answer(k, q) \leftarrow C(k, \_), \neg NPC(k, \_, \_), q = 0.$  /\* Nepilo sa Curaçao \*/

```

select N.k, sum(V.m) as q from Navstivil N, Vypil V
where N.i=V.i and V.a = " Curaçao"
group by k having sum(V.m) < 10
union
select C.k, 0 as q from Capuju C
where C.k not in (select N.k, sum(V.m) as q from Navstivil N, Vypil V
                 where N.i=V.i and V.a = " Curaçao");

```

2) Zapište nasledujúci dotaz v slovenskom jazyku (čo najpresnejšie) (1), v SQL (2) a v Datalogu (2):  $\{P: \forall K \forall A \exists I \exists O \exists D$

$$((\text{lubi}(P, A) \wedge \text{capuju}(K, A)) \Rightarrow \text{navstivil}(I, P, K, O, D))\}$$

Pijani, ktorí navštívili všetky krčmy, kde čapujú aspoň jeden alkohol, ktorí ľúbia.

```

select p from L union N /* To je skratka za union ten istý dotaz pre N */
where not exists (select k from C, L where C.a = L.a and
                 C.k not in (select N.k from N where N.p=p) );

```

Datalóg:

$Vsetci(p) \leftarrow L(p, \_) \vee N(\_, p, \_, \_).$

/\*Skratka za dve pravidlá s tou istou ľavou stranou.\*/

$Neuspesny(p) \leftarrow L(p, a), C(k, a), \neg N(\_, p, k, \_).$

$Answer(p) \leftarrow Vsetci(p), \neg Neuspesny(p).$

3) Nájdite všetkých pijanov, z ktorých každý vypil len to, čo ľúbi. Sformulujte tento dotaz v relačnom kalkule (2), Datalogu (2) a SQL (2).

$$\{p: \forall(i,k,o,d,m,a) N(i,p,k,o,d) \wedge V(i,a,m) \Rightarrow L(p,a)\}$$

$$\{p: ((\exists a)(\forall i,k,o,d)(L(p,a) \wedge \neg N(i,p,k,o,d))) \vee \neg((\exists i,k,o,d,a)(N(i,p,k,o,d) \wedge V(i,a,m) \wedge \neg L(p,a)))\}$$

Ani jedna formula nie je presným prekladom dotazu. Presne vzaté sú to všetci pijani, ktorí pri návštevách krčmy nepili žiaden alkohol, čo neľúbia. Lepšie sa to urobiť nedá, lebo o tom, čo a koľko pili doma, databáza nemá evidenciu. (Tajne predpokladáme, že doma pili len to, čo ľúbia.)

VP(p) ← L(p,\_) ∨ N(\_,p,\_,\_,\_). /\* Všetci pijani \*/

VaN(p) ← N(i,p,\_,\_,\_), V(i,a,\_), ¬L(p,a). /\*Vypil a niečo, čo neľúbi.\*/

Answer(p) ← Vp(p), ¬VaN(p).

```
with (select L.p as p from L union select N.p as p from N) as VP
select Vp.p from Vp where Vp.p not in
  (select N.p from N, V where V.i = N.i and V.a not in
    (select L.a from L where L.p = N.p ));
```

4) Nájdite dvojice [pijan, maxpocet], ktoré hovoria o každom pijanovi, maximálne koľko rôznych druhov alkoholu vypil ten pijan na jedno posedenie (t.j. pri 1 návšteve krčmy). Sformulujte tento dotaz v SQL (3).

Otázka je, či majú byť vo výsledku aj pijani, ktorí nikdy nesedeli v krčme. Rozhol som sa, že tam budú s hodnotou maxpocet 0.

**with**

```
select N.p, N.i, count(distinct V.a) from N, V group by N.p, N.i as G(p, i, pd)
select G.p as pijan, max(G.pd) as maxpocet from G group by G.p
union select L.p, 0 as maxpocet from L where L.p not in (select N.p from N);
```

5) Sformulujte nasledujúci dotaz v relačnom kalkule (2), relačnej algebre (2), Datalogu (2) a SQL (2): Nájdite dvojice [pijan, pocet], ktoré o každom pijanovi hovoria, pri koľkých návštevách pil nejaký alkohol, ktorý neľúbi.

$$\{\langle p,n \rangle: (\exists n = \text{count}(i))((\exists k, o, d, m)(N(i,p,k,o,d) \wedge V(i,a,m) \wedge \neg L(p,a)))\}$$

$$\Gamma_{p,n \leftarrow \text{count}(i)}(\prod_{p,i,a}(N \bowtie V) - \prod_{p,i,a}(L \bowtie N))$$

PSS(p, i) ← N(i, p, \_, \_, \_), V(i, a, \_), ¬L(p,a).

Answer(p, n) ← Subtotal(PSS(p,i); p; n=count(i)).

```
select N.p, count(N.i) as n from Navstivil N, Vypil V
where N.i = V.i and V.a not in (select L.a from Lubi L where L.p = N.p)
group by N.p;
```

6) Zapište nasledujúci dotaz v relačnom kalkule (2), v SQL (2) a v Datalogu (2):  
Nájdite všetky krčmy, pre ktoré platí, že ktorýkoľvek pijan, ktorý tú krčmu navštívil, v nej našiel aspoň jeden druh alkoholu, ktorý ľúbi (z toho ešte nutne nevyplýva, že ten pijan ten alkohol pil).

$$\{k: (\forall p,i,o,d)(\exists a) N(i,p,k,o,d) \wedge L(p,a) \Rightarrow C(k,a)\}$$

UN(k,i)  $\leftarrow$  N(i,p,k,\_,\_), L(p,a), C(k,a). /\*Úspešná návšteva krčmy\*/

NK(k)  $\leftarrow$  N(i,\_,k,\_,\_),  $\neg$ UN(k,i). /\*Krčma mala neúspešnú návštevu\*/

Answer(k)  $\leftarrow$  C(k, \_),  $\neg$ NK(k). /\*Krčmy, čo nemali neúspešnú návštevu\*/

**select distinct C.k from Capuju C**

**where C.k not in (select N.k from Navstivil N**

**where N.i not in (select N.i from Navstivil N, Lubi L, Capuju C**

**where L.p = N.p and C.k = N.k and L.a = C.a));**

7.) Popíšte spôsob, akým sa prekladá SQL do relačnej algebry (1). b) Popíšte spôsob, akým sa prekladá Datalog do SQL (2). c) Demonštrujte preklad z Datalogu do SQL na dotaze „Nájdite pijanov, ktorí neľúbia Gin a zároveň nikdy nenavštívili krčmu 'Hacienda'“. (Pijan je definovaný ako niekto, kto buď ľúbi nejaký alkohol alebo navštevuje krčmy, nie nutne oboje súčasne.) (3)

Asi ťažké a nejednoznačné. SQL je príliš barokný.

Základná schéma: **select**  $a_1, a_2, \dots, a_m$  **from**  $R_1, R_2, \dots, R_n$

**where** C;

sa prekladá na BE  $\equiv \prod_{a_1, a_2, \dots, a_m} \sigma_C R_1 \times R_2 \times \dots \times R_n$

Ak obsahuje group by a agregáčné funkcie použije sa na neagregovany select operácia  $\Gamma$  prípadne ďalšia selekcia alebo join pre podmienku having struktura výrazu je

$\sigma_H(\Gamma(BE))$ . Taketo selekty môžu byť pospájané operáciami union ( $\cup$ ) alebo minus ( $-$ ). Optimalizácia je nahradiť kartézsky súčin a selekciu príslušným joinom. Dôležité je hlavne v prípade selekcií na rovnosť ( $\bowtie$ ). Joiny vznikajú aj z vnorených pozitívnych dotazov. Vnorené negatívne dotazy generujú rozdiel. Rozdiel a zjednotenie požadujú, aby operandy boli relácie rovnakej arity. To sa dosahuje vhodným kartézskym súčinom, alebo joinom. Niekedy sa hodí operácia „antijoin“ ( $R(x,y) \triangleleft S(y,z) = R - \prod_{x,y} R \bowtie S$ ).

Je to štandardný trik ako urobiť rozdiel relácií rôznej arity.

Preklad datalógu do SQL. Ešte komplikovanejšie.

Pokiaľ ide o jedno pravidlo preložil by som ho najprv do algebry. To je ľahké:

Najprv, konštanty v predikatoch nahradíme premennymi a pridáme

"premena=konštanta".

Textové náhrady. Najprv " $\neg$ " nahradíme " $\triangleleft$ ", potom zvyšné čiarky nahradíme prirodzeným spojením. Ešte prirodzené spojenia s nekonečnými

(zabudovanými) reláciami prepíšeme na selekcie a nakoniec urobíme projekciu na atributy hlavy. Z takehoto výrazu SQL výraz ako



**select** <zoznam atributov hlavy> **from** <pozitívne predikaty>  
**where** <spájacie podmienky> **and** <selekčné podmienky> **and** <vložené dotazy>;

Vložený dotaz vyrobíme pre každý antijoin ako

<spoločné atribúty> **not in** (**select** <spoločné atribúty>**from** Relácia **where** <selekčné podmienky>). Takto preložíme, každé je pravidlo.

Subtotal priamo zodpovedá sql príkazu. Pravidla z rovnakou hlavou spojíme do jedného príkazu tak, že preklady ich tiel spojíme v SQL príkazom **union**.

Keď máme všetky pravidlá preložené preložíme celý program ako:

**with** <preklad pravidla 1> **as** hlava1, ..., <preklad pravidla n> **as** hlavam  
 <pravidlo pre answer>;

Takto vznikly SQL nie je pekný, množstvo SQL príkazov za **with**. Ak sa definovaný predikát používa len, pri výpočte jedného predikátu je výhodné vložiť ho do SQL príkazu pre výpočet tohto predikátu. Pre takéto rozhodnutia je vhodné nakresliť si graf závislosti predikátov.

Pokiaľ náš SQL podporuje **with recursive** (with iterate) môžeme prekladať i rekurzívne pravidlá. Funguje to pre bezpečné pravidlá a stratifikovanú sémantiku negácie.

Vsetci(p) ← Lubi(p,\_).

Vsetci(p) ← Navstivil(\_,p,\_,\_,\_).

Answer(p) ← Vsetci(p), ¬Lubi(p, "Gin"), ¬Navstivil(\_p, "Hacienda", \_,\_).

**with select** L.p **as** p **from** Lubi L **union select** N.p **as** p **from** Navstivil N  
**select** VP.p **from** Vsetci VP

**where** VP.p **not in** (**select** L.p **from** Lubi L **where** a = "Gin") **and**

VP.p **not in** (**select** N.p **from** Navstivil N **where** k = "Hacienda");

Všimnite si, že syntax SQL vyžaduje prispôsobenie mien premenných.

Namiesto príkazu **with** možno použiť I príkaz **create view** tým sa, ale definitívne vzdávame rekurzívnych programov, navyše tento príkaz vyžaduje privilégium create voči databáze, ktoré nie vždy musíme mať.

Niekomu môže vadiť veľa vnorených môže sa ich nahradiť vnorením negácie jedného pozitívneho dotazu. V danom príklade:

**with select** L.p **as** p **from** Lubi L **union select** N.p **as** p **from** Navstivil N  
**select** VP.p **from** Vsetci VP

**where** VP.p **not in** (**select** L.p **from** Lubi L **where** a = "Gin"

**union**

**select** N.p **from** Navstivil N **where** k = "Hacienda");

V zložitejších príkladoch je dobre robiť to formálne pomocou DeMorganových zákonov. Vyhnete sa zbytočným chybám.