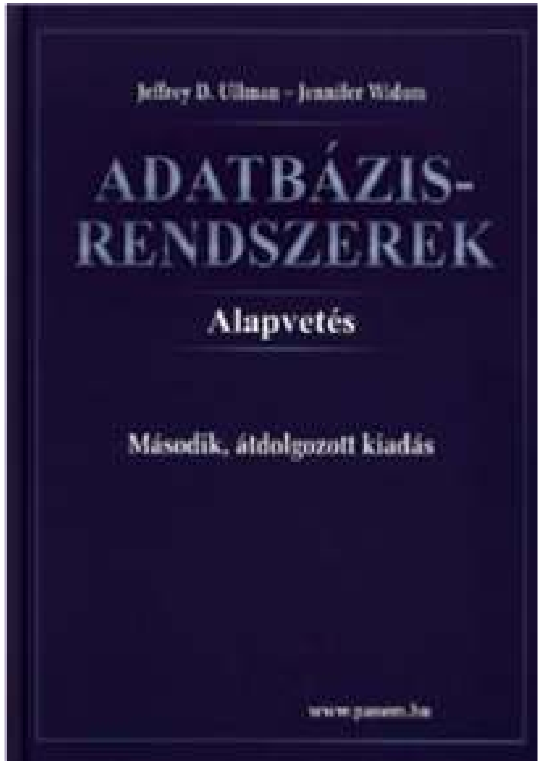
Relációs adatbázisok tervezése ---2

Tankönyv: Ullman-Widom:
Adatbázisrendszerek Alapvetés
Második, átdolgozott kiadás,
Panem, 2009



3.2.8. Funkcionális függ-ek vetítése
3.3.3. Boyce-Codd normálforma
3.3.4. BCNF-ra való felbontás
3.4.1-3.4.3. Információ visszanyerése
a komponensekből. Chase-teszt
a veszteségmentesség ellenőrzésére

Relációs sématervezés (vázlat)

- I. Függőségek: a sématervezésnél használjuk
  - Funkcionális függőség (10ea_RelTerv1FF)
  - Többértékű függőség (12ea_RelTerv3NF)
- II. Normalizálás: „jó” sémákra való felbontás
  - Funkcionális függ. -> BCNF (11ea_RelTerv2VM)
  - Funkcionális függ. -> 3NF (12ea_RelTerv3NF)
  - Többértékű függ. -> 4NF (12ea_RelTerv3NF)
- III. Felbontás tulajdonságai: „jó” tulajdonságok
  - Veszteségmentesség (11ea_RelTerv2VM)
  - Függőségőrző felbontás (12ea_RelTerv3NF)

FF-i halmaz vetülete (definíció)

- Tegyük fel, hogy adott az R reláció egy F funkcionális függőségi halmazzal.
- Vegyük R egy vetítését L-re: $R_1 = \Pi_L (R)$, ahol L az R reláció sémájának néhány attribútuma.
- Mely függőségek állnak fenn a vetületben?
- Erre a választ az F funkcionális függőségek L-re való vetülete adja, azok a függőségek, amelyek
  - (1) az F-ből levezethetők és
  - (2) csak az L attribútumait tartalmazzák.

FF-ek vetítése

- Motiváció: „normalizálás”, melynek során egy reláció sémát több sémára bonthatunk szét.
- Példa: R=*ABCD* F={*AB* ->*C*, *C* ->*D*, *D* ->*A* }
  - Bontsuk fel R-et R_1=*ABC* és R_2=*AD*-re.
  - Milyen FF-k teljesülnek R_1=*ABC*-n?
  - *ABC* -n nemcsak az *AB* ->*C*, de a *C* ->*A* is teljesül!

Miért igaz az előző példa?

Példa: $R=ABCD$ F={$AB \to C$, $C \to D$, $D \to A$ }

d={ABC, AD}. Milyen FF-k teljesülnek ABC -n?

ABCD $a_1b_1cd_1$ $a_2b_2cd_2$

ebből

ABC a_1b_1c a_2b_2c

$d_1=d_2$ mert $C \to D$

$a_1=a_2$ mert $D \to A$

Emiatt, ha két vetített sor C-n megegyezik A-n is, azaz: $C \to A$.

Ezért ABC -n az $AB \to C$ és a $C \to A$ is teljesül!

FF-i halmaz vetületének kiszámítása

- Induljunk ki a megadott FF-ekből és keressük meg az összes *nem triviális* FF-t, ami a megadott FF-ekből következik.
  - Nem triviális = vagyis a jobboldalt nem tartalmazza a baloldal, lásd A1 axióma.
- Csak azokkal az FF-kel foglalkozzunk, amelyekben a projektált séma attribútumai szerepelnek.

Algoritmus (FF-i halmaz vetülete)

- Legyen T az előálló FF-ek halmaza. Kezdetben T üres
- Minden X attribútum halmazra számítsük ki X^+-t.
- Adjuk hozzá a függőségeinkhez $X \to A$-t minden A-ra $X^+ - X$-ből.
- Dobjuk ki $XY \to A$ -t, ha $X \to A$ is teljesül

  mert $XY \to A$ $X \to A$ -ból minden esetben következik
- Végül csak azokat az FF-ket használjuk, amelyekben csak a projektált attribútumok szerepelnek.

Példa: FF-k projekciója

- ABC, $A \rightarrow B$ és $B \rightarrow C$ FF-kel. Projektáljunk AC-re.
  - $A^+ = ABC$; ebből $A \rightarrow B$, $A \rightarrow C$.

    Nem kell kiszámítani AB^+ és AC^+ lezárásokat.
  - $B^+ = BC$; ebből $B \rightarrow C$.
  - $C^+ = C$; semmit nem ad.
  - $BC^+ = BC$; semmit nem ad.
- A kapott FF-ek: $A \rightarrow B$, $A \rightarrow C$ és $B \rightarrow C$.
- AC -re projekció: $A \rightarrow C$.

Boyce-Codd normálforma

- Definíció: *R* reláció Boyce-Codd normálformában, BCNF-ban (BCNF) van, ha
  - minden *X* -> *Y* nemtriviális FF-re *R*-ben (nemtriviális, vagyis *Y* nem része *X*-nek)
  - az *X* szuperkulcs

  (szuperkulcs, vagyis tartalmaz kulcsot).

Példa BCNF-ra

Sörivók(<u>név</u>, cím, <u>kedveltSörök</u>, gyártó, kedvencSör)

FF-ek: név->cím kedvencSör, kedveltSörök->gyártó

- Itt egy kulcs van: {név, kedveltSörök}.
- A baloldalak egyik FF esetén sem szuperkulcsok.
- Emiatt az *Sörivók* reláció nincs BCNF normálformában.

egy másik példa BCNF-ra

Sörök(<u>név</u>, gyártó, gyártóCím)

FF-ek: név->gyártó, gyártó->gyártóCím

- Az egyetlen kulcs {név} .
- név->gyártó nem sérti a BCNF feltételét, de a gyártó->gyártóCím függőség igen.

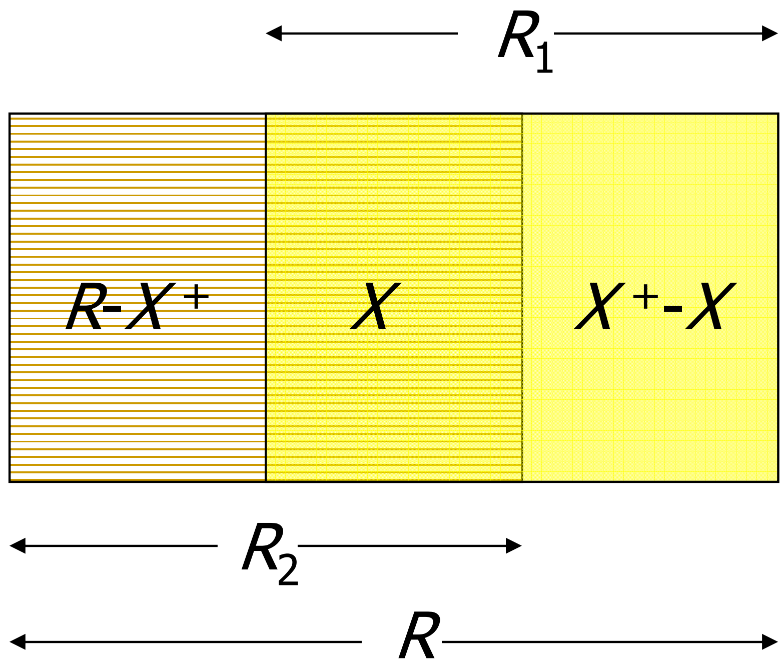
BCNF-re való felbontás ---1

- Adott *R* reláció és *F* funkcionális függőségek.
- Van-e olyan $X \to Y$ FF, ami sérti a BCNF-t?
  - Ha van olyan következmény FF *F*-ben, ami sérti a BCNF-t, akkor egy *F*-beli FF is sérti.
- Kiszámítjuk X^+-t:
  - Ha itt nem szerepel az összes attribútum, *X* nem szuperkulcs.

BCNF-re való felbontás ---2

- R-t helyettesítsük az alábbiakkal:
  1. $R_1 = X^+$.
  2. $R_2 = R - (X^+ - X)$.
- *Projektáljuk* a meglévő F -beli FF-eket a két új relációsémára.

Dekomponálási kép



Példa: BCNF dekompozíció ---1

Sörivók(<u>név</u>, cím, <u>kedveltSörök</u>, gyártó, kedvencSör)

F = név->cím, név->kedvenSör,
kedveltSörök->gyártó

- Vegyük név->cím FF-t:
- {név}⁺ = {név, cím, kedvencSör}.
- A dekomponált relációsémák:
  1. Sörivók1(<u>név</u>, cím, kedvencSör)
  2. Sörivók2(<u>név</u>, <u>kedveltSörök</u>, gyártó)

Példa: BCNF dekompozíció ---2

- Meg kell néznünk, hogy az Sörivók1 és Sörivók2 táblák BCNF-ben vannak-e.
- Az FF-ek projektálása könnyű.
- A Sörivók1(<u>név</u>, cím, kedvencSör), az FF-ek név->cím és név->kedvencSör.
  - Tehát az egyetlen kulcs: {név}, azaz Sörivók1 relációséma BCNF-ben van.

Példa: BCNF dekompozíció ---3

- Az Sörivók2(<u>név</u>, <u>kedveltSörök</u>, gyártó) esetén
  az egyetlen FF: kedveltSörök->gyártó,
  az egyetlen kulcs: {név, kedveltSörök}.
  - Sérül a BCNF.
- kedveltSörök^{+} = {kedveltSörök, gyártó}, a *Sörivók2* felbontása:
  1. Sörivók3(<u>kedveltSörök</u>, gyártó)
  2. Sörivók4(<u>név</u>, <u>kedveltSörök</u>)

Példa: BCNF dekompozíció ---4

- Az *Sörivók* dekompozíciója tehát:
  1. Sörivók1(<u>név</u>, cím, kedvencSör)
  2. Sörivók 3(<u>kedveltSörök</u>, gyártó)
  3. Sörivók 4(<u>név</u>, <u>kedveltSörök</u>)
- A *Sörivók1* a sörivókról, a *Sörivók3* a sörökről, az *Sörivók4* a sörivók és kedvelt söreikről tartalmaz információt.

Miért működik a BCNF?

- Feladat-1: Az algoritmus befejeződik, mert legrosszabb esetben két attribútumból álló sémáig jutunk. Bebizonyítandó, hogy minden két attribútumú séma BCNF-ban van! (mert nincs olyan FF, ami sértené a BCNF definíciót)
- Feladat-2: A felbontás jó tulajdonágú, vagyis veszteségmentes felbontást ad, visszatérünk erre: Bizonyítsuk be, hogy ha R(A, B, C) reláció esetén $B \rightarrow C$ teljesül, akkor $R_1(A, B)$, $R_2(B, C)$ felbontás mindig veszteségmentes

Példa: BCNF-ra való felbontás

$R=ABCD$, $F=\{AB \to C,\ C \to A\}$

Például R: lakcím
A: város, kerület
B: utca, házszám
D: emelet, ajtó
C: irányítószám

$(ABCD, \{AB \to C,\ C \to A\})$
Kulcsok: ABD, BCD

$(ABC, \{AB \to C,\ C \to A\})$
Kulcsok: AB, BC

$(ABD, \emptyset)$

$(AC, \{C \to A\})$

$(BC, \emptyset)$

Tehát d=(AC,BC,ABD) veszteségmentes BCNF dekompozíció.
($\emptyset$ azt jelenti, hogy csak a triviális függőségek teljesülnek a sémában.)

Tankönyv 3.5.2. feladata (111.o.)

- Órarend adatbázis: Kurzus(K), Oktató(O), Időpont(I), Terem(T), Diák(D), Jegy(J)
- Feltételek:

  Egy kurzust csak egy oktató tarthat: $K \to O$.

  Egy helyen egy időben egy kurzus lehet: $IT \to K$.

  Egy időben egy tanár csak egy helyen lehet: $IO \to T$.

  Egy időben egy diák csak egy helyen lehet: $ID \to T$.

  Egy diák egy kurzust egy végső jeggyel zár: $KD \to J$.
- R=KOITDJ F= $\{K \to O, IT \to K, IO \to T, ID \to T, KD \to J\}$
- Feladat: Adjuk meg az algoritmussal egy BCNF dekompozícióját!

Felbontásra vonatkozó elvárások

- Elvárások

(1) A vetületek legyenek jó tulajdonságúak, és a vetületi függőségi rendszere egyszerű legyen (normálformák: BCNF, 3NF, 4NF)

(2) Veszteségmentes legyen a felbontás, vagyis ne legyen információvesztés

(3) Függőségek megőrzése a vetületekben (FŐ)

- BCNF-ra való felbontás algoritmusa
  - mindig veszteségmentes felbontást ad
  - De nem feltétlen függőségőrző a felbontás

Veszteségmentes szétvágás ---1

- A fenti jelölésekkel: ha $r = \Pi_{R_1}(r) \bowtie \ldots \bowtie \Pi_{R_k}(r)$ teljesül, akkor az előbbi összekapcsolásra azt mondjuk, hogy veszteségmentes. Itt r egy R sémájú reláció-előfordulást jelöl.

R

A	B	C
a	b	c
d	e	f
c	b	c

R_1

A	B
a	b
d	e
c	b

R_2

B	C
b	c
e	f

Veszteségmentes szétvágás ---2

- Megjegyzés: Könnyen belátható, hogy $r \subseteq \Pi_{R_1}(r) \bowtie \ldots \bowtie \Pi_{R_k}(r)$ mindig teljesül.
- Példa: itt a szétvágás után keletkező relációk összekapcsolása nem veszteségmentes:

R

A	B	C
a	b	c
c	b	e

R_1

A	B
a	b
c	b

R_2

B	C
b	c
b	e

Chase-teszt VM ellenőrzése ---1

- Példa: adott R(A, B, C, D), F = { A → B, B → C, CD → A } és az R_1(A, D), R_2(A, C), R_3(B, C, D) felbontás. Kérdés veszteségmentes-e a felbontás?
- Vegyük $R_1 \bowtie R_2 \bowtie R_3$ egy t = (a, b, c, d) sorát. Bizonyítani kell, hogy t R egy sora. A következő tablót készítjük:

A	B	C	D
a	b_1	c_1	d
a	b_2	c	d_2
a_3	b	c	d

Itt pl. az (a, b_1, c_1, d) sor azt jelzi, hogy R-nek van olyan sora, aminek R_1-re való levetítése (a, d), ám ennek a B és C attribútumokhoz tartozó értéke ismeretlen, így egyáltalán nem biztos, hogy a t sorról van szó.

Chase-teszt VM ellenőrzése ---2

- Az F-beli függőségeket használva egyenlővé tesszük azokat a szimbólumokat, amelyeknek ugyanazoknak kell lennie, hogy valamelyik függőség ne sérüljön.
  - Ha a két egyenlővé teendő szimbólum közül az egyik index nélküli, akkor a másik is ezt az értéket kapja.
  - Két indexes szimbólum esetén a kisebbik indexű értéket kapja meg a másik.
  - A szimbólumok minden előfordulását helyettesíteni kell az új értékkel.
- Az algoritmus véget ér, ha valamelyik sor t-vel lesz egyenlő, vagy több szimbólumot már nem tudunk egyenlővé tenni.

Chase-teszt VM ellenőrzése ---3

A	B	C	D
a	b_1	c_1	d
a	b_2	c	d_2
a_3	b	c	d

$A \to B$

A	B	C	D
a	b_1	c_1	d
a	b_1	c	d_2
a_3	b	c	d

$B \to C$

A	B	C	D
a	b_1	c	d
a	b_1	c	d_2
a_3	b	c	d

$CD \to A$

A	B	C	D
a	b_1	c	d
a	b_1	c	d_2
a	b	c	d

Chase-teszt VM ellenőrzése ---5

- Ha t szerepel a tablóban, akkor valóban R-nek egy sora, s mivel t-t tetszőlegesen választottuk, ezért a felbontás veszteségmentes.
- Ha nem kapjuk meg t-t, akkor viszont a felbontás nem veszteségmentes.
- Példa: R(A, B, C, D), F = { B → AD }, a felbontás: R_1(A, B), R_2(B, C), R_3(C, D).

A	B	C	D
a	b	c_1	d_1
a_2	b	c	d_2
a_3	b_3	c	d

B → AD

A	B	C	D
a	b	c_1	d_1
a	b	c	d_1
a_3	b_3	c	d

Itt az eredmény jó ellenpélda, hiszen az összekapcsolásban szerepel t = (a, b, c, d), míg az eredeti relációban nem.

Kérdés/Válasz

- Köszönöm a figyelmet! Kérdés/Válasz?

- A gyakorlaton lesz illetve HF: Oracle PL/SQL
- HF: Oracle Példatár 9-10.fejezetének feladatai lekérdezések használata programból, kurzorok, tárolt eljárások és függvények.