

Uvažujte databázu: $\text{capuje}(\text{Krcma}, \text{Alkohol}, \text{Cena})$,
 $\text{lubi}(\text{Pijan}, \text{Alkohol})$, $\text{navstivil}(\text{Idn}, \text{Pijan}, \text{Krcma})$,
 $\text{vypil}(\text{Idn}, \text{Alkohol}, \text{Mnozstvo})$.

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

Každá krčma čapuje nejaký alkohol.

Sformulujte bezpečný dotaz v Datalogu, ktorý nájde:

1. Dvojice $[P, \text{Suma}]$, kde Suma vyjadruje koľko peňazí pijan P celkovo prepil v krčmách, ktoré čapujú len alkoholy, ktoré P neľúbi. Vo výsledku nemajú byť dvojice s nulovou sumou. **(6)**

```
answer(P, Suma) ←  
    subtotal(ucet_capuje_nelubi(_, P, A, U), [P], [Suma = sum(U)]).
```

```
ucet_capuje_nelubi(I, P, A, U) ←  
    navstivil(I, P, K),  
    capuje(K, A, C),  
    vypil(I, A, M),  
    U = M * C,  
    not nieco_lubi(P, K).
```

```
nieco_lubi(P, K) ←  
    lubi(P, A),  
    capuje(K, A, _).
```

2. Dvojice $[P, K]$ také, že pijan P niekedy navštívil krčmu K, a pri každej návšteve krčmy K vypil rovnakú množinu alkoholov. **(6)**

```
answer(P, K) ←  
    navstivil(_, P, K),  
    not vypil_nevypil(P, K).
```

```
vypil_nevypil(P, K) ←  
    navstivil(I1, P, K),  
    v(I1, A),  
    navstivil(I2, P, K),  
    not v(I2, A).
```

```
v(I, A) ←  
    vypil(I, A, _).
```

3. Dvojice [K, Suma], ktoré hovoria, koľko peňazí v krčme K celkovo prepili takí pijani, ktorí krčmu K navštívili viac než stokrát, a okrem K navštívili nanajvýš jednu inú krčmu. Vo výsledku nemajú byť dvojice s nulovou sumou. **(6)**

```
answer(K, Suma) ←  
    subtotal(ucet100plus1(_, K, _, U), [K], [Suma = sum(U)]).
```

```
ucet100plus1(I, K, A, U) ←  
    subtotal(navstivil(I, P, K), [P, K], [N = count(I)],  
        N >= 100,  
        navstivil(I, P, K),  
        vypil(I, A, M),  
        capuje(K, A, C),  
        U = M * C,  
        not navstivil_dve_ine(P, K).
```

```
navstivil_dve_ine(P, K) ←  
    navstivil(_, P, K),  
    navstivil(_, P, K1),  
    navstivil(_, P, K2),  
    not K = K1,  
    not K = K2,  
    not K1 = K2.
```

4. Dvojice [P, X], kde X je počet krčiem, v ktorých pijan P niečo vypil, a prepil v nich aspoň 15 EUR počas každej svojej návštevy. **(6)**

```
answer(P, X) ←  
    subtotal(prepil15(P, K), [P], [X = count(K)]).
```

```
prepil15(P, K) ←  
    navstivil(_, P, K),  
    not niekedy_malo(P, K).
```

```
niekedy_malo(P, K) ←  
    navstivil(I, P, K),  
    not v(I).
```

```
niekedy_malo(P, K) ←  
    navstivil(I, P, K),  
    subtotal(ucet(I, _, U), [I], [S = sum(U)]),  
    S < 15.
```

```
ucet(I, A, U) ←  
    navstivil(I, _, K),  
    vypil(I, A, M),  
    capuje(K, A, C),  
    U = M * C.
```

```
v(I) ←  
    vypil(I, _, _).
```

5. Dvojice $[K, A]$ pre ktoré platí, že alkohol A vypilo v krčme K aspoň 20 rôznych pijanov, pričom každý z týchto pijanov vypil alkohol A v krčme K v celkovom množstve aspoň 100. **(6)**

```
answer(K, A) ←  
  subtotal(aspon100(P, K, A), [K, A], [C = count(P)]),  
  C >= 20.  
  
aspon100(P, K, A) ←  
  subtotal(konzum(_, P, K, A, M), [P, K, A], [S = sum(M)]),  
  S >= 100.  
  
konzum(I, P, K, A, M) ←  
  navstivil(I, P, K),  
  vypil(I, A, M).
```

6. Dvojice $[P, K]$, také, že P navštívil K aspoň dvakrát; pri každej návšteve K vypil aspoň 3 druhy alkoholu, z ktorých jeden bol fernet alebo rum; ale pri žiadnej návšteve K nevypil fernet aj rum. **(6)**

```
answer(P, K) ←  
  navstivil(I1, P, K),  
  navstivil(I2, P, K),  
  not I1 = I2,  
  not niekedy_nespravne(P, K).  
  
niekedy_nespravne(P, K) ←  
  navstivil(I, P, K),  
  not vypil_dost(I).  
  
niekedy_nespravne(P, K) ←  
  navstivil(I, P, K),  
  not v(I, fernet),  
  not v(I, rum).  
  
niekedy_nespravne(P, K) ←  
  navstivil(I, P, K),  
  vypil(I, fernet, _),  
  vypil(I, rum, _).  
  
vypil_dost(I) ←  
  navstivil(I, _, _),  
  v(I, A1),  
  v(I, A2),  
  v(I, A3),  
  not A1 = A2,  
  not A1 = A3,  
  not A2 = A3.  
  
v(I, A) ←  
  vypil(I, A, _).
```

7. Dvojice $[P, A]$, pre ktoré platí, že pijan P vypil alkohol A v aspoň 30 rôznych krčmách, pričom v každej z tých 30 krčiem vypil celkovo aspoň 50 jednotiek alkoholu A . **(6)**

```
answer(P, A) ←  
  subtotal(aspon50(P, A, K), [P, A], [C = count(K)]),  
  C >= 30.
```

```
aspon50(P, A, K) ←  
  subtotal(konzum(_, P, A, K, M), [P, A, K], [S = sum(M)]),  
  S >= 50.
```

```
konzum(I, P, A, K, M) ←  
  navstivil(I, P, K),  
  vypil(I, A, M).
```

8. Dvojice pijanov $[P1, P2]$ také, že $P1$ navštívil všetky krčmy, ktoré navštívil $P2$, a okrem nich $P1$ navštívil ešte aspoň 2 iné. **(6)**

```
answer(P1, P2) ←  
  n(P, K1),  
  n(P, K2),  
  pijan(P2),  
  not K1 = K2,  
  not n(P2, K1),  
  not n(P2, K2),  
  not nenavstivil_niektoru(P1, P2).
```

```
pijan(P) ←  
  lubi(P, _).
```

```
pijan(P) ←  
  n(P, _).
```

```
nenavstivil_niektoru(P1, P2) ←  
  n(P1, _),  
  navstivil(_, P2, K),  
  not n(P1, K).
```

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

9. Dvojice krčiem [K1, K2] také, že každý pijan, ktorý navštívil K2, navštívil K1 viackrát než K2 a prepil v K1 viacej peňazí než v K2. **(6)**

```
answer(K1, K2) ←  
    capuje(K1, _, _),  
    capuje(K2, _, _),  
    not navstivil_malo(K1, K2),  
    not prepil_malo(K1, K2).
```

```
navstivil_malo(K1, K2) ←  
    capuje(K1, _, _),  
    navstivil(_, P, K2),  
    not n(P, K1).
```

```
navstivil_malo(K1, K2) ←  
    subtotal(navstivil(I, P, K1), [P, K1], [C1 = count(I)]),  
    subtotal(navstivil(I, P, K2), [P, K2], [C2 = count(I)]),  
    C1 <= C2.
```

```
prepil_malo(K1, K2) ←  
    capuje(K1, _, _),  
    v(P, K2),  
    not v(P, K1).
```

```
prepil_malo(K1, K2) ←  
    subtotal(ucet(_, P, K1, _, U1), [P, K1], [S1 = sum(U1)]),  
    subtotal(ucet(_, P, K2, _, U2), [P, K2], [S2 = sum(U2)]),  
    S1 <= S2.
```

```
ucet(I, P, K, A, U) ←  
    navstivil(I, P, K),  
    vypil(I, A, M),  
    capuje(K, A, C),  
    U = M * C.
```

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

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

10. Trojice $[P, K, A]$ také, že krčma K čapuje alkohol A ; a zároveň P je jediný návštěvník K , který líbí A a který alkohol A v krčmě K nikdy nevypil. **(6)**

```
answer(P, K, A) ←  
    capuje(K, A, _),  
    navstivil(_, P, K),  
    lubi(P, A),  
    not niekedy_vypil(P, K, A),  
    not iny_taky(P, K, A),
```

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

```
iny_taky(P, K, A) ←  
    lubi(P, A),  
    navstivil(_, P2, K),  
    lubi(P2, A),  
    not P2 = P,  
    not niekedy_vypil(P2, K, A).
```