

1 Relačný kalkul

Všetky dotazy so všeob. kvantifikátorom následne prepísať cez existenčný.

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 majú všetky vzťahy symetrické
- osoby, ktoré medzi Ferovými známymi poznajú aspoň jedného, ale nie všetkých
- osoby, ktoré poznajú všetkých známych svojich známych
- osoby, ktoré poznajú aspoň jedného známeho každého svojho známeho

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

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

2 Relačný kalkul bez všeob. kvantifikátora, datalog

Syntax a sémantika datalogu (implicitné existenčné kvantifikátory). Význam premennej $_$.

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

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

2. $EDB = \{hodnotenie(Student, Predmet, Znamka)\}$

- študenti, ktorí majú aspoň 3 známky A
- študenti, ktorí boli hodnotení aspoň raz, ale nemajú Fx
- predmety, z ktorých bol každý študent hodnotený viac ako raz (a aspoň jeden študent bol hodnotený)

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

- ľudia, ktorí nevstúpili dvakrát do tej istej rieky
- rieky, kam vstúpil každý, kto do nejakej rieky vstúpil (a vstúpil do nich aspoň niekto)
- ľudia, ktorí vstúpili do každej rieky presne dvakrát

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

- 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

5. $EDB = \{politik(Meno), cestny(Meno), podplatil(Kto, Koho, Uplatok)\}$

- všetci politici sú nečestní
- niektorí nečestní sú politikmi
- politici, ktorí nikdy neprijali uplatok
- politici, ktorí síce nie sú čestní, ale prijímajú len úplatky do 100

(a) $EDB = \{kozmonaut(Meno, Vek, Stat, Lod)\}$

- lode, ktoré používal Armstrong alebo Gagarin
- mená a veki kozmonautov, ktorí použili aspoň dve lode
- štáty, ktoré majú práve jedného kozmonauta

3 Datalog

6. $EDB = \{part(Name), component(Item, Subitem, AttachmentType)\}$

- 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
- súčiastky, ktoré sú priskrutkované k iným

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

- krčmy, kde sa čapuje pivo a nič iné
- krčmy, kde sa pije pivo a nič iné
- pijani, ktorí ľúbia rum a práve jeden iný alkohol
- pijani-štamgasti, ktorí doteraz pili v jedinej krčme
- 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 jeden taký alkohol, ktorý čapuje každá krčma, v ktorej ten pijan niečo vypil
- pijani, ktorí pri niektorej svojej návšteve krčmy vytvorili doteraz platný rekord v pití vodky v danej krčme
- pijanov, ktorí vypili len tie alkoholy, ktoré vypil pijan Felix (t.j. hľadani 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: hľadanú 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 krčiem, 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
- dvojice $[P, K]$ také, že pijan P pri každej návšteve krčmy K vypil niektorý z alkoholov, ktoré ľúbi (pri rôznych návštevách mohol vypíť 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 $[K, A]$ také, že krčma K čapuje alkohol A , a zároveň každý pijan, ktorý ľúbi alkohol A , ho vypil pri niektorej návšteve krčmy K
- 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 ľúbi alkohol A ; a v každej krčme, ktorá 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)

4 cvičenie: datalog, SQL

1. **Rozcvička.** $EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol), navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$
 - datalog
 - dvojice $[K, A]$ také, že krčma K čapuje alkohol A , a zároveň každý pijan, ktorý ľúbi alkohol A , ho vypil pri niektorej návšteve krčmy K (čiže tú krčmu aj navštívil)
2. SQL; $EDB = \{hodnotenie(Student, Predmet, Znamka)\}$
 - študenti, ktorí dostali aspoň jedno A
 - študenti, ktorí dostali aspoň jedno A a nedostali Fx
 - študenti, ktorí dostali A aspoň z troch predmetov [využiť agregáciu]
3. úloha z rozcvičky
 - zapísať datalog (dobrovoľník nejaký)
 - strojový preklad z datalogu do SQL vysvetliť (ja)
4. SQL $EDB = \{kope(Kto, Komu, Jama), padol(Meno, Jama)\}$
 - 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

5 cvičenie: SQL, relačná algebra

1. **Rozcvička.** $EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol), navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$
 - datalog
 - krčmy K , ktoré navštívil aspoň jeden pijan, ktorý ľúbi všetky alkoholy, ktoré K čapuje (predpokladajte, že každá krčma čapuje nejaký alkohol)
2. $EDB = \{osoba(A), pozna(Kto, Koho)\}$
 - osoby, ktoré poznajú sysľa
 - osoby, ktoré nepoznajú nikoho a nič (nielen osoby)
 - osoby, ktoré majú aspoň štyri známe osoby
 - osoby, ktoré pozná presne jedna entita
 - osoby, ktoré poznajú iba Jožka
 - osoby, ktoré poznajú všetkých známych svojich známych
3. vyriešiť rozcvičku (datalog, SQL, algebra)
Relačná algebra; nad multimnožinami:
 - \cup zjednotenie
 - \cap prienik
 - $-$ rozdiel
 - ρ premenovanie (stĺpcov)
 - \times karteziánsky súčin (počet stĺpcov vo výsledku je súčtom počtov stĺpcov operandov)
 - $\pi_X(r)$ projekcia; X je množina atribútov, ktoré ostanú, ostatné sa zahodia
 - $\sigma_{c(X)}(r)$ selekcia; vyberie riadky, ktoré spĺňajú podmienku $c(X)$
 - $r_1 \bowtie r_2$ natural join
 - $r_1 \bowtie_c r_2$ theta-join = selekcia aplikovaná na karteziánsky súčin
 - δ eliminácia duplikátov
 - Γ grupovanie a agregácia, napr. $\Gamma_{A,B,AVG(C)}(r)$
 - T triedenie
 - *OUTERJOIN*

zápisy: jediný výraz / postupné priradzovanie do pomenovaných relácií / strom výpočtu

6 cvičenie: Datalog, SQL, relačná algebra

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

- relačná algebra, SQL
- alkoholy, ktoré čapuje krčma Stein, ale nikdy ich tam nik nepil

2. agregácia pijanov: datalog, SQL $EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol, Cena), navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$

- 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)

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 $nK = nA = 0$)

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)

3. agregácia pijanov: datalog, SQL $EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol, Cena), navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$

- 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 nezaujímajú)
- 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)

7 cvičenie: Datalog, SQL, relačná algebra

1. Rozcvička. $EDB = \{capuje(Krcma, Alkohol)\}$

- SQL
- dvojice $[K, n]$ také, že v krčme K sa čapuje n alkoholov s cenou vyššou ako je priemerná cena (cez všetky krčmy, ktoré ich čapujú)

2. agregácia pijanov: datalog, $SQL\ EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol, Cena), navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$

- trojice $[P, A, Pocet]$, ktoré hovoria, pri koľkých návštevách pijan P vypil alkohol A (netreba nájsť trojice s počtom 0)
- 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)
- pijani, ktorí sú v niektorej krčme lokálnymi šampiónmi v pití rumu (t.j. hľadaný pijan v aspoň jednej krčme pil aspoň raz a vypil dokopy aspoň toľko rumu ako ľubovoľný iný pijan)
- pijani, ktorí sú v každej krčme lokálnymi šampiónmi v pití rumu (t.j. hľadaný pijan v aspoň jednej krčme pil aspoň raz a vypil dokopy aspoň toľko rumu ako 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)
- 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
- $[P, nK, nA]$, kde nK je počet rôznych krčiem, ktoré pijan P navštívil, a nA je počet rôznych alkoholov, ktoré pijan P vypil (nechceme trojice, kde $nK = nA = 0$)
- $[K, A, m]$, kde m je celkové množstvo alkoholu A vypitého v krčme K (chceme vo výsledku každú dvojicu K, A , kde K čapuje A)

3. agregácia pijanov: datalog, $SQL\ EDB = \{lubi(Pijan, Alkohol), capuje(Krcma, Alkohol, Cena), navstivil(Id, Pijan, Krcma), vypil(Id, Alkohol, Mnozstvo)\}$

- 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 nezaujímajú)
- 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)

4. pojmy:

- funkčná závislosť $X \rightarrow Y$

5. nájdite funkčné závislosti v reláciách:

(meno, rodné číslo, ulica, mesto, kraj, PSČ, telefónne číslo)
(moleculeId, x, y, z, vx, vy, vz)

8 cvičenie: funkčné závislosti, kľúče

- Rozcvička.** $EDB = \{lubi(P, A), capuje(K, A), navstivil(Id, P, K), vypil(Id, A, Mnozstvo)\}$
 - datalog, SQL
 - 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
- Armstrongove axiomy
 - $X \subseteq Y \implies Y \rightarrow X$
 - $X \rightarrow Y \implies XZ \rightarrow YZ$
 - $X \rightarrow Y \wedge Y \rightarrow Z \implies X \rightarrow Z$
- uzáver X^+ množiny atribútov X vzhľadom na množinu funkčných závislostí F : množina všetkých atribútov Y takých, že $X \rightarrow Y$ sa dá odvodiť z F
 - algoritmus: vezmeme $X^+ := X$ a postupne prechádzame všetky závislosti v F ; ak závislosť odvodí nové atribúty, pridáme ich do X^+ a závislosť zahodíme (nič nové sa z nej už nebude dať odvodiť); ak sa už nedá použiť žiadna zo zvyšných závislostí, končíme
- Nájdite $\{EF\}^+$ vzhľadom na $\mathcal{F} = \{BE \rightarrow GH, BEG \rightarrow F, AD \rightarrow C, F \rightarrow B, BF \rightarrow A\}$.
- uzáver \mathcal{F}^+ množiny funkčných závislostí \mathcal{F} je množina funkčných závislostí, ktoré sú dôsledkom \mathcal{F} , t.j. dajú sa odvodiť pomocou Armstrongových axiém
 - stačí uvádzať *maximálne* funkčné závislosti, t.j. kde sa nedá vynechať atribút z ľavej strany ani pridať atribút na pravú stranu bez porušenia platnosti
- množina funkčných závislostí \mathcal{G} *pokrýva* množ. funkčných závislostí \mathcal{F} , ak $\mathcal{F}^+ \subseteq \mathcal{G}^+$
 - algoritmus: otestujeme, či každú závislosť v \mathcal{F} vieme odvodiť z \mathcal{G}
 - test pre jednu závislosť: overíme, či pravá strana závislosti je podmnožinou uzáveru ľavej strany vzhľadom na \mathcal{G}
- minimálne pokrytie množiny funkčných závislostí \mathcal{F} je množina kanonických funkčných závislostí \mathcal{G} taká, že \mathcal{F} a \mathcal{G} sa navzájom pokrývajú a pri vynechaní ľubovoľnej závislosti z \mathcal{G} alebo ľubovoľného atribútu z ľavej strany takej závislosti už \mathcal{G} prestane byť pokrytím \mathcal{F}
 - *kanonická* funkčná závislosť: na pravej strane práve jeden atribút
 - minimálnych pokrytí môže existovať viac
 - algoritmus:
 - rozbitie pravých strán (nahradenie $X \rightarrow Y$ množinou $\{X \rightarrow A, A \in Y\}$)
 - vynechanie redundantných atribútov na ľavých stranách (každý atribút z ľavej strany otestovať raz, počítame uzáver ľavej strany s vynechaným atribútom vzhľadom na F)
 - vynechanie redundantných závislostí (vezmeme ľavú stranu závislosti α a počítame jej uzáver vzhľadom na $F - \{\alpha\}$, ak obsahuje pravú stranu, závislosť možno vynechať)
- Nájdite minimálne pokrytie množiny $\mathcal{F} = \{BE \rightarrow GH, BEG \rightarrow F, AD \rightarrow C, F \rightarrow B, BF \rightarrow A\}$ pre reláciu $r(A, B, C, D, E, F, G)$.
- nech r je relácia nad množinou atribútov U ; *nadkľúč* je množina atribútov K taká, že $K \rightarrow U$; *kľúč* je nadkľúč minimálny vzhľadom na inklúziu
 - algoritmus hľadania kľúčov zhora nadol
 - vezmeme najprv množinu všetkých atribútov a pokúšame sa z nej postupne odstrániť jednotlivé atribúty
 - podčiarknutý atribút: nachádza sa v každom kľúči
 - atribúty, ktoré nie sú na pravej strane žiadnej závislosti, sú automaticky podčiarknuté

- (d) atribúty, ktoré nie sú na ľavej strane žiadnej závislosti, ale sú na pravej strane, môžeme úplne vynechať, lebo nebudú v žiadnom kľúči
- (e) kreslíme strom (pre atribút X , ktorý nie je podčiarknutý: ľavá vetva neobsahuje X , pravá obsahuje X)
- (f) pri prehľadávaní:
- Pred prehľadávaním stromu vypočítaj uzáver z množiny podčiarknutých atribútov — ak sú podčiarknuté atribúty nadkľúčom, tak to je kľúč (ukončí vetvu)
 - V ľavej vetve over, či uzáver naďalej obsahuje ten atribút, ktorý sa v tej vetve vynecháva (ak nie, ukončí vetvu)
 - Po nájdení nejakého kľúča neprehľadávaj podstromy, ktoré ten kľúč obsahujú
10. Nájdite všetky nadkľúče pre relačnú schému $r(A, B, C, D, E, F, G)$, $\mathcal{F} = \{ABCD \rightarrow EF, ABE \rightarrow FG, ABDG \rightarrow CF, G \rightarrow BD\}$.
11. Uvažujme relačnú schému $r(A, B, C, D, E)$, $\mathcal{F} = \{A \rightarrow CE, ACD \rightarrow BE, BC \rightarrow D, BE \rightarrow AC\}$.
- (a) Nájdite minimálne pokrytie \mathcal{F} .
- (b) Nájdite všetky kľúče r .
12. medián v SQL a datalogu: dvojice $[P, m]$, kde m je medián počtu čapovaných alkoholov, ktoré pijan P ľúbi (cez všetky krčmy)

9 cvičenie: funkčné závislosti, kľúče, dekompozícia do normálnych foriem

- **Rozcvička.** Dokážte, že $\mathcal{F} = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, BI \rightarrow J\}$ je pokrytím $\mathcal{G} = \{A \rightarrow BCE, AB \rightarrow DE, BI \rightarrow J\}$. Rozhodnite, či existuje $\varphi \in \mathcal{F}$ tak, že $\mathcal{G}^+ \not\subseteq (\mathcal{F} \setminus \{\varphi\})^+$.
- Relačná schéma (r, F) je v *tretej normálnej forme* (3NF), ak pre každú platnú netriviálnu (takú, že ľavá strana sa nedá skratiť) funkčnú závislosť $X \rightarrow Y$ platí, že buď X je nadkľúč v r alebo Y je časťou nejakého kľúča v r .
- Relačná schéma (r, F) je v *Boyce-Coddovej normálnej forme* (BCNF), ak pre každú platnú netriviálnu funkčnú závislosť $X \rightarrow Y$ platí, že X je nadkľúč v r .
- Dekompozícia do 3NF zachovávalajúca funkčné závislosti

Vstup: Relačná schéma r a minimálne pokrytie závislostí F

Výstup: Relačné schémy bestratovej dekompozície do 3NF, pričom všetky funkčné závislosti ostanú zachované

Dekompozícia do 3NF zachovávalajúca funkčné závislosti:

Ak F obsahuje závislosť, ktorá obsahuje všetky atribúty r , potom r je už v 3NF

Inak každej funkčnej závislosti v F zodpovedá jedna relačná schéma

Ak je niektorá podschéma podmnožinou inej, treba ju vynechať

Ak žiadna z tých relačných podschém neobsahuje kľúč r , treba ešte pridať schému s nejakým kľúčom. (Testovanie, či podschéma obsahuje nejaký kľúč r , nevyžaduje výpočet kľúčov. Stačí overiť, či atribúty niektorej podschémy sú nadkľúčom v r . Ak áno, tak tá podschéma obsahuje nejaký kľúč r , inak neobsahuje.)

- Algoritmus dekompozície relácie R do BCNF (je prirodzené začať s dobrou 3NF dekompozíciou):
 1. Dekomponuj r do 3NF so zachovaním funkčných závislostí.
 2. Over, či každá relácia dekompozície je v BCNF. Ak nie, nájdi v nej funkčnú závislosť $X \rightarrow Y$ ktorá porušuje BCNF a dekomponuj reláciu r_i do dvoch relácií, $r_i - Y$ a XY .
 3. Opakuj overovanie a rozkladanie až kým sú všetky relácie v BCNF.

- chase 1. Vytvor maticu S s 1 riadkom pre každú podreláciu r_1, \dots, r_m a s 1 stĺpcom pre každý atribút r
- 2. Nastav počiatočné hodnoty, $S[i, j] := b_{i,j}$
- 3. for ($i = 1 \dots m$) for ($j = 1 \dots n$) if (atribút A_j patrí do r_i) then $S[i, j] := a_j$
- 4. Opakuj kým matica mení: for all ($X \rightarrow Y \in F$) for (všetky dvojice riadkov s rovnakými symbolmi v tých stĺpcoch, ktoré zodpovedajú atribútom v X) Zjednoť riadky v tých 2 riadkoch, pritom preferuj symboly a^* .
- 5. Ak jeden riadok obsahuje len symboly a^* , rozklad je bezstratový, inak nie.
- Pre danú relačnú schému r a množinu funkčných závislostí \mathcal{F} :
 - (a) Nájdite všetky kľúče r .
 - (b) Nájdite minimálne pokrytie \mathcal{F} .
 - (c) Zo zostrojeného minimálneho pokrytia vytvorte dekompozíciu do 3NF zachovávajúcu závislosti.
 - (d) Použite chase a dokážte, že vytvorená dekompozícia sa spája bezstratovo.
 - (e) Rozhodnite, či je vytvorená dekompozícia v BCNF.
 1. TODO preratat alebo vymeniť zadanie
 2. $r(A, B, C, D, E, F, G)$, $\mathcal{F} = \{AB \rightarrow C, ACEF \rightarrow G, BC \rightarrow AD, D \rightarrow E, DE \rightarrow F, DF \rightarrow C, F \rightarrow DE, G \rightarrow F\}$; [kluce: AB, BC, BD, BF, BG]
 3. $r(A, B, C, D, E, F, G, H)$ s funkčnými závislosťami $BE \rightarrow GH, BEG \rightarrow FA, D \rightarrow C, F \rightarrow B, BF \rightarrow A$.
 4. $r(A, B, C, D, E, F, G, H)$, $\mathcal{F} = \{BF \rightarrow ACG, C \rightarrow AE, AH \rightarrow F, AF \rightarrow H, BG \rightarrow F, E \rightarrow G, BCE \rightarrow F, GH \rightarrow AF, H \rightarrow D\}$

10 cvičenie: funkčné závislosti, kľúče, dekompozícia do normálnych foriem

- mal plachetka

11 cvičenie: transakcie

- rozcvička: nájdite minimálne pokrytie a zostrojte z neho bezstratovú dekompozíciu do 3NF, ktorá zachováva funkčné závislosti:
 $r(A, B, C, D, E, F, G, H)$ s funkčnými závislosťami $BE \rightarrow GH, BEG \rightarrow FA, D \rightarrow C, F \rightarrow B, BF \rightarrow A$.
- v rozvrhu sú dve operácie *konfliktné*, ak patria rôznym transakciám, ich operandom je rovnaký objekt a aspoň jedna z tých operácií je write
 - dva rozvrhy sú konflikt-ekvivalentné práve vtedy, ak pozostávajú z rovnakých operácií a relatívne poradie každých dvoch konfliktných operácií je rovnaké v oboch históriách
 - rozvrh je *sériový* práve vtedy, ak je kompletný (t.j. každá transakcia v tom rozvrhu končí commitom alebo abortom) a pre každú dvojicu transakcií T1, T2 platí, že buď všetky operácie T1 v tom rozvrhu predchádzajú operáciám T2 alebo naopak
 - rozvrh je *konflikt-sériovateľný* práve vtedy, ak jeho projekcia na commitované transakcie je konflikt-ekvivalentná niektorému sériovému rozvrhu tých commitovaných transakcií
 - Rozvrh je *konflikt-sériovateľný* práve vtedy, ak jeho precedenčný graf je acyklický. Topologické usporiadanie jeho precedenčného grafu hovorí ktorému sériovému rozvrhu je ten rozvrh ekvivalentný.
- rozhodnite, či sú rozvrhy konflikt-sériovateľné
 - $r1(X), w1(X), r3(Y), r2(X), r1(Y), w2(Y), w3(X)$
 - $r1(X), w1(X), r2(Y), w2(Y), r3(Y), r2(X), r1(Y), w3(X)$
- transakcia T2 číta X od transakcie T1, ak v tom rozvrhu existuje operácia $w1(X)$ a neskôr operácia $r2(X)$, pričom T1 je v čase operácie $r2(X)$ stále aktívna (neukončená)
 - dva rozvrhy H a H' sú *view-ekvivalentné*, ak (sú definované nad tými istými transakciami a zároveň) (1) pre každú dvojicu operácií v H, kde nejaká transakcia T1 číta X od T2 existuje taká istá dvojica operácií v H', kde T1 tiež číta X od T2, a zároveň (2) pre každý dátový objekt X, ak transakcia Ti je posledná transakcia ktorá píše do X v H, tak aj v H' je Ti posledná transakcia ktorá píše do X (final write) (Intuitívne, rozvrhy sú view-ekvivalentné, ak majú rovnaký efekt.)
 - rozvrh je *view-sériovateľný*, ak commitovaná projekcia každého jeho prefixu je view-ekvivalentná niektorému sériovému rozvrhu
- rozhodnite, či nasledujúce rozvrhy sú konflikt-sériovateľný a view-sériovateľný
 - $r3(b), w3(b), w4(b), r2(b), r1(a), r1(c), w1(a), w1(c), r3(a), w3(c), r2(a), w3(c)$
 - $r1(a), r1(b), w1(a), r3(a), r2(b), w3(c), r2(c), w2(b), r1(a), w3(a), w2(c), w2(a)$
 - $r3(Z), w3(X), w1(Z), w2(X), r2(Z), r2(Y), w1(X), w3(Z), w1(Y), c1, r3(X), c2, c3$
- dvojfázové zamykanie: transakcia musí na čítanie i písanie vlastniť zámok; po uvoľnení nejakého zámku už nesmie o iný požiadať; read lock môže mať viac transakcií zároveň, write lock je úplne exkluzívny
- vytvorte rozvrh pre nasledujúce transakcie, overte, že je konflikt-sériovateľný, a nájdite ekvivalentný sériový rozvrh: T1: $r(X), w(X), r(Y), c$ T2: $r(Y), w(Y), r(X), c$ T3: $r(Y), w(X), c$
- Uveďte príklad rozvrhu, v ktorom všetky transakcie končia commitom, a ktorý je konflikt-sériovateľný a nedá sa generovať dvojfázovým zamykaním $[r1(X), r2(X), w1(X), c1, c2]$