Adatbázisok elmélete 15. előadás

Katona Gyula Y.
Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudományi Tsz.
I. B. 137/b
kiskat@cs.bme.hu
http://www.cs.bme.hu/˜kiskat

2004

Többértékű függés

A legfontosabb a funkcionális függés, de vannak másféle függések is.

Motiváló példa: R(Név, Tantárgy, Gyereknév)

Név	Tantárgy	Gyereknév
Katona	Algel	Dani
Katona	Adatbázis	Lilla
Katona	Algel	Lilla
Katona	Adatbázis	Dani

Ez BCNF, de mégis redundáns, mert ha valamelyik tárgynál szerepel egy gyereknév, akkor az összes többinél is szerepelnie kell. (Pl. beszúrni nehéz, mert amikor egy sort beszúrok, figyelni kell arra, hogy egy másikat is beszúrjak.)
Jobb lenne tárolni (Név, Tantárgy) és (Név, Gyereknév) felbontásban.

Ok: a Tantárgy és a Gyereknév független (minden kombinációban előfordulnak) $\Longrightarrow$ ha látjuk az első két sort, tudjuk, hogy a másik kettő is ott van.

Többértékű függés

Definíció. *Az X attribútumhalmaztól* ***többértékűen függ*** *az Y attribútumhalmaz az r relációban (jele: $X \twoheadrightarrow Y$), ha tetszőleges $t_1, t_2 \in r$ sorokra, melyekre $t_1[X] = t_2[X]$, létezik $t_3, t_4 \in r$, melyekre*

- $t_3[XY] = t_1[XY]$
- $t_3[R \setminus XY] = t_2[R \setminus XY]$
- $t_4[XY] = t_2[XY]$
- $t_4[R \setminus XY] = t_1[R \setminus XY]$

	$\overbrace{\dots\dots\dots}^{X}$	$\overbrace{\dots\dots\dots}^{Y}$	$\overbrace{\dots\dots\dots}^{R \setminus XY}$
t_1	AAAAAAA	BBBBBBB	CCCCCCC
t_2	AAAAAAA	DDDDDDD	EEEEEEE
	⋮	⋮	⋮
t_3	AAAAAAA	BBBBBBB	EEEEEEE
t_4	AAAAAAA	DDDDDDD	CCCCCCC

Megjegyzés: A funkcionális függőség **egyenlőséggeneráló**. Ha két dolog egyenlő, akkor másik két dolog is egyenlő lesz. A többértékű függőség **sorgeneráló**. Ha van két sor ami valahol egyenlő, akkor vannak más sorok is.
Az előbbi példában: Név↠ Tantárgy, Név↠ Gyereknév

Többértékű függések levezetése

Definíció. ***Triviális többértékű függések*** *(amik mindig igazak):*

- $Y \subseteq X \Longrightarrow X \twoheadrightarrow Y$, *mert $t_3 = t_2$ és $t_4 = t_1$ jó lesz.*
- $XY = R \Longrightarrow X \twoheadrightarrow Y$, *mert $t_3 = t_1$ és $t_4 = t_2$ jó lesz.*

Ezentúl a többértékű függések is a séma részei lesznek és definiálhatjuk a levezethetőséget ($\vdash$) és a logikai következményt ($\models$) úgy, hogy funkcionális függőségek és többértékű függőségek is vannak F-ben.

Logikai következmény: egy F (funkcionális és többértékű függéseket is tartalmazó) függéshalmaznak logikai következménye egy (funkcionális vagy többértékű) függés, ha minden olyan relációban, amiben F minden függése fennáll, fenn kell hogy álljon a mondott függés is.

Levezetés: Armstrong-axiómák (a funkcionális függésekre) és 5 új axióma, amiben $\rightarrow$ és $\twoheadrightarrow$ is van. Amilyen függés ezekkel előáll F-ből, arra mondjuk, hogy levezethető.

Hasonló elmélet, mint $\rightarrow$-nél $\Longrightarrow$ belátható, hogy $\vdash \sim \models$ itt is igaz lesz.

Többértékű levezetési szabályok

Két fontos új szabály

- $X \to Y \vdash X \twoheadrightarrow Y$, mert $t_3 = t_2$ és $t_4 = t_1$ jó lesz.
- $X \twoheadrightarrow Y \vdash X \twoheadrightarrow R \setminus XY$, mert $t'_3 = t_4$ és $t'_4 = t_3$ jó lesz.
- De pl. $X \twoheadrightarrow AB \nvdash X \twoheadrightarrow A$, nem szétvágható. (Sok minden máshogy van a többértékű függéseknél.)

Tétel. *Legyen $\rho = (R_1, R_2)$ az (R, F) séma felbontása, ahol F most funkcionális és többértékű függéseket is tartalmaz. ρ akkor és csak akkor hűséges felbontás, ha $R_1 \cap R_2 \twoheadrightarrow R_2 \setminus R_1$.*

Megjegyzés: Nem kell a „vagy $R_1 \cap R_2 \twoheadrightarrow R_1 \setminus R_2$” a fenti 2. szabály miatt, mert ha $R_1 \cap R_2 \twoheadrightarrow R_1 \setminus R_2$ igaz, akkor $R_1 \cap R_2 \twoheadrightarrow R \setminus (R_1 \setminus R_2)$ is igaz, ebből meg már következik $R_1 \cap R_2 \twoheadrightarrow R_2 \setminus R_1$.

a tétel bizonyítása hasonló, mint a funkcionális függésnél, de nem bizonyítjuk.

4NF

Cél: olyan normálforma, amiben többértékű függés miatt sincs redundancia.

BCNF mintájára:
Definíció. *Az (R, F) séma **4NF (negyedik normálformájú)**, ha tetszőleges nemtriviális $X \twoheadrightarrow Y \in F^+$ esetén X szuperkulcs (a szuperkulcsot a régi értelemben, csak funkcionális függőségekkel definiálva).*

Következmény. *Ha egy séma 4NF, akkor BCNF is.*

Bizonyítás: Indirekt tegyük fel, hogy létezik olyan $X \to A \in F^+$ nemtriviális függés, ahol X nem szuperkulcs. $\implies$ Ekkor $\lightning$, amiatt, hogy $X \to A$-ból következik, hogy $X \twoheadrightarrow A$.

Megjegyzések:

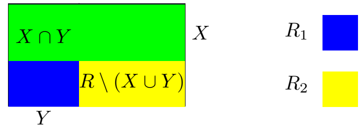
- Ha F-ben csak funkcionális függőségek vannak, akkor 4NF=BCNF
- 2 attribútumos reláció mindig 4NF, hiszen nincs nemtriviális többértékű függés, azt meg már láttuk, hogy ha csak funkcionális függések vannak, akkor a BCNF-ség rendben van kétattribútumos relációnál.
- Van olyan reláció, ami BCNF, de nem 4NF (a korábbi gyerekes példa, mert ott a Név nem szuperkulcs)

Tétel. *Legyen (R, F) egy séma, ahol F funkcionális és többértékű függések halmaza. Ekkor (R, F) felbontható hűségesen 4NF relációkra.*

Algoritmus: Hasonlóan BCNF-hez, mindig két valódi részre bontjuk hűségesen, addig, amíg mindegyik rész 4NF nem lesz.

Keresünk egy $X \twoheadrightarrow Y$ függést, ami megsérti a 4NF feltételt.
(Ha van $\to$, ami megsérti, akkor van $\twoheadrightarrow$ is.)
Nem tanuljuk, hogy ezt hogy kell általában, mert bonyolult, de ha nem kell keresni, mert ott van, akkor meg tudjuk csinálni (ezt tudni kell majd ZH-n, vizsgán)

$$R_1 = XY \qquad R_2 = R \setminus (Y \setminus X) \ (= X \cup (R \setminus Y))$$



Ez valódi felbontás:
Ha $R_1 = R \implies X \twoheadrightarrow Y$ triviális függés lenne, $\lightning$.
Ha $R_2 = R \implies Y \subseteq X \implies X \twoheadrightarrow Y$ triviális függés lenne, $\lightning$.

Ez hűséges felbontás:
$R_1 \cap R_2 = X$; $R_2 \setminus R_1 = R \setminus XY$ és $X \twoheadrightarrow R \setminus XY$ fennáll $X \twoheadrightarrow Y$ miatt.

Példa

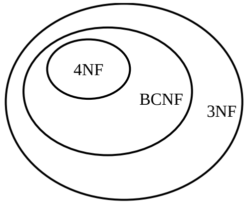
R(Színész, Város, Utca, Filmcím, Filmév)

F = {Színész $\twoheadrightarrow$ Város, Utca}

Ez megsérti a 4NF tulajdonságot, ha Színész nem szuperkulcs.

4NF felbontás: R_1 =(Színész, Város, Utca) R_2 =(Színész, Filmcím, Filmév)

Normálformák összefoglalása



Jellemzők	3NF	BCNF	4NF
Megszünteti a funkcionális függőségekből eredő redundanciát	Gyakran	Igen	Igen
Megszünteti a többértékű függőségekből eredő redundanciát	Nem	Nem	Igen
Az ilyen felbontás megőrzi a funkcionális függőségeket	Igen	Lehet	Lehet
Az ilyen felbontás megőrzi a többértékű függőségeket	Lehet	Lehet	Lehet

Fontos elv: Ne bontsuk tovább, amit már nem muszáj.

A normalizálás azért fontos, mert . . .

Adatbázisrendszerek megvalósítása

Eddig az adatbáziskezelők működéséről tanultunk. Az év hátralevő részében az ilyen rendszerek belső működését tanulmányozzuk egy kicsit.

Három nagyobb témakör:

1. Lekérdezésfeldolgozás: hogyan értékelődnek ki a lekérdezések, milyen módszerek vannak a lekérdezések végrehajtására?
2. Fizikai szervezés, tárkezelés: hogyan tároljuk a relációkat oly módon, hogy gyorsan lehessen keresni, illetve módosítani?
3. Tranzakciókezelés: többfelhasználós működés biztosítása, illetve rendszerhibák elleni védelem.

Lekérdezések végrehajtása, „optimalizálása”

Elemzés (parsing):
- szintaktikai ellenőrzés $\Longrightarrow$ *megfelelő parancsok, megfelelő sorrendben*
- átírás elemzőfa alakra

Előfeldolgozó:
- Relációk használatának ellenőrzése $\Longrightarrow$ *van-e ilyen*
- Attribútumnevek használatának ellenőrzése $\Longrightarrow$ *pl. egyértelmű-e, melyik attribútum melyik relációban van, benne van-e egyáltalán*
- típusellenőrzések $\Longrightarrow$ *pl. LIKE használatakor csak karakterlánc lehet*

Logikai lekérdezési terv:
- Átírás (kibővített) relációs algebrai alakra
- Transzformációk $\Longrightarrow$ *több terv, gyorsítás*
- Legjobb terv kiválasztása költségbecsléssel

Fizikai terv kiválasztása:
- Algoritmusok a műveletekhez
- Pufferkezelés
- Közbülső relációk eltárolása

A relációs algebra kibővítése

Az SQL többet tud, mint a relációs algebra, de az extra dolgokat is át akarjuk írni relációs formába. Néhány különbség:

- Multihalmazok $\Longrightarrow \cap_H, \cap_M$
- Kiválasztásnál, $\underset{\theta}{\bowtie}$-nál a feltételben használhatunk aritmetikai műveleteket $\Longrightarrow \sigma_{A+B<5}(R),\ R \underset{A+R.B<C+S.B}{\bowtie} S$
- Vetítés aritmetikai műveletekkel és átnevezéssel $\Longrightarrow \pi_{A,\, B+C \to X}(R)$
- Ismétlődések kiszűrése $\Longrightarrow \delta(R)$
- Csoportosítások, aggregátumok $\Longrightarrow$ SELECT $A, MIN(B)$ AS $minB$ FROM R GROUP BY A $\Longrightarrow \gamma_{A,MIN(B)\to minB}(R)$

Fizikai végrehajtás

Leginkább az I/O műveletigény érdekes. Ha „túl nagy” a számítási igény az is baj lehet.

Soronkénti, unáris műveletek: Kiválasztás és vetítés. Egyszerre csak egy sort kell vizsgálni, az algoritmus nem függ a memória nagyságától.

Unáris, teljes relációs műveletek: Pl. $\delta(R), \gamma(R)$. Ha nem fér el a reláció a memóriában, akkor mást kell csinálni.

Bináris, teljes relációs műveletek: $\cup, \cap, \setminus, \times, \bowtie$. Sok minden függ a méretektől.

$\sigma_C(R)$ végrehajtása: Blokkonként beolvassuk R-et. Soronként megnézzük teljesül-e C. Ha igen, kiírjuk.
I/O műveletigény: $B(R)$, blokkszám
Ha $\sigma_{A='c'}(R)$-t akarjuk, és van index A-ra: sokkal gyorsabb lehet.