

1 Dotazovacie jazyky

- relačný kalkul
- datalog
- SQL
- relačná algebra (na rozdiel od predošlých aj špecifikuje postup výpočtu)

Pri premýšľaní nad dotazom (alebo jeho časťou) môžete použiť ľubovoľný z jazykov a potom výsledok prepísať do iného jazyka. Toto sa asi najľahšie realizuje pri prepise z datalogu do SQL alebo relačnej algebry. SQL je jazyk pomerne ťažkopádny a má niekoľko nevýhod (pozri napr. <https://web.archive.org/web/20110305230025/http://www.cs.duke.edu/courses/spring03/cps216/papers/date-1983.pdf>), ktoré vznikli tým, že nebol navrhnutý „profesionálne“, ale skôr narýchlo ad-hoc a potom sa presadil na trhu napriek existencii technicky lepších alternatív.

Nebojte sa jazyky kombinovať. Jazyk je v princípe nástroj, a preto sa netreba báť použiť kombináciu nástrojov na riešenie jedného problému. Napríklad na vyjadrenie komplikovanejšej štruktúry kvantifikátorov datalog, ktorý neskôr mechanicky prepíšeme do SQL, a agregáciu nad tými kvantifikovanými dátami zapíšeme rovno v SQL. (Výhodou datalogu oproti SQL pri kvantifikátoroch je ďaleko stručnejší zápis a hlavne sa nestatíme v množstve podmienok za WHERE, ktoré stotožňujú výskyty jednej premennej — v datalogu takéto niečo máme zadarmo, stačí použiť rovnaké písmeno.)

Voľba jazyka, v ktorom premýšľame nad riešením problému, je veľmi dôležitá. Skúste chvíľu reprezentovať čísla pomocou rímskych číslíc (vrátane výpočtov typu násobenie) a rýchlo pochopíte jeden z dôvodov, prečo to Rimania v matematike ďaleko nedotiahli. O nič lepšia nie je reprezentácia čísel pomocou kmeňových zlomkov ako v Egypte. Na druhej strane Arabi s nulou a pozičnou sústavou spravili citeľný pokrok. Alternatívne, predstavte si, ako by ste písali webserver alebo databázové dotazy v asembleri.

Riešenia sú naschvál zapisované s rôznym spôsobom formátovania: vyberte si taký, čo sa vám pozdáva (najmä čo sa týka SQL). Snažil som sa nepoužiť ťažko čitateľné spôsoby; tie uvedené by mali byť aspoň v princípe OK, aj keď niektoré z nich sa mi pozdávajú viac ako iné.

Pri práci v relačnej algebre či SQL môžete využiť online prostredie RelaX. Linky nižšie už obsahujú aj databázovú schému pre databázu pijanov (rozdiel je len v názvoch stĺpcov).

<https://dbis-uibk.github.io/relax/calc/gist/51930198d605017448b37c128684e555>

<https://dbis-uibk.github.io/relax/calc/gist/5cbffeac631d5fab2f1885e5e922841e>

2 Úvodné príklady (relačný kalkul, datalog)

$$EDB = \{\text{osoba}(O), \text{firma}(F), \text{vlastni}(Kto, Co)\}$$

V tejto časti uvádzame len správne riešenia úloh, nie príklady nesprávnych riešení.

Úloha 2.1. Firmy vlastnené jedinou osobou a ničím iným.

$$\begin{aligned} &\{F \mid \text{firma}(F) \wedge \exists O(\text{osoba}(O) \wedge \text{vlastni}(O, F) \wedge \forall X(\text{vlastni}(X, F) \implies X = O))\} \\ &\{F \mid \text{firma}(F) \wedge \exists O(\text{osoba}(O) \wedge \text{vlastni}(O, F) \wedge \neg \exists X(\text{vlastni}(X, F) \wedge X \neq O))\} \end{aligned}$$

$$\text{answer}(F) \leftarrow \text{vlastni}(O, F), \text{osoba}(O), \neg \text{inyVlastnik}(F, O).$$

$$\text{inyVlastnik}(F, O) \leftarrow \text{vlastni}(O, F), \text{vlastni}(O_2, F), O_2 \neq O.$$

$$\text{answer}(F) \leftarrow \text{vlastni}(O, F), \text{osoba}(O), \neg \text{dvajaVlastnici}(F).$$

$$\text{dvajaVlastnici}(F) \leftarrow \text{vlastni}(O_1, F), \text{vlastni}(O_2, F), O_1 \neq O_2.$$

Úloha 2.2. Koneční užívatelia výhod, t.j. osoby, ktoré vlastní firmu, a nie sú vlastnené inou osobou ani firmou. (Pozor: v databáze zatiaľ nie je žiadna kontrola integrity, čiže môžu byť ako vlastníci uvedené entity, ktoré nie sú osobami ani firmami.)

$$\{O \mid \text{osoba}(O) \wedge \exists F(\text{firma}(F) \wedge \text{vlastni}(O, F)) \wedge \neg \exists X(\text{vlastni}(X, O) \wedge (\text{osoba}(X) \vee \text{firma}(X)))\}$$

$$\text{answer}(O) \leftarrow \text{osoba}(O), \text{firma}(F), \text{vlastni}(O, F), \neg \text{owned}(O).$$

$$\text{owned}(O) \leftarrow \text{vlastni}(X, O), \text{osoba}(X).$$

$$\text{owned}(O) \leftarrow \text{vlastni}(X, O), \text{firma}(X).$$

3 Nesprávnych riešení je hocikolko... (relačný kalkul, datalog)

Posolstvom tejto časti je ukázať, že je bezpočet možností, ako napísať chybné riešenie, hoci mnohé chybné riešenia majú k správne celkom blízko. Všetky riešenia pochádzajú z istej rozcvičky písanej na treťom cvičení.

Mali by ste trénovať spochybňovanie akéhokolvek kódu či formálneho zápisu, čo vám príde do rúk (vlastného i cudzieho). Preštudujte si všetky chybné riešenia a skúste pre každé z nich v prirodzenom jazyku vyjadriť, čo popisuje, alebo aspoň zdôvodniť, prečo je nesprávne.

$$EDB = \{\text{ponuka}(\text{Miesto}, \text{Akcía}), \text{chce}(\text{Človek}, \text{Miesto}, \text{Akcía})\}$$

Úloha 3.1. Akcie, ktoré sú v ponuke na každom mieste, kde ich chcú aspoň dvaja (a sú aspoň niekde v ponuke).

Správne riešenie (relačný kalkul, tri ekvivalentné zápisy):

$$\{A \mid (\exists M \text{ponuka}(M, A)) \wedge \forall M \forall C_1 \forall C_2 [C_1 \neq C_2 \wedge \text{chce}(C_1, M, A) \wedge \text{chce}(C_2, M, A) \implies \text{ponuka}(M, A)]\}$$

$$\{A \mid (\exists M \text{ponuka}(M, A)) \wedge \neg \exists M \exists C_1 \exists C_2 \neg [C_1 \neq C_2 \wedge \text{chce}(C_1, M, A) \wedge \text{chce}(C_2, M, A) \implies \text{ponuka}(M, A)]\}$$

$$\{A \mid (\exists M \text{ponuka}(M, A)) \wedge \neg \exists M \exists C_1 \exists C_2 [C_1 \neq C_2 \wedge \text{chce}(C_1, M, A) \wedge \text{chce}(C_2, M, A) \wedge \neg \text{ponuka}(M, A)]\}$$

Správne riešenie (relačný kalkul, tri ekvivalentné zápisy):

$$\{A \mid (\exists M \text{ponuka}(M, A)) \wedge \forall M [(\exists C_1 \exists C_2 (C_1 \neq C_2 \wedge \text{chce}(C_1, M, A) \wedge \text{chce}(C_2, M, A))) \implies \text{ponuka}(M, A)]\}$$

$$\{A \mid (\exists M \text{ponuka}(M, A)) \wedge \neg \exists M [(\exists C_1 \exists C_2 (C_1 \neq C_2 \wedge \text{chce}(C_1, M, A) \wedge \text{chce}(C_2, M, A))) \wedge \neg \text{ponuka}(M, A)]\}$$

$$\{A \mid (\exists M \text{ponuka}(M, A)) \wedge \neg \exists M \exists C_1 \exists C_2 [C_1 \neq C_2 \wedge \text{chce}(C_1, M, A) \wedge \text{chce}(C_2, M, A) \wedge \neg \text{ponuka}(M, A)]\}$$

Správne riešenie (datalog) — porovnajte so zápsmi vyššie:

$$\text{answer}(A) \leftarrow \text{ponuka}(_, A), \neg \text{niekdeNeponuka}(A).$$

$$\text{niekdeNeponuka}(A) \leftarrow \text{chce}(C_1, M, A), \text{chce}(C_2, M, A), C_1 \neq C_2, \neg \text{ponuka}(M, A).$$

Všetky nasledujúce riešenia sú nesprávne (preto tá červená farba). Viete zistiť, v čom sú chybné?

$$\text{answer}(A) \leftarrow \text{ponuka}(M, A), \text{chce}(C_1, M, A), \text{chce}(C_2, M, A), C_1 \neq C_2.$$

$$\text{answer}(A) \leftarrow \text{ponuka}(M, A), \text{miestoChcuDvaja}(M).$$

$$\text{miestoChcuDvaja}(M) \leftarrow \text{chce}(C_1, M, _), \text{chce}(C_2, M, _), C_1 \neq C_2.$$

$$\text{chcuAsponDvaja}(M, A) \leftarrow \text{chce}(C_1, M, A), \text{chce}(C_2, M, A), C_1 \neq C_2.$$

$$\text{zlaAkcía}(M, A) \leftarrow \text{chcuAsponDvaja}(M, A), \neg \text{ponuka}(M, A).$$

$$\text{vysledok}(A) \leftarrow \text{ponuka}(_, A), \text{chce}(_, M, _), \neg \text{zlaAkcía}(M, A).$$

$\text{akciuAsponDvaja}(M, A) \leftarrow \text{ponuka}(M, A), \text{chce}(C_1, M, A), \text{chce}(C_2, M, A), C_1 \neq C_2.$
 $\text{nechcuDvaja}(A) \leftarrow \text{ponuka}(M, A), \neg \text{akciuAsponDvaja}(M, A).$
 $\text{odpoved}(A) \leftarrow \text{ponuka}(_, A), \neg \text{nechcuDvaja}(A).$

$\text{pomocna}(A) \leftarrow \text{chce}(C_1, M, A), \text{chce}(C_2, M, A), C_1 \neq C_2, \neg \text{ponuka}(M, A).$
 $\text{answer}(A) \leftarrow \text{ponuka}(_, A), \text{pomocna}(A).$

$\text{chcu2}(A) \leftarrow \text{chce}(C_1, _, A), \text{chce}(C_2, _, A), C_1 \neq C_2, \neg \text{nievsetky}(A).$
 $\text{nievsetky}(A) \leftarrow \text{ponuka}(_, A), \neg \text{ponuka}(M, A).$

$\text{ans}(a) \leftarrow \text{ponuka}(_, A), \neg \text{pomoc1}(A).$
 $\text{pomoc1}(A) \leftarrow \text{ponuka}(M, A), \text{pomoc2}(M, A).$
 $\text{pomoc2}(M, A) \leftarrow \text{chce}(P_1, M, A), \text{chce}(P_2, M, A), P_1 \neq P_2, \neg \text{ponuka}(M, A).$

$\text{answer}(A) \leftarrow \text{ponuka}(_, A), \neg \text{chceAviac2neni}(A).$
 $\text{chceAviac2neni}(A) \leftarrow \text{ponuka}(_, A), \neg \text{chceAmenej2aje}(A).$
 $\text{chceAmenej2aje}(A) \leftarrow \text{chce}(C_1, M, A), \text{chce}(C_2, M, A), C_1 \neq C_2.$

$\text{answer}(A) \leftarrow \text{ponuka}(M, A), \neg \text{nehcudvaja}(A, M), \text{chce}(_, M, A).$
 $\text{nehcudvaja}(A, M) \leftarrow \text{chce}(C_1, M, A), \text{chce}(C_2, M, A), C_1 = C_2.$

$\text{answer}(A) \leftarrow \text{ponuka}(_, A), \neg \text{notchcu}(A).$
 $\text{notchcu}(A) \leftarrow \text{ponuka}(_, A), \text{chce}(C_1, M, _), \text{chce}(C_2, M, _), C_1 \neq C_2, \neg \text{not2}(A, M).$
 $\text{not2}(A, M) \leftarrow \text{chce}(_, M, A).$

4 Bežné chyby

V mnohých oblastiach ľudskej činnosti viete dosiahnuť veľmi dobré výsledky čisto tým, že sa vyhýbate chybám (silne odporúčam pohľadať na internete niečo na tému „invert, always invert“; môžete začať napríklad týmito: <https://www.biznews.com/thought-leaders/1986/06/13/charlie-mungers-speech-to-the-harvard-school-june-1986>).

Do istej miery to platí to aj pre tento predmet. Prejdite si preto občas nasledovný zoznam, napríklad zakaždým pri riešení úloh z databázového praktika či pred rozsvičkou.

1. všeobecné

- nekorektné argumenty (napr. zámena predikátov/relácií navstívil a vypil)
- množina (relačný kalkul, datalog) vs. multimnožina (SQL, relačná algebra)

- nesprávna interpretácia kvantifikátorov:
 $\forall x$ — pre každý / ľubovoľný / hociktorý prvok x , pre všetky x
 $\exists x$ — pre nejaký / niektorý / aspoň jeden prvok x , existuje x
- zlá interpretácia zadaných relácií (napr. prehliadnutie odlišných cien alkoholu v jednotlivých krčmách)

2. relačný kalkul

- voľné (nekvantifikované) premenné, napr. Y vo výraze $\{X \mid p(X, Y)\}$
- chýbajúce či nesprávne uzátvorkovanie (logické spojky nemajú štandardne definovanú prioritu ani asociovanie ako operátory v jazyku C)
- použitie premennej mimo oblasti platnosti (napr. za zátvorkou, vnútri ktorej je kvantifikovaná)
- porovnávanie predikátov, napr. $p(X) = h(Y)$
- použitie prvkov mimo jazyka (\nexists , $\exists!$)
- miešanie prvkov datalogu ($_$)
- chýbajúce podmienky na rôznosť premenných, napr. $\exists X \exists Y (p(X) \wedge p(Y))$ (ekvivalentné $(\exists X)p(X)$)

3. datalog

- použitie pravidiel, ktoré nie sú bezpečné (naučte sa to zakaždým explicitne skontrolovať)
- nepoužitie pomocného pravidla pri negovaní na miestach, kde je to nutné, napr. $p(X) \leftarrow r(X), \neg q(X, _)$
- nevyužívanie premennej $_$ (sťažuje čítanie, znemožňuje rýchlu kontrolu, ľahšie sami seba popletiete)
- nevhodný názov pomocného predikátu (nezrozumiteľný, v rozpore s definíciou predikátu atď.)
- prebytočný argument v pomocnom predikáte — mení význam: ak popisujeme vlastnosť krčmy, musí byť táto jediným argumentom, jeden argument navyše spôsobí, že už popisujeme vlastnosť dvojice

4. SQL

- syntaktické chyby
 - `stlpec = tabulka miesto stlpec = (SELECT ... FROM tabulka)`
 - `NOT EXISTS IN miesto NOT IN; NOT IN tabulka miesto NOT IN (SELECT ... FROM tabulka)`
 - použitie agreg. funkcie na výsledok dotazu, napr. `MAX(SELECT ... FROM ...)`
 - použitie agreg. funkcie na reláciu, napr. `MAX(tabulka)`
- chyby vychádzajúce z nepochopenia významu kľúčových slov (nekorektná štruktúra dotazu)
 - vymenovanie nepovolených stĺpcov za `SELECT` (všetky musia byť aj za `GROUP BY`)
 - dvojnásobné agregovanie: `MAX(COUNT(X))`
 - použitie agregáčnej funkcie za `WHERE` (patrí hneď za `HAVING` alebo `SELECT`)
 - podmienka (najmä za `HAVING`), ktorá nevracia bool, napr. `HAVING MAX(t.x)` miesto korektného výpočtu `arg max` s dvojnásobným použitím `SELECT`
 - chybné pokusy o `arg max`: `SELECT A from t WHERE t.c = MAX(t.c)` — na nájdenie objektov, pre ktoré sa dosahuje maximum hodnoty, ktorú treba najprv vypočítať grupovaním, treba vždy aspoň dva `selecty`, napr. `SELECT A from t WHERE t.c = (SELECT MAX(t.c) FROM t)`
 - používanie `HAVING` (resp. `WHERE`) na definovanie/pomenovanie nového stĺpca — `HAVING` slúži výlučne na filtrovanie skupín, pomenovania pre stĺpce sa pridávajú za `SELECT` a agregáčnú funkciu je nutné zopakovať, napr. `SELECT COUNT(x) AS c FROM t GROUP BY y HAVING COUNT(x) > 2`
 - nepochopenie faktu, že ak `GROUP BY` chýba, ale za `SELECT` je uvedená agregáčna funkcia, bude vo výsledku jediný riadok, pretože všetko je v jednej skupine
- ďalšie chyby
 - chýbajúce podmienky pre `join` (skúste si to porátať pre každý atribút: ak do vnoreného `selectu` zvonka vstupuje nejaká hodnota povedzme pre alkohol, pridáva to 1 podmienku; ak sa tam vyskytuje ten atribút v joine povedzme `4x`, treba pridať ďalšie 3 podmienky stotožňujúce jednotlivé výskyty — porovnajte si to s datalogom, tam to funguje analogicky)
 - viacnásobné použitie relácie bez premenovania, ak sa prekrýva oblasť platnosti (scope) pre jednotlivé použitia
 - používanie pôvodného mena relácie, ak je premenovaná
 - neželané duplikáty vo výsledku (alebo za `COUNT`)
 - neúmyselný nekorelovaný subselect (vnorený dotaz, ktorého výsledok nezávisí od riadka hlavného dotazu, do ktorého je vnorený)

- nesprávne či chýbajúce použitie GROUP BY
- zámena agreg. funkcie, napr. COUNT miesto SUM

5. relačná algebra (na rozdiel od predošlých aj špecifikuje postup výpočtu)

- absencia zátvoriek, ktoré by jednoznačne definovali poradie operácií

5 Príklady (bez agregácie)

$EDB = \{\text{lubi}(P, A), \text{capuje}(K, A), \text{navstivil}(I, P, K), \text{vypil}(I, A, M)\}$

P — pijan, A — alkohol, K — krčma, M — množstvo ($M > 0$),

I — identifikátor návštevy (v navstivil sa I vyskytuje len raz

V príkladoch uvádzame správne a niekedy i nesprávne riešenia (červená farba zvýrazňuje chybu alebo označuje nesprávne riešenie ako celok).

Úloha 5.1. Alkoholy, ktoré sa čapujú, ale nikto ich nepil.

$\text{answer}(A) \leftarrow \text{capuje}(_, A), \neg \text{niektoPil}(A).$
 $\text{niektoPil}(A) \leftarrow \text{vypil}(_, A, _).$

$\text{answer}(A) \leftarrow \text{capuje}(_, A), \neg \text{vypil}(_, A, _).$

/ * nájdite 3 chyby */

$\pi_A(\text{capuje}) \triangleright \text{vypil}$ $\text{answer}(A) \leftarrow \text{capuje}(_, A), \text{niktoNepil}(A).$
 $\text{niktoNepil}(A) \leftarrow \text{vypil}(_, A_2, _), A_2 \neq A.$

Úloha 5.2. Alkoholy, ktoré ľubi každý pijan, čo niečo ľubi (a aspoň niekto).

$\text{answer}(A) \leftarrow \text{lubi}(_, A), \neg \text{pijanCoNelubi}(A).$
 $\text{pijanCoNelubi}(A) \leftarrow \text{lubi}(P, _), \text{lubi}(_, A), \neg \text{lubi}(P, A).$

$p = (\pi_P(\text{lubi}) \times \pi_A(\text{lubi})) \triangleright \text{lubi}$
 $\text{answer} = \pi_A(\text{lubi}) \triangleright p$

```
SELECT DISTINCT l.A
FROM lubi l
WHERE NOT EXISTS (
    SELECT 1 FROM lubi l2
    WHERE NOT EXISTS (
        SELECT 1 FROM lubi l3
        WHERE l3.P = l2.P AND l3.A = l.A
    )
)
```

Úloha 5.3. Alkoholy, ktoré ľubi každý pijan (ktorý niečo ľubi), čapujú ich všade (kde niečo čapujú) a niekto ich už pil.

$\left\{ A \mid (\exists I \exists M \text{vypil}(I, A, M)) \wedge \neg \left[\exists P (\exists A_2 \text{lubi}(P, A_2)) \wedge \neg \text{lubi}(P, A) \right] \right\}$
 $\wedge \neg \left[\exists K \left((\exists A_2 \text{capuje}(K, A_2)) \wedge \neg \text{capuje}(K, A) \right) \right] \right\}$

$\text{nelubenyNiekym}(A) \leftarrow \text{vypil}(_, A, _), \text{lubi}(P, _), \neg \text{lubi}(P, A).$
 $\text{necapovanyNiekde}(A) \leftarrow \text{vypil}(_, A, _), \text{capuje}(K, _), \neg \text{capuje}(K, A).$
 $\text{answer}(A) \leftarrow \text{vypil}(_, A, _), \neg \text{nelubenyNiekym}(A), \neg \text{necapovanyNiekde}(A).$

$\text{nelubenyNiekym} = (\pi_A(\text{vypil}) \times \pi_P(\text{lubi})) \triangleright \text{lubi}$
 $\text{necapovanyNiekde} = (\pi_A(\text{vypil}) \times \pi_K(\text{capuje})) \triangleright \text{capuje}$
 $\text{answer} = ((\pi_A(\text{vypil}) \triangleright \text{nelubenyNiekym}) \triangleright \text{necapovanyNiekde})$

```

SELECT DISTINCT v.A
FROM vypil v
WHERE NOT EXISTS (SELECT 1 FROM lubi l
                  WHERE NOT EXISTS (SELECT 1 FROM lubi l2
                                    WHERE l2.P = l.P AND l2.A = v.A))
AND NOT EXISTS (SELECT 1 FROM capuje c
                WHERE NOT EXISTS (SELECT 1 FROM capuje c2
                                    WHERE c2.K = c.K AND c2.A = v.A))

```

Úloha 5.4. Krčmy, ktoré navštívil každý pijan, čo niekedy pil a čapujú všetko, čo sa kedy pilo (a aspoň niečo).

$$\left\{ K \mid (\exists A \text{ capuje}(K, A)) \wedge \neg \exists P \left[\left(\exists I \exists K_2 \exists A \exists M (\text{vypil}(I, A, M) \wedge \text{navstivil}(I, P, K_2)) \right) \wedge \neg \exists I_2 \text{ navstivil}(I_2, P, K) \right] \right. \\
\left. \wedge \neg \left[\exists A \left((\exists I \exists M \text{ vypil}(I, A, M)) \wedge \neg \text{capuje}(K, A) \right) \right] \right\}$$

$\text{answer}(K) \leftarrow \text{capuje}(K, _), \neg \text{nenavstivenaNiekym}(K), \neg \text{necapujeNieco}(K).$
 $\text{necapujeNieco}(K) \leftarrow \text{capuje}(K, _), \text{vypil}(_, A, _), \neg \text{capuje}(K, A).$
 $\text{navstivilNiekedy}(P, K) \leftarrow \text{navstivil}(_, P, K).$
 $\text{nenavstivenaNiekym}(K) \leftarrow \text{capuje}(K, _), \text{navstivil}(I, P, _), \text{vypil}(I, _, _), \neg \text{navstivilNiekedy}(P, K).$

Chybné verzie nenavstivenaNiekym:

$\text{nenavstivenaNiekym}(K) \leftarrow \text{navstivil}(I, P, _), \text{vypil}(I, _, _), \neg \text{navstivil}(_, P, K), \text{capuje}(K, _).$
 $\text{nenavstivenaNiekym}(K) \leftarrow \text{navstivil}(I, P, _), \text{vypil}(I, _, _), \neg \text{navstivil}(I_2, P, K), \text{navstivil}(I_2, _, _), \text{capuje}(K, _).$

Úloha 5.5. Alkoholy, ktoré niekto ľúbi a čapuje ich každá krčma, ktorú nenavštívil Fero.

$\text{bolTam}(P, K) \iff \exists I \text{ navstivil}(I, P, K)$
 $\{A \mid \text{lubi}(_, A) \wedge \forall K (\neg \text{bolTam}(\text{fero}, K) \implies \text{capuje}(K, A))\}$
 $\{A \mid \text{lubi}(_, A) \wedge \neg \exists K (\neg \text{bolTam}(\text{fero}, K) \wedge \neg \text{capuje}(K, A))\}$
 $\text{navstivilFero}(K) \leftarrow \text{navstivil}(I, \text{fero}, K).$
 $\text{nevhodna}(A) \leftarrow \text{capuje}(K, _), \neg \text{navstivilFero}(K), \neg \text{capuje}(K, A), \text{lubi}(_, A).$
 $\text{answer}(A) \leftarrow \text{lubi}(_, A), \neg \text{nevhodna}(A).$

```

SELECT DISTINCT l.A
FROM l
WHERE NOT EXISTS (
    SELECT 1
    FROM c
    /* c.K nenavstivil Fero a necapuje l.A */
    WHERE NOT EXISTS (
        SELECT 1
        FROM n
        WHERE n.K = c.K AND n.P = 'Fero'
    ) AND NOT EXISTS (
        SELECT 1
        FROM c c2
        WHERE c2.K = c.K AND c2.A = l.A
    )
);

CREATE TEMPORARY TABLE krcmyBezFera AS (
    SELECT c.K
    FROM capuje c
    WHERE NOT EXISTS (
        SELECT 1
        FROM navstivil n
        WHERE n.K = c.K AND n.P = 'Fero'
    )
);

SELECT DISTINCT l.A
FROM l
WHERE NOT EXISTS (
    SELECT 1
    FROM krcmyBezFera kbf
    WHERE NOT EXISTS (
        SELECT 1
        FROM c
        WHERE c.K = kbf.K AND c.A = l.A
    )
);

```

Úloha 5.6. Nájdite pijanov, ktorí navštívili presne tie isté krčmy ako Ignác. Predpokladáme, že Ignác navštívil krčmu aspoň raz.

$$\begin{aligned}
 \text{answer}(P) &\leftarrow \text{navstivil}(_, P, K), \neg \text{navIgnac}(P), \neg \text{nenavIgnac}(P). \\
 \text{nav}(P, K) &\leftarrow \text{navstivil}(_, P, K). \\
 \text{navIgnac}(P) &\leftarrow \text{nav}(\text{ignac}, K), \neg \text{nav}(P, K), \text{navstivil}(_, P, _). \\
 \text{nenavIgnac}(P) &\leftarrow \text{nav}(P, K), \neg \text{nav}(\text{ignac}, K).
 \end{aligned}$$

6 Príklady (s agregáciou)

$$EDB = \{\text{lubi}(P, A), \text{capuje}(K, A, C), \text{navstivil}(I, P, K), \text{vypil}(I, A, M)\}$$

P — pijan, A — alkohol, K — krčma, C — jednotková cena, M — množstvo ($M > 0$),
 I — identifikátor návštevy (v navstivil sa I vyskytuje len raz)

Úloha 6.1. Krčmy, ktoré čapujú aspoň 5 alkoholov a nečapujú žiadne dva alkoholy za rovnakú cenu.

```

SELECT c.K
FROM capuje AS c
WHERE NOT EXISTS (SELECT 1
    FROM capuje AS c2, capuje AS c3
    WHERE c2.K = c.K AND
        c3.K = c.K AND
        c2.A <> c3.A AND
        c2.C = c3.C)

GROUP BY c.K
HAVING COUNT(c.A) >= 5;

SELECT c.K
FROM capuje AS c
GROUP BY c.K
HAVING COUNT(c.A) >= 5

EXCEPT

SELECT c1.K
FROM capuje c1, capuje c2
WHERE c1.K = c2.K AND
    c1.A <> c2.A AND
    c1.C = c2.C;

```

```

WITH K_unikatne_ceny AS (
    SELECT DISTINCT K
    FROM capuje

    EXCEPT

    SELECT K
    FROM capuje
    GROUP BY K, C
    HAVING COUNT(A) > 1

    SELECT c.K
    FROM capuje c
    GROUP BY c.K
    HAVING COUNT(c.A) >= 5
    AND COUNT(c.A) = COUNT(DISTINCT c.C);
)
SELECT c.K
FROM capuje c
JOIN K_unikatne_ceny kuc ON c.K = kuc.K
GROUP BY c.K
HAVING COUNT(c.A) >= 5;

```

Úloha 6.2. Krčmy, ktoré navštevuje jediný pijan a boli navštívené aspoň 5-krát.

```

SELECT n.K
FROM n
GROUP BY n.K
HAVING COUNT(n.I) >= 5
EXCEPT
SELECT n1.K
FROM n AS n1, n AS n2
WHERE n1.K = n2.K AND n1.P <> n2.P;

SELECT n.K
FROM n
WHERE NOT EXISTS (SELECT 1
FROM n AS n2
WHERE n2.K = n.K AND
n2.P <> n.P)
GROUP BY n.K
HAVING COUNT(n.I) >= 5;

```

```

SELECT n.K
FROM n
GROUP BY n.K
HAVING COUNT(DISTINCT n.P) = 1 AND COUNT(n.I) >= 5;

```

Úloha 6.3. Počet krčiem, ktoré čapujú aspoň dva alkoholy z tých, čo ľúbi Fero.

```

feroveOblubeneA =  $\pi$  Alkohol ( $\sigma$  lubi.Pijan = 'fero' (lubi))
c =  $\pi$  Krcma, Alkohol (capuje)
c1 =  $\rho$  Alkohol  $\rightarrow$  A1 ( $c \bowtie$  feroveOblubeneA)
c2 =  $\rho$  Alkohol  $\rightarrow$  A2 ( $c \bowtie$  feroveOblubeneA)
 $\gamma$  ; COUNT(DISTINCT Krcma)  $\rightarrow$  N ( $\pi$  Krcma ( $\sigma$  A1 <> A2 ( $c1 \bowtie c2$ )))

```

```

ferove = ( $\sigma$  P = 'Fero' (lubi))  $\bowtie$  capuje
krcmy =  $\pi$  K ( $\sigma$  p>=2 ( $\gamma$  capuje.K; COUNT(A) $\rightarrow$ p (ferove))
 $\gamma$  ; COUNT(K) (krcmy)

```

Úloha 6.4. Pijani, ktorí ľúbia najviac alkoholov spomedzi tých, čo sa nikde nečapujú (a aspoň jeden taký alkohol ľúbia).


```

WITH pocty AS (
    SELECT l.P, COUNT(l.A) AS N
    FROM lubi l
    WHERE NOT EXISTS (
        SELECT 1
        FROM capuje c
        WHERE c.A = l.A
    )
    GROUP BY l.P
)

SELECT p.P
FROM pocty p
WHERE p.N = (
    SELECT MAX(p2.N)
    FROM pocty p2
)

WITH pocetLubenychNecapovanych AS (
    SELECT l.P, COUNT(DISTINCT l.A) AS pocet
    FROM lubi l
    WHERE NOT EXISTS (
        SELECT 1 FROM capuje c
        WHERE l.A = c.A
    )
    GROUP BY l.P
)

SELECT DISTINCT pln.P
FROM pocetLubenychNecapovanych pln
WHERE NOT EXISTS (
    SELECT 1
    FROM pln pln2
    WHERE pln.pocet < pln2.pocet
);

```

$$\begin{aligned}
 p &= \Gamma_{P; \text{COUNT}(\delta A) \rightarrow \text{Pocet}}(\text{lubi} \triangleright \text{capuje}) \\
 \text{max} &= \rho_{M \rightarrow \text{Pocet}}(\Gamma_{\text{MAX}(\text{Pocet}) \rightarrow M}(p)) \\
 \text{answer} &= \pi_P(p \bowtie \text{max})
 \end{aligned}$$

Úloha 6.5. Počet pijanov, ktorí ochutnali každý alkohol čapovaný v aspoň dvoch krčmách (a niečo pili).

```

nv = navstivil ⋈ vypil
c = ρ Krcma → K, Alkohol → A (capuje)
c2 = ρ Krcma → K2, Alkohol → A2 (capuje)
alkohol_v_dvoch = π A (c) ⋈ K != K2 ∧ A = A2 (c2)
nepili_jeden_taky = π Pijan (((π Pijan (nv)) x alkohol_v_dvoch) ▷ nv)
γ ; COUNT(DISTINCT Pijan) (nv ▷ nepili_jeden_taky)

```

```

nv = navstivil ⋈ vypil
ap = γ Alkohol; count(Krcma)→Pocet (capuje)
nepili_co_mali = π Pijan (((π Pijan (nv)) x π A (σ Pocet >= 2 (ap))) ▷ nv)
γ ; COUNT(DISTINCT Pijan) (nv ▷ nepili_co_mali)

```

```

SELECT COUNT(DISTINCT n.P)
FROM navstivil n JOIN vypil v ON n.I = v.I
WHERE NOT EXISTS (
    SELECT 1
    FROM capuje c1, capuje c2
    WHERE c1.A = c2.A AND c1.K <> c2.K
    AND NOT EXISTS (
        SELECT 1
        FROM navstivil nav
        JOIN vypil vyp ON nav.I = vyp.I
        WHERE nav.P = n.P AND vyp.A = c1.A
    )
)

```

Úloha 6.6. Pod pijanmi budeme v tejto úlohe rozumieť návštevníkov krčiem. Úspešný pijan: v každej krčme, čo navštívil, čapujú niečo, čo ľúbi. Vytvorte zoznam úspešných pijanov a vypočítajte ich podiel medzi všetkými pijanmi.

```

WITH neuspesni AS (
    SELECT n.P
    FROM n
    WHERE NOT EXISTS (
        SELECT 1 FROM c
        JOIN l ON l.P = n.P AND l.A = c.A
        WHERE c.K = n.K
    )
)
CREATE TEMPORARY TABLE uspesni AS (
    SELECT DISTINCT P FROM n
    EXCEPT
    SELECT P FROM neuspesni
);
SELECT (SELECT COUNT(P) FROM uspesni)
      / (SELECT COUNT(DISTINCT P) FROM n);

CREATE TEMPORARY TABLE uspesni AS (
    SELECT DISTINCT P
    FROM n
    WHERE NOT EXISTS (
        SELECT 1
        FROM n n2
        WHERE n2.P = n.P
        AND NOT EXISTS (
            SELECT 1
            FROM c, l
            WHERE c.K = n2.K
            AND l.A = c.A
            AND l.P = n.P
        )
    )
);
SELECT COUNT(DISTINCT u.P) /
      COUNT(DISTINCT n.P)
FROM n, uspesni u;

```

Úloha 6.7. Dvojice $[K, N]$, kde K je krčma, čo niečo čapuje, a N je počet návštev K , počas ktorých boli vypité všetky alkoholy, čo K čapuje.

```

WITH UspesneNavstevy AS (
    SELECT n.K, n.I
    FROM navstivil n
    WHERE NOT EXISTS (
        SELECT 1
        FROM capuje c2
        WHERE c2.K = n.K
        AND NOT EXISTS (
            SELECT 1
            FROM vypil v
            WHERE v.I = n.I AND v.A = c2.A
        )
    )
)
SELECT
    c.K,
    COUNT(un.I) AS N
FROM
    (SELECT DISTINCT K FROM capuje) c
LEFT JOIN
    UspesneNavstevy un ON c.K = un.K
GROUP BY
    c.K;

```

Úloha 6.8. Vypočítajte celkové tržby T za predaj najobľúbenejšieho alkoholu A (t.j. takého, ktorý ľúbi najviac pijanov — ak je takých viac, počítame tržby osobitne pre každý z nich); vo výsledku dvojice $[A, T]$.

```
CREATE TEMPORARY TABLE alcohol_count AS (
    SELECT A, COUNT(P) as pocet
    FROM l
    GROUP BY l.A
);
```

```
CREATE TEMPORARY TABLE najoblub AS (
    SELECT ac.A
    FROM alcohol_count AS ac
    WHERE ac.pocet = (
        SELECT MAX(a.pocet)
        FROM alcohol_count AS a
    )
);
```

```
SELECT v.A, SUM(v.M * c.C)
FROM v JOIN n ON n.I = v.I
    JOIN c ON c.A = v.A AND c.K = n.K
    JOIN najoblub AS na ON na.A = v.A
GROUP BY v.A
```

Úloha 6.9. Pijani, ktorí už pili a pijú len alkoholy čapované za cenu nižšiu ako priemerná (cez všetky krčmy čapujúce daný alkohol).

```
priemerne_ceny =  $\gamma$  A; avg(Cena)  $\rightarrow$  PriemernaCena (capuje)
drahe =  $\pi$  K, A ( $\sigma$  Cena  $\geq$  PriemernaCena (capuje  $\bowtie$  priemerne_ceny))
( $\pi$  P (navstivil  $\bowtie$  vypil))  $\triangleright$  ((navstivil  $\bowtie$  vypil)  $\bowtie$  drahe)
```

```
WITH priemery AS (
    SELECT c.A, AVG(c.C) AS priemer
    FROM capuje c
    GROUP BY c.A
)
SELECT n1.P
FROM navstivil n1, vypil v1
WHERE n1.I = v1.I AND NOT EXISTS (
    SELECT 1
    FROM navstivil n2, vypil v2, priemery p, capuje c
    WHERE n2.I = v2.I AND c.K = n2.K AND c.A = v2.A AND n2.P = n1.P AND p.A = c.A AND c.C >= p.priemer
)
```

```
WITH priemery AS ...
SELECT DISTINCT n1.P
FROM navstivil n1, vypil v1, priemery
WHERE n1.I = v1.I AND NOT EXISTS (
    SELECT 1
    FROM navstivil n2 JOIN vypil v2, capuje c2
    WHERE n2.I = n1.I AND c2.K = n2.K AND c2.A = v2.A AND n2.P = n1.P
        AND priemery.A = v2.A AND c2.C >= priemery.priemer
);
```

Úloha 6.10. Krčmy, ktoré niečo čapujú, ale väčšina návštevníkov v nich nič neľúbi (ak pijan navštívi krčmu viackrát, rátame ho medzi návštevníkov len raz).

```

WITH lubi_v_krcme(K, P) AS (
    SELECT DISTINCT n.K, n.P
    FROM navstivil n
    JOIN capuje c ON n.K = c.K
    JOIN lubi l ON n.P = l.P AND c.A = l.A
),
vsetci_navstevnici(K, P) AS (
    SELECT DISTINCT K, P
    FROM navstivil
),
pocety AS (
    SELECT
        v.K,
        COUNT(l.P) AS pocet_lubi,
        COUNT(v.P) AS pocet_vsetci
    FROM vsetci_navstevnici v
    LEFT JOIN lubi_v_krcme l ON v.K = l.K AND v.P = l.P
    GROUP BY v.K
)
SELECT K
FROM pocety
WHERE (pocet_vsetci - pocet_lubi) > pocet_lubi;

```

Úloha 6.11. Nájdite všetkých pijanov, ktorí pili aspoň raz a nie je pravda, že by v nejakej krčme celkovo prepili viac ako v niektorých dvoch iných dokopy.

<pre> CREATE TEMPORARY TABLE prepil (SELECT n.P, n.K, SUM(v.M * c.C) AS Suma FROM navstivil n, capuje c, vypil v WHERE n.I = v.I AND c.A = v.A AND n.K = c.K GROUP BY n.P, n.K) CREATE TEMPORARY TABLE dvojice (SELECT p1.P, p1.K AS K1, p2.K AS K2, (p1.Suma + p2.Suma) AS Sucet FROM prepil p1, prepil p2 WHERE p1.P = p2.P AND p1.K != p2.K) CREATE TEMPORARY TABLE zli_pijani (SELECT DISTINCT p.P FROM prepil p, dvojice d WHERE p.P = d.P AND p.K != d.K1 AND p.K != d.K2 AND p.Suma > p.Sucet) SELECT DISTINCT n.P FROM navstivil n, vypil v WHERE n.I = v.I EXCEPT (SELECT P from zli_pijani) </pre>	<pre> CREATE TEMPORARY TABLE P_celkove_prepil_v_K AS (SELECT n.P, n.K, SUM(v.M*c.C) AS cena FROM navstivil AS n, capuje AS c, vypil AS v WHERE n.I=v.I AND n.K=c.K AND c.A=v.A GROUP BY n.P, n.K); SELECT n.P FROM n, v WHERE n.I = v.I AND NOT EXISTS (SELECT 1 FROM P_celkove_prepil_v_K AS b1, P_celkove_prepil_v_K AS b2, P_celkove_prepil_v_K AS b3 WHERE b1.cena > b2.cena + b3.cena AND b2.K <> b3.K AND b1.P = n.P AND b2.P = n.P AND b3.P = n.P) </pre>
--	--

$$\begin{aligned}
\text{answer}(P) &\leftarrow \text{navstivil}(I, P, _), \text{vypil}(I, _, _), \neg \text{mocnyUcet}(P). \\
\text{mocnyUcet}(P) &\leftarrow \text{ucet}(P, K_1, U_1), \text{ucet}(P, K_2, U_2), \text{ucet}(P, K_3, U_3), \\
&\quad K_1 \neq K_2, K_2 \neq K_3, K_1 \neq K_3, U_1 > U_2 + U_3. \\
\text{ucet}(P, K, U) &\leftarrow \text{subtotal}(\text{p}(_, P, K, _, C, M), [P, K], [U = \text{SUM}(C * M)]). \\
\text{p}(I, P, K, A, C, M) &\leftarrow \text{navstivil}(I, P, K), \text{vypil}(I, A, M), \text{capuje}(K, A, C).
\end{aligned}$$

Úloha 6.12. Krčmy, čo niečo čapujú a v ktorých je viac ako polovica tržieb dosiahnutá za alkoholy, ktoré ľubi aspoň niekto a ľubia ich všetci pijani (čo niečo ľubia) okrem práve jedného.

Najprv alkoholy, ktoré ľubi aspoň niekto a ľubia ich všetci pijani (čo niečo ľubia) okrem práve jedného.

$$\begin{aligned}
\text{alkoholyCoLubiaVsetciOkrem1}(K) &\leftarrow \text{lubi}(_, A), \text{lubi}(P, _), \neg \text{lubi}(P, A), \neg \text{niektoInyNelubi}(A, P). \\
\text{niektoInyNelubi}(A, P) &\leftarrow \text{lubi}(_, A), \text{lubi}(P, _), \text{lubi}(P2, _), P2 \neq P, \neg \text{lubi}(P2, A).
\end{aligned}$$

$$\begin{aligned}
\text{alkoholyCoLubiaVsetciOkrem1}(K) &\leftarrow \text{lubi}(_, A), \text{lubi}(P, _), \neg \text{lubi}(P, A), \neg \text{dvajaNelubia}(A). \\
\text{dvajaNelubia}(A) &\leftarrow \text{lubi}(_, A), \text{lubi}(P1, _), \text{lubi}(P2, _), P1 \neq P2, \neg \text{lubi}(P1, A), \neg \text{lubi}(P2, A).
\end{aligned}$$

```

CREATE TEMPORARY TABLE spravne_alkoholy as (
    SELECT l.A
    FROM lubi l
    WHERE EXISTS (
        SELECT p.P
        FROM lubi p
        WHERE NOT EXISTS (
            SELECT 1
            FROM lubi
            WHERE lubi.P = p.P AND lubi.A = l.A
        )
    ) AND NOT EXISTS (
        SELECT p1.P, p2.P
        FROM lubi p1, lubi p2
        WHERE p1.P <> p2.P AND NOT EXISTS (
            SELECT 1
            FROM lubi
            WHERE lubi.P = p1.P AND lubi.A = l.A
        ) AND NOT EXISTS (
            SELECT 1
            FROM lubi
            WHERE lubi.P = p2.P AND lubi.A = l.A
        )
    )
);

CREATE TEMPORARY TABLE spravne_alkoholy AS (
    SELECT a.A
    FROM lubi a, lubi p
    WHERE NOT EXISTS (
        SELECT 1
        FROM lubi l
        WHERE l.A = a.A AND l.P = p.P
    )
    GROUP BY a.A
    HAVING COUNT(DISTINCT p.P) = 1
);

```

A teraz už len zrátame tržby.

```

/* trzby za všetky alkoholy */
WITH vsetky_trzby as (
    SELECT c.K, SUM(v.M * c.C) AS trzba
    FROM capuje c, navstivil n, vypil v
    WHERE n.I = v.I AND c.K = n.K AND c.A = v.A
    GROUP BY c.K
),
/* trzby za spravne alkoholy */
spravne_trzby as (
    SELECT c.K, SUM(v.M * c.C) AS trzba
    FROM capuje c, navstivil n, vypil v
    WHERE n.I = v.I AND c.K = n.K AND c.A = v.A
    AND EXISTS (
        SELECT 1
        FROM spravne_alkoholy sa
        WHERE v.A = sa.A
    )
    GROUP BY c.K
)

SELECT c.K
FROM capuje c
WHERE EXISTS (
    SELECT 1
    FROM spravne_trzby s, vsetky_trzby v
    WHERE s.K = c.K AND v.K = c.K
    AND 2 * s.trzba > v.trzba
)

```

Podstatne stručnejšie v relačnej algebre a datalogu (premýšľajte si, kde by sa ešte dal skrátiť):

```

nelubi_prave_jeden = n Alkohol (σ N=1 (γ A; count(P)→N (((n P lubi) x (n A lubi)) ▷ lubi)))
nvc = (navstivil ⋈ vypil) ⋈ capuje
trzby_npj = γ K; Tnpj <- SUM(T_alkohol) ( n K, T_alkohol <- M * Cena ( (γ K, A, Cena; SUM(Mnozstvo)→M (nvc ⋈ nelubi_prave_jeden) ) ) )
trzby_celkovo = γ K; T <- SUM(T_alkohol) ( n K, T_alkohol <- M * Cena ( (γ K, A, Cena; SUM(Mnozstvo)→M (nvc) ) ) )
trzby_npj = n K, Tnpj ← M*Cena ( (γ K, A, Cena; SUM(Mnozstvo)→M (nvc ⋈ nelubi_prave_jeden) ) )
trzby_celkovo = n K, T ← M*Cena ( (γ K, A, Cena; SUM(Mnozstvo)→M (nvc) ) )
n K (σ Tnpj > T/2 (trzby_celkovo ⋈ trzby_npj))

krcma_group(I,K,A,C,M) :- navstivil(I,P,K), vypil(I,A,M), capuje(K,A,C).
trzby_celkovo(K,T) :- subtotal(krcma_group(_,K,_,C,M), [K], [T=SUM(C*M)]).
alkoholy_lubia_vsetci(A) :- lubi(_,A), \+zly_pijan(A, P2), lubi(P2,_), \+lubu(P2,A).
zly_pijan(A,P2) :- lubi(P,_), lubi(_,A), \+lubu(P,A), lubi(P2,_), \+P=P2.
krcma_group2(I,K,A,C,M) :- navstivil(I,P,K), vypil(I,A,M), capuje(K,A,C), alkoholy_lubia_vsetci(A).
trzby_za_alkoholy(K,T) :- subtotal(krcma_group2(_,K,_,C,M), [K], [T=SUM(C*M)]).
result(K) :- capuje(K,_,_), trzby_celkovo(K,T), trzby_za_alkoholy(K,T2), T2>(T/2).

```

Úloha 6.13. Dvojice $[K, A]$ také, že krčma K čapuje alkohol A za cenu nižšiu ako je priemerná cena A (cez všetky krčmy, čo A čapujú), a každý pijan, ktorý niekde pil A za cenu vyššiu ako v K , ho pil aj v K .

```

priemernaCenaAlkoholu(A, PC) :- subtotal(capuje(K, A, C), [A], [PC = AVG(C)]).
asponRazVypilAvK(P, A, K) :- n(Id, P, K), vypil(Id, A, _).
pijanNiekdeVypilDrahsiAleNikdyNevypilAvK(K, A, C) :- capuje(K, A, C), capuje(K2, A, C2), C2 > C,
    n(I, P, K2), vypil(I, A, _), \+asponRazVypilAvK(P, A, K).
answer(K, A) :- capuje(K, A, C), priemernaCenaAlkoholu(A, PriemernaCenaA), C < PriemernaCenaA,
    \+pijanNiekdeVypilDrahsiAleNikdyNevypilAvK(K, A, C).

```

```

WITH priemerne_ceny AS (
    SELECT A, AVG(Cena) as p
    FROM capuje
    GROUP BY A
)
SELECT c.K, c.A
FROM capuje c, priemerne_ceny p
WHERE p.A = c.A AND c.Cena < p.p AND NOT EXISTS (

```

```

/* neexistuje taky P, ktorý vypil c.A niekde drahsie a zároveň ho nevypil v c.K */
SELECT n.P
FROM navstivil n, vypil v, capuje c2
WHERE n.I = v.I AND v.A = c.A AND c2.K = n.K AND c2.A = c.A AND c2.Cena > c.Cena
      AND NOT EXISTS (
        SELECT 1 FROM vypil v2, navstivil n2
        WHERE v2.I = n2.I AND n2.P = n.P AND n2.K = c.K AND v2.A = c.A
      )
)

```

Úloha 6.14. Rozšírme záznamy návštev krčiem o začiatok a koniec návštevy, čiže $\text{navstivil}(I, P, K, Od, Do)$. Predpokladáme, že návštevy jedného pijana sa neprekrývajú ani v koncových bodoch. Ak v jednom momente pijan prichádza a iný odchádza, sú v tom momente obaja v krčme.

Pre každú navštívenú krčmu nájdite maximálny počet pijanov, ktorí v nej boli prítomní naraz.

```

-- Všetky casy, v ktorých sa niečo stalo
vc = (p Cas<-Od (n Od navstivil)) ∪ (p Cas<-Do (n Do navstivil))

-- všetky návštevy, ktoré obsahujú dany cas pre každý cas
casnavsteva = σ Cas ≥ Od and Cas ≤ Do (vc × navstivil)

caskrcmapocet = γ Cas, K; COUNT(I)→Pocet (casnavsteva)
γ K; max(Pocet)→Max (caskrcmapocet)

γ K; max(N)→M (γ T, K; count(I)→N ( (navstivil ⋈ T≥Od ∧ T≤Do (p Od→T (n Od (navstivil))))))

```

Úloha 6.15. Rozšírme záznamy návštev krčiem o začiatok a koniec návštevy, čiže $\text{navstivil}(I, P, K, Od, Do)$. Predpokladáme, že návštevy jedného pijana sa neprekrývajú ani v koncových bodoch.

V jednej krčme je kvôli koronavírusu možné mať naraz najviac 20 zákazníkov. Pre každú krčmu zistíte súčet dĺžok časových úsekov, počas ktorých bolo toto pravidlo porušené.

```

WITH okamihy AS (
  SELECT DISTINCT Cas
  FROM (
    SELECT n.Od AS Cas
    FROM navstivil n
    UNION
    SELECT n.Do AS Cas
    FROM navstivil n
  )
),
najmensie_intervaly AS (
  -- celý cas rozdelený na najkratsie zmysluplne disjunktne casove useky
  SELECT o1.Cas AS Od, o2.Cas AS Do
  FROM okamihy o1, okamihy o2
  WHERE o1.Cas < o2.Cas
  AND NOT EXISTS (
    SELECT 1
    FROM okamihy o3
    WHERE o1.Cas < o3.Cas AND o3.Cas < o2.Cas
  )
),
pocety_v_intervaloch AS (
  -- kolko ľudí bolo medzi Od-Do v Krcme
  SELECT n.Krcma, i.Od, i.Do, COUNT(n.Id) as Pocet
  FROM navstivil n, najmensie_intervaly i
  WHERE n.Od ≤ i.Od

```

```

        AND n.Do >= i.Do
        GROUP BY n.Krcma, i.Od, i.Do
    )
SELECT p.Krcma, SUM(p.Do - p.Od) AS Trvanie
FROM pocty_v_intervaloch p
WHERE p.Pocet > 20
GROUP BY p.Krcma

CREATE TEMPORARY TABLE krcma_intervaly AS (
    (
        SELECT n.K, n.Od AS Int
        FROM navstivil AS n
    )
    UNION
    (
        SELECT n.K, n.Do AS Int
        FROM navstivil AS n
    )
);

CREATE TEMPORARY TABLE krcma_ocislovane_interavaly AS(
    SELECT ki.K, ki.Int, RANK() OVER (ORDER BY ki.Int) as rank
    FROM krcma_intervaly AS ki
);

CREATE TEMPORARY TABLE krcma_zaciatok_koniec_pocet AS (
    SELECT koi1.K, koi1.Int as zac, koi2.Int as kon, COUNT(n.I) as pocet
    FROM krcma_ocislovane_intervaly koi1, krcma_ocislovane_intervali koi2 , navstivil AS n
    WHERE koi1.K = koi2.K AND koi2.K = n.K AND
        koi1.rank = koi2.rank - 1 AND
        n.Od <= koi1.Int AND n.Do >= koi2.Int
    GROUP BY koi1.K, koi1.Int, koi2.Int
);

SELECT kzkp.K, SUM(kzkp.kon - kzkp.kon)
FROM krcma_zaciatok_koniec_pocet kzkp
WHERE kzkp.pocet >=20
GROUP BY kzkp.K

navstivil_v_case(K,T,P) :- navstivil(_,P,K,Od,Do), Od < T, Do > T.
navstivilo_v_cas(K,T,C) :- subtotal(navstivil_v_case(K,T,P), [K,T], [C=count(P)]).
nad_20_v_case(K,T) :- navstivilo_v_cas(K,T,C), C > 20.
nie_je_najblizsii_odchod(K,Od,Do) :- navstivil(_,_,K,_,T), \+nad_20_v_case(K,T), Od < T, Do > T.
nad_20_cas(K,C) :- navstivil(_,_,K,Od,_), nad_20_v_case(K,Od), navstivil(_,_,K,_,Do), Od < Do, \+nad_20_v_case(K,Do).
answer(K,C) :- subtotal(nad_20_cas(K,T), [K], [C=sum(T)]).

```

7 Zadania (bez agregácie)

1. $EDB = \{osoba(A), pozna(Kto, Koho)\}$

- osoby, ktoré poznajú sysľa
- osoby, ktoré nepoznajú nikoho (žiadne iné osoby)
- osoby, ktoré majú aspoň dvoch známych (osoby)
- osoby, ktoré pozná presne jedna osoba
- osoby, ktoré poznajú iba Jožka
- osoby, ktoré poznajú všetkých známych svojich známych
- osoby, ktoré majú všetky vzťahy symetrické

2. $EDB = \{integer(X), multiply(X, Y, Result)\}$

- hodnota $5 \cdot 4$
- množina párnych čísel
- množina nepárnych čísel
- dvojice nesúdeliteľných čísel

3. $EDB = \{clovek(Meno), rieka(R), vstupil(Id_vstupu, Meno, Rieka)\}$

- ľudia, ktorí nevstúpili dvakrát do tej istej rieky
- ľudia, ktorí do aspoň jednej rieky vstúpili presne raz
- rieky, do ktorých vstupovali najviac dvaja rôzni ľudia
- dvojice $[C, R]$, kde C je človek, ktorý vstúpil do všetkých riek okrem R a do R nie

4. $EDB = \{hodnotenie(Student, Predmet, Znamka)\}$ (dvojice $[Student, Predmet]$ sú unikátne, čiže študent má z daného predmetu najviac jednu známku)

- študenti, ktorí majú aspoň 3 známky A
- študenti, ktorí boli známkovaní aspoň raz, ale nemajú Fx
- dvojice študentov, ktorí majú rovnaké známky zo všetkých predmetov, na ktorých boli obaja hodnotení
- študenti, ktorí majú z aspoň dvoch predmetov rovnaké známky ako nejaký iný študent

5. $EDB = \{blysti(Vec), zlate(Vec)\}$

- nie je všetko zlato, čo sa blyští
- veci, ktoré sú zlaté, ale neblyštia sa
- veci, ktoré ani nie sú zlaté, ani sa neblyštia

6. $EDB = \{kope(Kto, Komu, Jama), padol(Meno, Jama)\}$

- všetci, čo druhému jamu kopú, ale sami do nej padli
- tí, čo žiadnu jamu nekopú, ale do nejakej padli
- tí, čo padli do každej jamy, ktorú kopali
- jamy, kto ktorých padol každý, kto nejakú jamu kopal

7. $EDB = \{ponuka(Miesto, Akcia), chce(Clovek, Miesto, Akcia)\}$

- exkluzívne akcie, ktoré sa ponúkajú na jedinom mieste a niekto ich tam chce
- akcie, ktoré sa neponúkajú na žiadnom mieste, kde ich niekto chce, ale niekde inde áno
- akcie, ktoré sa ponúkajú všade, kde ich niekto chce
- miesta, kde sa ponúkajú všetky akcie, ktoré chce vtákokopysk

8. $EDB = \{part(Item), component(Item, Subitem)\}$

- súčiastky, ktoré sú zložené z aspoň dvoch komponentov
- atomické súčiastky (z ničoho sa neskladajú)
- všetky komponenty motora (či už atomické alebo nie)
- atomické súčiastky potrebné na zloženie televízora

9. $EDB = \{citatel(CitadelID, Meno, DatumPrihlasenia), kniha(KnihaID, Nazov, Autor),$
 $vypozička(CitadelID, KnihaID, DatumPozicania, DatumVratenia)$

(dátumy sú vyjadrené ako počet sekúnd od 1. 1. 1970; DatumVratenia je null, ak kniha nebola vrátená)

- trojice $[M, K, DatumPozicania]$, kde K je názov knihy, ktorú čitateľ s menom M ešte nevrátil
- názvy kníh, ktoré si ešte nikto nepožičal
- mená čitateľov, ktorí si už požičali všetky knihy
- mena čitateľov, ktorí vrátili všetko, čo si požičali
- dvojice $[M, K]$, kde K je názov knihy, ktorú si čitateľ M požičal práve dvakrát
- dvojice $[M, K]$, kde K je názov knihy, ktorú si čitateľ M požičal skôr ako Pipi Dlhá Pančucha

- čitatelia, ktorí si požičali všetky Haškove knihy z knižnice skôr, než niektorú z nich vrátili
- mená čitateľov, ktorí si požičali aspoň 2 rôzne knihy, pričom všetky ich požičané knihy boli od toho istého autora
- dvojice $[M, K]$, kde K je prvá požičaná kniha daného čitateľa
- mená autorov, ktorých knihu si požičal každý čitateľ (bez duplikátov)
- dvojice $[M, N]$, kde N je počet dní čitateľovej najdlhšej výpožičky (pozor na null)
- čitatelia, ktorí si nepožičali nič v deň, keď sa zaregistrovali, a naraz majú požičanú vždy nanajvýš jednu knihu
- čitatelia, ktorí si čítajú knihy „po autoroch“: ak raz precitajú knihu od nejakého autora, čítajú len knihy tohto autora, až kým neprečítajú všetky jeho knihy, ktoré sú v knižnici

10. $EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol),$
 $navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$

- krčmy, kde sa čapuje pivo a nič iné
- krčmy, kde sa pije pivo a nič iné
- pijani, ktorí ľúbia len rum
- pijani, ktorí ľúbia práve jeden alkohol
- pijani-štamgasti, ktorí doteraz navštívili jedinu krčmu
- pijani, ktorí niekedy niekde vypili viac ako pol litra jedného alkoholu
- pijani, ktorí nikdy neodolali rumu (pili ho pri každej návšteve krčmy, v ktorej ho čapujú)
- pijani, ktorí ľúbia aspoň niečo a ľúbia každý alkohol, ktorý je čapovaný v aspoň dvoch krčmách
- pijani, ktorí v aspoň dvoch krčmách ľúbia všetky tam čapované alkoholy
- pijani, ktorí ľúbia aspoň jeden taký alkohol, ktorý čapuje každá krčma, v ktorej ten pijan niečo vypil
- pijani, ktorí pili rum a pili ho v každej krčme, kde ho čapujú, s výnimkou najviac jednej
- pijani, ktorí pri niektorej svojej návšteve krčmy vytvorili doteraz platný rekord v pití vodky v danej krčme
- alkoholy, ktoré ľúbi každý, čo niečo ľúbi (a aspoň niekto), ale nepili sa v každej krčme, čo niečo čapuje a kde sa niečo pilo
- pijanov, ktorí vypili len tie alkoholy, ktoré vypil pijan Felix (t.j. hľadanie pijani vypili nejakú neprázdnu podmnožinu alkoholov, ktoré vypil pijan Felix; a okrem tých alkoholov nevypili žiadne iné)
- alkoholy, pre ktoré platí, že ak ten alkohol niektorý pijan niekedy vypil, tak ho ten pijan vypil pri každej svojej návšteve krčmy (vo výsledku majú byť aj alkoholy, ktoré nikto nikdy nevypil)
- krčmy, pre ktoré platí: hľadanie krčmu nenavštívil žiaden pijan, ktorý ľúbi všetky alkoholy, ktoré tá krčma čapuje (predpokladajte, že každá krčma čapuje nejaký alkohol)
- všetkých takých pijanov, ktorí neľúbia pivo ani borovičku; a zároveň sa dôsledne vyhýbajú návštevám takým krčmám, v ktorých sa čapuje len pivo alebo borovička; a zároveň nikdy pivo ani borovičku nevypili
- alkoholy, ktoré ľúbia len tí pijani, ktorí nikdy nenavštívili krčmu Wasa
- pijani, ktorí z každého alkoholu A , čo ľúbia, v aspoň jednej krčme aspoň raz vypili na jedno posedenie viac, ako hocikto iný, čo tam niečo pil
- dvojice $[P, K]$ také, že pijan P pri každej návšteve krčmy K vypil niektorý z alkoholov, ktoré ľúbi (pri rôznych návštevách mohol vypiť rôzne obľúbené alkoholy, chceme len dvojice, kde P niekedy navštívil K)
- dvojice $[P, A]$, ktoré hovoria, ktoré alkoholy A pijan P vypil pri každej svojej návšteve krčmy (abstinenti nemajú byť vo výsledku)
- dvojice $[P, A]$ také, že pijan P ľúbi alkohol A , a zároveň každá krčma, v ktorej P vypil A , čapuje alkohol A lacnejšie než ktorákoľvek iná krčma, ktorá čapuje A
- dvojice $[P, A]$ také, že pijan P ľúbi alkohol A a ešte také dva ďalšie (navzájom rôzne) alkoholy, že pri každej návšteve krčmy, pri ktorej P vypil A , vypil aj niektorý z týchto dvoch ďalších alkoholov
- dvojice $[A, K]$ také, že alkohol A čapovaný v krčme K vypil (pri aspoň jednej návšteve) každý pijan, ktorý K niekedy navštívil

- dvojice $[P, A]$ také, že pijan P líbí alkohol A ; a v každé krčmě, která čapuje alkohol A , vypil P počas niektorej návštevy viacej alkoholu A než ktorýkoľvek iný pijan počas jednej návštevy (teda P je rekordérom v pití A na jedno posedenie v každej krčme, ktorá A čapuje)
- pijanov, ktorí každý akt vypitia alkoholu urobili v jednej z krčiem, kde je ten alkohol najlacnejší (abstinenti nemajú byť vo výsledku)
- dvojice $[K, A]$ také, že krčma K čapuje alkohol A , a zároveň každý pijan, ktorý líbí alkohol A , ho vypil pri niektorej návšteve krčmy K (čiže tú krčmu aj navštívil)

8 Zadania (s agregáciou)

1. $EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol, Cena), navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$
 - pijani, ktorí ľúbia aspoň 10 rôznych alkoholov
 - alkoholy, ktoré boli vypité v krčme Stein v celkovom množstve väčšom ako 20
 - dvojice $[A, Suma]$, ktoré popisujú množstvo alkoholu A vypitého v krčme Carlton (vo výsledku len tie, čo sa niekedy pili)
 - dvojice $[P, Pocet]$, ktoré hovoria, v koľkých krčmách prepil pijan P aspoň 10 EUR počas niektorej (jednej) návštevy
 - trojice $[P, A, Pocet]$, ktoré hovoria, pri koľkých návštevách pijan P vypil alkohol A (netreba nájsť trojice s počtom 0)
 - pijani, ktorí sú v niektorej krčme lokálnymi šampiónmi v pití rumu (t.j. hľadaný pijan v aspoň jednej krčme vypil dokopy viacej rumu než ľubovoľný iný pijan)
 - dvojice $[P, Suma]$, ktoré hovoria, koľko peňazí pijan P celkovo prepil v krčmách, ktoré čapujú len alkoholy, ktoré P neľúbi (dvojice s nulovou sumou nemajú byť vo výsledku)
 - trojice $[P, nK, nA]$, kde nK je počet rôznych krčiem, ktoré pijan P navštívil, a nA je počet rôznych alkoholov, ktoré pijan P vypil (nechceme trojice, kde $nK = nA = 0$)
 - trojice $[K, A, m]$, kde m je celkové množstvo alkoholu A vypitého v krčme K (chceme vo výsledku každú dvojicu K, A , kde K čapuje A)
 - dvojice $[K, A]$ také, že alkohol A sa v krčme K vypil v celkovom množstve väčšom ako 50
 - trojice $[P, K, Pocet]$, ktoré hovoria, pri koľkých návštevách pijan P vypil v krčme K aspoň 5 borovičiek na jedno posedenie (trojice s nulovým počtom nás nezaujmajú)
 - dvojice $[A, M]$ také, že M je mediánom ceny alkoholu A cez všetky krčmy, ktoré alkohol A čapujú
 - dvojice $[K, Suma]$, ktoré hovoria, koľko peňazí v krčme K celkovo prepili pijani, ktorí tú krčmu navštívili viac než stokrát (dvojice s nulovou sumou nemajú byť vo výsledku)
 - trojice $[P, K, Priemer]$ ktoré hovoria, koľko peňazí utratil pijan P v priemere pri jednej návšteve krčmy K (trojice s nulovým priemerom nemajú byť vo výsledku)
 - všetky dvojice $[K, R]$, kde K je krčma, ktorú niekto navštívil, a $R \in [0, 1]$ je podiel sklamaných pijanov, čiže podiel počtu pijanov, ktorí K navštívili, ale neľúbia žiaden alkohol, ktorý K čapuje, k celkovému počtu pijanov, ktorí K navštívili