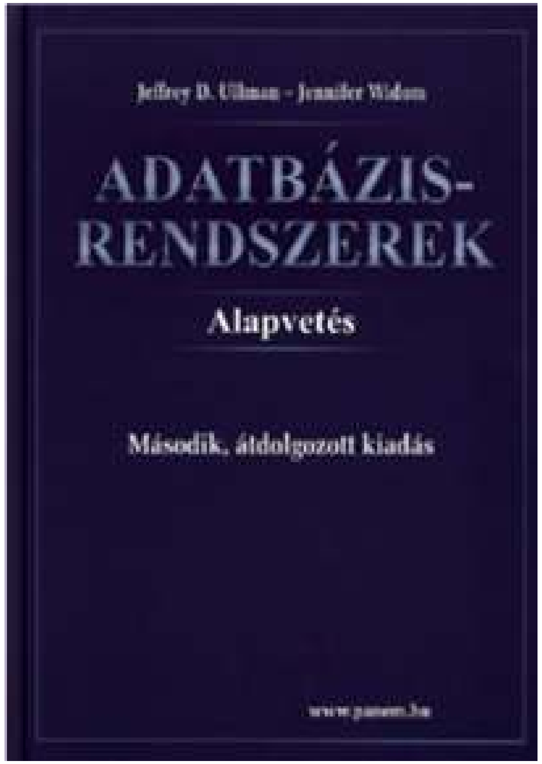
Relációs adatbázisok tervezése ---1

Tankönyv: Ullman-Widom:

Adatbázisrendszerek Alapvetés

Második, átdolgozott kiadás, 2009



3.3.1. Bevezetés: anomáliák

3.3.2. Relációk felbontása

3.1. Funkcionális függőségek

3.2. Funk.függőségek szabályai,
X⁺ attribútumhalmaz lezártja,
és X⁺ kiszámítására algoritmus,
Funkc.függőségi halmazok lezárása,
Funkcionális függőségek vetítése

Folyt. 2.részben: 3.3.3-3.3.4. Boyce-Codd normálforma és
3.4. A felbontások tulajdonságai, 3.részben 3NF, stb.

Relációs adatmodell története

- E.F. Codd 1970-ban publikált egy cikket
  A Relational Model of Data for Large Shared Data Banks
  Link: http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf

  amelyben azt javasolta, hogy az adatokat táblázatokban, **relációkban** tárolják. Az elméletére alapozva jött létre a relációs adatmodell, és erre épülve jöttek létre a relációs adatmodellen alapuló (piaci) relációs adatbázis-kezelők.
- Relációs sématervezés: függőségeken alapuló felbontás, normalizálás: Ezen a kurzuson a funkcionális függőségen alapuló Boyce_Codd normálformát (BCNF) és a 3NF-t, illetve a többértékű függőségen alapuló 4normálformát tanuljuk, és megvizsgáljuk a felbontások tulajdonságait (veszteségmentesség, függőségőrzés).

Tankönyv 3.fejezet: Bevezető példa

- Több tábla helyett -> vegyük egy táblában lenne Melyik séma jobb?

Sörivó(<u>név</u>, cím, <u>kedveltSörök</u>, gyártó, aKedvencSör)

név	cím	kedveltSörök	gyártó	kedvencS
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	???	WickedAle	Pete's	???
Spock	Enterprise	Bud	???	Bud

- Redundáns adat, a ??? helyén mi szerepel, ha mindenkinek csak egy lakcíme és aKedvencSöre lehet, vagyis a név meghatározza a címet és a aKedvencSör-t és a Söröknek is egy gyártója lehet

Hibás tervezés problémái

A rosszul tervezettség anomáliákat is eredményez

Sörivó(<u>név</u>, cím, <u>kedveltSörök</u>, gyártó, aKedvencSör)

név	cím	kedveltSörök	gyártó	kedvencS
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	???	WickedAle	Pete's	???
Spock	Enterprise	Bud	???	Bud

- Módosítási anomália: ha Janeway-t Jane-re módosítjuk, megtesszük-e ezt minden sornál?
- Törlési anomália: Ha senki sem szereti a Bud sört, azzal töröljük azt az infót is, hogy ki gyártotta.
- Beillesztési anomália: és felvinni ilyen gyártót?

Relációs sémák tervezése

- Cél: az anomáliák és a redundancia megszüntetése.
  - Módosítási anomália : egy adat egy előfordulását megváltoztatjuk, más előfordulásait azonban nem.
  - Törlési anomália : törléskor olyan adatot is elveszítünk, amit nem szeretnénk.
  - Beillesztési anomália : megszorítás, trigger kell, hogy ellenőrizni tudjuk (pl. a kulcsfüggőséget)
  - Redundancia (többszörös tárolás feleslegesen)

Dekomponálás (felbontás)

A fenti problémáktól dekomponálással (felbontással) tudunk megszabadulni:

- Definíció:
  $d=\{R_1,...,R_k\}$ az (R,F) dekompozíciója, ha nem marad ki attribútum, azaz $R_1 \cup ... \cup R_k = R$.
  Az adattábla felbontását projekcióval végezzük.
- Példa:
  R=ABCDE, d={AD,BCE,ABE}
  3 tagú dekompozíció, ahol
  R_1=AD, R_2=BCE, R_3=ABE,

Felbontásra vonatkozó elvárások

- Elvárások

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

(2) A felbontás is jó tulajdonságú legyen, vagyis ne legyen információvesztés: Veszteségmentes legyen a felbontás (VM)

(3) Függőségek megőrzése a vetületekben (FŐ)

- Tételek (ezekre nézünk majd algoritmusokat)
  - Mindig van VM BCNF-ra való felbontás
  - Mindig van VM FŐ 3NF-ra való felbontás

Relációs sématervezés (vázlat)

- I. Függőségek: a sématervezésnél használjuk
  - Funkcionális függőség
  - Többértékű függőség
- II. Normalizálás: „jó” sémákra való felbontás
  - Funkcionális függőség -> BCNF
  - Funkcionális függőség -> 3NF
  - Többértékű függőség -> 4NF
- III. Felbontás tulajdonságai: „jó” tulajdonságok
  - Veszteségmentesség
  - Függőségőrző felbontás

Funkcionális függőségek

- X ->Y *az R* relációra vonatkozó megszorítás, miszerint ha két sor megegyezik *X* összes attribútumán, *Y* attribútumain is meg kell, hogy egyezzenek.
  - Jelölés: *X*, *Y*, *Z* ,… attribútum halmazokat; *A*, *B*, *C*,… attribútumokat jelöl.
  - Jelölés: {*A*,*B*,*C* } attribútum halmaz helyett *ABC*-t írunk.

Funkcionális függőségek (definíció)

Definíció. Legyen R(U) egy relációséma, továbbá X és Y az U attribútum-halmaz részhalmazai. X-től *funkcionálisan függ* Y (jelölésben $X \rightarrow Y$), ha bármely R feletti T tábla esetén valahányszor két sor megegyezik X-en, akkor megegyezik Y-on is $\forall$ t1,t2$\in$T esetén (t1[X]=t2[X]$\Rightarrow$ t1[Y]=t2[Y]).

Ez lényegében azt jelenti, hogy az X-beli attribútumok értéke egyértelműen meghatározza az Y-beli attribútumok értékét.

- Jelölés: R |= $X \rightarrow Y$ vagyis
  R kielégíti $X \rightarrow Y$ függőséget

Példa: Funkcionális függőség

Sörivók(név, cím, kedveltSörök, gyártó, aKedvencSör)

Feltehetjük például, hogy az alábbi FF-ek teljesülnek:

név	cím	kedveltSörök	gyártó	aKedvencSö
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	Voyager	WickedAle	Pete's	WickedAle
Spock	Enterprise	Bud	A.B.	Bud

Mert név -> cím

Mert név -> aKedvencSör

Mert kedveltSörök -> gyártó

Jobboldalak szétvágása (FF)

- $X\text{->}A_1A_2\ldots A_n$ akkor és csak akkor teljesül R relációra, ha $X\text{->}A_1$, $X\text{->}A_2,\ldots,$ $X\text{->}A_n$ is teljesül R-en.
- Példa: $A\text{->}BC$ ekvivalens $A\text{->}B$ és $A\text{->}C$ függőségek kettősével.
- Baloldalak szétvágására nincs szabály!!!
- Ezért elég nézni, ha a FF-k jobboldalán egyetlen attribútum szerepel

Kulcs, szuperkulcs

- Funkcionális függőség $X \rightarrow Y$ speciális esetben, ha $Y = U$, ez a kulcsfüggőség.
- R(U) relációséma esetén az U attribútum-halmaz egy K részhalmaza akkor és csak akkor szuperkulcs, ha a $K \rightarrow U$ FF teljesül.
- A kulcsot tehát a függőség fogalma alapján is lehet definiálni: olyan K attribútum-halmazt nevezünk kulcsnak, amelytől az összes többi attribútum függ (vagyis szuperkulcs), de K-ból bármely attribútumot elhagyva ez már nem teljesül (vagyis minimális szuperkulcs)

Példa: szuperkulcs, kulcs

Sörivók(név, cím, kedveltSörök, gyártó, aKedvencSör)

- {név, kedveltSörök} szuperkulcs, ez a két attr. meghatározza funkcionálisan a maradék attr-kat.

  név -> cím aKedvencSör

  kedveltSörök -> gyártó

- {név, kedveltSörök} kulcs, hiszen sem {név}, sem {kedveltSörök} nem szuperkulcs.

  név -> gyártól; kedveltSörök -> cím nem telj.

- Az előbbi kívül nincs több kulcs, de számos szuperkulcs megadható (ami ezt tartalmazza)

Másik példa (több kulcs is lehet)

- Legyen *ABC* sémán def.FF-ek: *AB* ->*C* és *C* ->*B*.
  - Példa: *A* = utca, *B* = város, *C* = irányítószám.
- Itt két kulcs is van: {*A*,*B* } és {*A*,*C* }.

Az implikációs probléma

- Legyenek X_1 -> A_1, X_2 -> A_2,…, X_n -> A_n adott FF-k, szeretnénk tudni, hogy Y -> B teljesül-e olyan relációkra, amire az előbbi FF-k teljesülnek.
  - Példa: A -> B és B -> C teljesülése esetén A -> C biztosan teljesül.
- Implikációs probléma eldöntése definíció alapján (minden előfordulásra ellenőrizni) túl nehéz, de van egyszerűbb lehetőség: levezetési szabályok segítségével, lásd Armstrong-axiómák.

Armstrong-axiómák

Legyen R(U) relációséma és X,Y $\subseteq$ U, és jelölje XY az X és Y attribútum-halmazok egyesítését.

F legyen funkcionális függőségek tetsz. halmaza.

Armstrong axiómák:

- A1 (reflexivitás): $Y \subseteq X$ esetén $X \rightarrow Y$.
- A2 (bővíthetőség): $X \rightarrow Y$ és tetszőleges Z esetén $XZ \rightarrow YZ$.
- A3 (tranzitivitás): $X \rightarrow Y$ és $Y \rightarrow Z$ esetén $X \rightarrow Z$.

Levezetés fogalma

- F implikálja $X \to Y$-t (F-nek következménye $X \to Y$), ha minden olyan táblában, amelyben F összes függősége teljesül, azokra $X \to Y$ is teljesül.

  Jelölés: $F \models X \to Y$, ha F implikálja $X \to Y$ –et.

- $X \to Y$ levezethető F-ből, ha van olyan $X_1 \to Y_1$, ..., $X_k \to Y_k$,..., $X \to Y$ véges levezetés, hogy $\forall k$-ra $X_k \to Y_k \in F$ vagy $X_k \to Y_k$ az FD1,FD2,FD3 axiómák alapján kapható a levezetésben előtte szereplő függőségekből.

  Jelölés: $F \vdash X \to Y$, ha $X \to Y$ levezethető F-ből

További levezethető szabályok:

4. Szétvághatósági (vagy felbontási) szabály
   $F \vdash X \rightarrow Y$ és $Z \subseteq Y$ esetén $F \vdash X \rightarrow Z$.
5. Összevonhatósági (vagy unió) szabály
   $F \vdash X \rightarrow Y$ és $F \vdash X \rightarrow Z$ esetén $F \vdash X \rightarrow YZ$.
6. Pszeudotranzitivitás
   $F \vdash X \rightarrow Y$ és $F \vdash WY \rightarrow Z$ esetén $F \vdash XW \rightarrow Z$.

Bizonyítás (4): Reflexivitási axióma miatt $F \vdash Y \rightarrow Z$, és tranzitivitási axióma miatt $F \vdash X \rightarrow Z$.

Bizonyítás (5): Bővítési axióma miatt $F \vdash XX \rightarrow YX$ és $F \vdash YX \rightarrow YZ$, és $XX=X$, valamint a tranzitivitási axióma miatt $F \vdash X \rightarrow YZ$.

Bizonyítás (6): Bővítési axióma miatt $F \vdash XW \rightarrow YW$, és $YW=WY$, és a tranzitivitási axióma miatt $F \vdash XW \rightarrow Z$.

Szétvághatóság/összevonhatóság

- A szétvághatósági és összevonhatósági szabályok következménye:

  $F \vdash X \rightarrow Y \Leftrightarrow \forall A_i \in Y$ esetén $F \vdash X \rightarrow A_i$

- A következmény miatt feltehető, hogy a függőségek jobb oldalai 1 attribútumból állnak.

- Fontos! A függőségeknek csak a jobboldalát lehet szétbontani, a baloldalra ez természetesen nem igaz (például {filmcím, év} $\rightarrow$ stúdió)

Armstrong-axiómák (tétel)

- TÉTEL: Az Armstrong-axiómarendszer helyes és teljes, azaz minden levezethető függőség implikálódik is, illetve azok a függőségek, amelyeket F implikál azok levezethetők F-ből.

$$F \vdash X \to Y \Leftrightarrow F \models X \to Y$$

Implikáció eldöntése --- Lezárással

- Implikációs probléma:
  Legyenek X_1 -> A_1, X_2 -> A_2,…, X_n -> A_n adott FF-k, szeretnénk tudni, hogy Y -> B teljesül-e minden olyan relációkra, amire az előbbi FF-k teljesülnek. Hogyan tudjuk ellenőrizni, hogy egy előfordulás nem teljesíti Y -> B ?
- Mivel az Armstrong axiómarendszer helyes és teljes, elegendő a levezetési szabályokkal levezetni. Még a levezetési szabályoknál is van egyszerűbb út: kiszámítjuk *Y lezártját*: Y^+-t
- Attribútum-halmaz lezárására teszt:

Attribútumhalmaz lezártja (definíció)

- Adott *R* séma és *F* funkcionális függőségek halmaza mellett, Y^+ az összes olyan *A* attribútum halmaza, amire *Y->A* következik *F*-ből.
- (R,F) séma esetén legyen $Y \subseteq R$.
- Definíció: $Y^{+(F)} := \{A \mid F \vdash Y \rightarrow A\}$
az Y attribútum-halmaz lezárása F-re nézve.

Attribútumhalmaz lezártja (lemma)

- LEMMA: $F \vdash Y \rightarrow Z \Leftrightarrow Z \subseteq Y^+$.

  Bizonyítás:

  ($\Rightarrow$) $\forall A \in Z$ esetén a reflexivitás és tranzitivitás miatt $F \vdash Y \rightarrow A$, azaz $Z \subseteq Y^+$.

  ($\Leftarrow$) $\forall A \in Z \subseteq Y^+$ esetén $F \vdash Y \rightarrow A$, és az egyesítési szabály miatt $F \vdash Y \rightarrow Z$.

- Lemma következménye: az implikációs probléma megoldásához elég az Y^+-t hatékonyan kiszámolni.

Algoritmus X^+attr.halmaz lezártja

- Input: Y attribútumhz., F funk.függőségek hz.
- Output: Y^+ (zárás, típusa: attribútumhalmaz)
- Algoritmus Y^+ kiszámítására:

  /* Iteráció, amíg Y(n) változik */

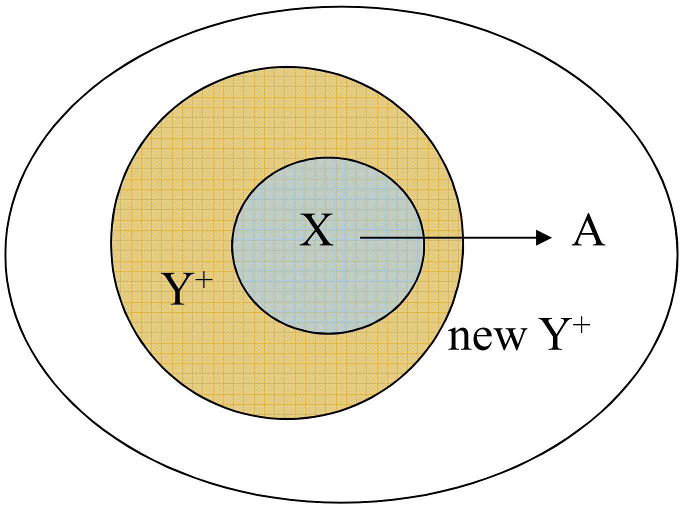
  $Y(0):=Y$

  $Y(n+1):= Y(n) \cup \{A \mid \underline{X \to Z \in F}, A \in Z, X \subseteq Y(n)\}$

  Ha $Y(v+1)=Y(v)$, akkor Output: $Y(v)=Y^+$.
- Miért működik az Y^+ lezárási algoritmus? (Tankönyv 3.2.5. szakasz, 81-83.oldal)

Lezárás (teszt)

- Kiindulás: $Y^+ = Y$
- Indukció: Olyan FF-ket keresünk, melyeknek a baloldala már benne van Y^+-ban. Ha $X \to A$ ilyen, A-t hozzáadjuk Y^+-hoz.



A lezárást kiszámító algoritmus „helyes”

- Az algoritmus „tényleg” Y^+-t számítja ki. Vagyis:
  1. Ha az A attribútum valamely *i*-re belekerül Y(i)-be, akkor A valóban eleme Y^+-nak.
  2. Másfelől, ha A $\in Y^+$, akkor létezik olyan *i*, amire A belekerül Y(i)-be.
- Az első állítás: Miért csak az igaz funkcionális függőségeket fogadja el a lezárási algoritmus? Könnyen bizonyítható indukcióval [Tk.3.2.5.]

A lezárást kiszámító algoritmus „teljes”

- A második állítás: Miért talál meg minden igaz függőséget a lezárási algoritmus? [Tk.3.2.5.]
- Konstrukciós bizonyítás: Tegyük fel, hogy $A \in Y^+$, és nem olyan *i*, amire A belekerül Y(i)-be

	X^+ elemei	más attribútumok
t	111 … 111	000 … 000
s	111 … 111	111 … 111

- Ekkor I-re minden F^+-beli függőség teljesül
- I-re viszont nem teljesül $X \rightarrow A$

Példa: Attribútumhalmaz lezárása

$R = ABCDEFG$, $\{AB \to C,\ B \to G,\ CD \to EG,\ BG \to E\}$

$X = ABF$, $X^+ = ?$

$X(0) := ABF$

$X(1) := ABF \cup \{C, G\} = ABCFG$

$X(2) := ABCFG \cup \{C, G, E\} = ABCEFG$

$X(3) := ABCEFG$

$X^+ = ABCEFG$

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: $IT \to K$.

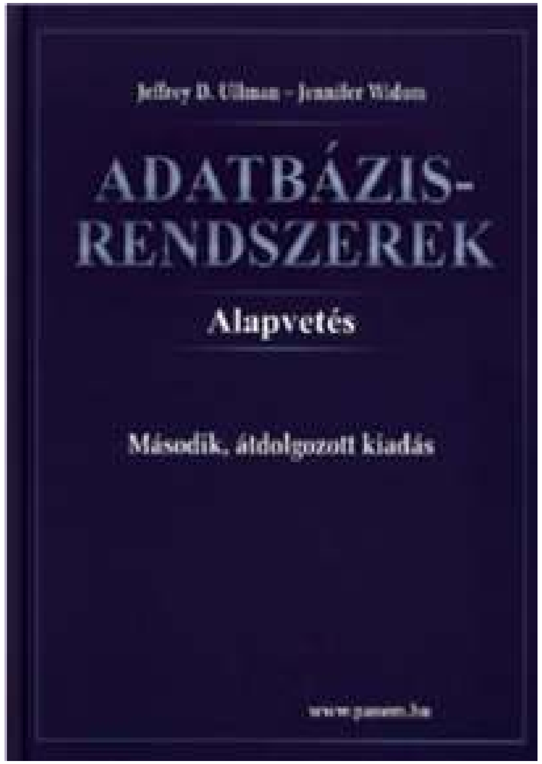
  Egy időben egy tanár csak egy helyen lehet: $IO \to T$.

  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: Határozzuk meg a (R, F) kulcsait az X^+ kiszámítási algoritmusa segítségével.

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
Folyt. 3.4. Információ visszanyerése
a komponensekből. Chase-teszt
a veszteségmentesség ellenőrzésére

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.
- Definíció: Függőségek vetülete

  Adott (R,F), és $R_i \subseteq R$ esetén:

  $\Pi_{R_i}(F) := \{ X \to Y \mid F \vdash X \to Y,\ XY \subseteq R_i \}$
- 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* ->*C*, *C* ->*D*, *D* ->*A* }
d={*ABC, AD*}. Milyen FF-k teljesülnek *ABC* -n?

ABCD $a_1 b_1 c d_1$ $a_2 b_2 c d_2$

ebből ↓ ↓

ABC $a_1 b_1 c$ $a_2 b_2 c$

$d_1=d_2$ mert
C -> D

$a_1=a_2$ mert
D -> A

Emiatt, ha két vetített sor C-n
megegyezik A-n is, azaz: *C -> A.*

Ezért *ABC* -n az *AB* ->*C* és a *C* ->*A* is teljesül!

Boyce-Codd normálforma

- Definíció: *R* reláció Boyce-Codd normálformában, BCNF-ban (BCNF) van, ha
  - minden $X \text{->} Y$ 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, aKedvencSör)

FF-ek: név->cím aKedvencSör, 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.

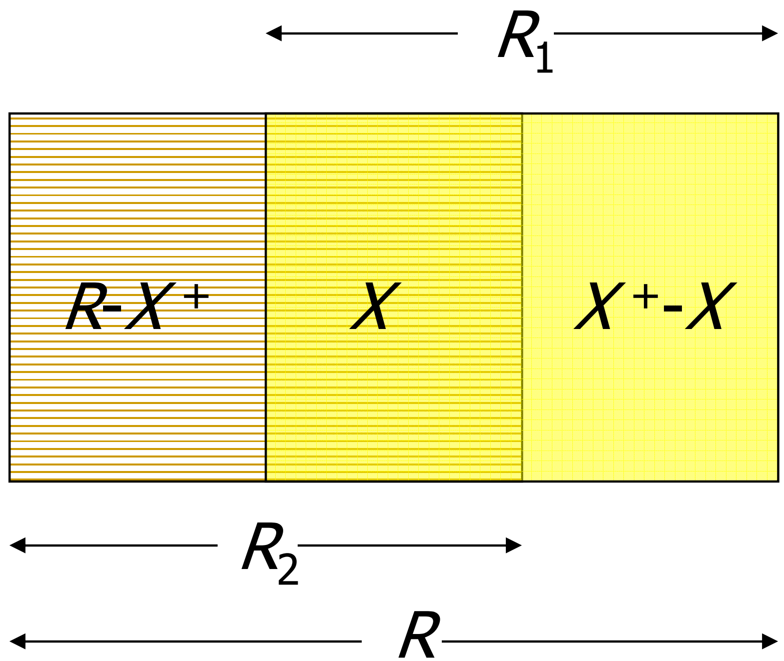
BCNF-re való felbontás ---1

- Adott *R* reláció és *F* funkcionális függőségek.
- Van-e olyan *X* -> *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(név, cím, kedveltSörök, gyártó, aKedvencSör)

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

- Vegyük név->cím FF-t:
- {név}$^+$ = {név, cím, aKedvencSör}.
- A dekomponált relációsémák:
  1. Sörivók1(név, cím, aKedvencSör)
  2. Sörivók2(név, kedveltSörök, 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, aKedvencSör), az FF-ek név->cím és név->aKedvencSör.
  - 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, aKedvencSör)
  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

$(ABD, \varnothing)$

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

$(BC, \varnothing)$

Tehát $d=(AC,BC,ABD)$ veszteségmentes BCNF dekompozíció.
($\varnothing$ 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: $K \rightarrow O$.

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

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

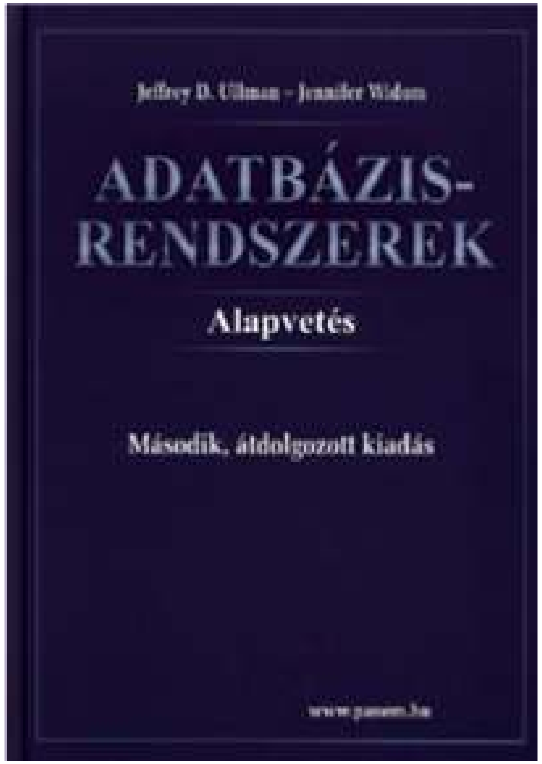
  Egy időben egy diák csak egy helyen lehet: $ID \rightarrow T$.

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

Relációs adatbázisok tervezése ---2b

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

3.4. Információ visszanyerése
a komponensekből. Chase-teszt
a veszteségmentesség ellenőrzésére



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 = { $A \rightarrow B$, $B \rightarrow C$, $CD \rightarrow A$ } 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

$A \rightarrow B$ ⇒

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

$B \rightarrow C$ ⇒

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

$CD \rightarrow A$ ⇒

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

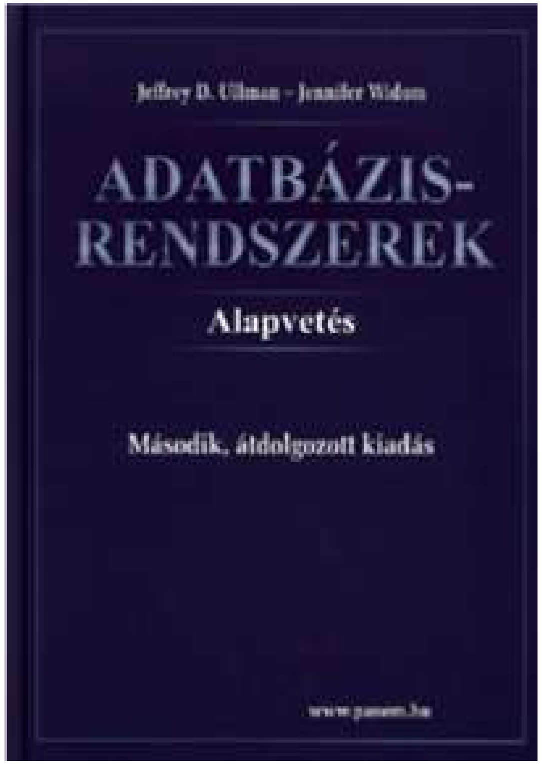
B → AD

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.

Relációs adatbázisok tervezése ---3

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



3.2.7. Funkcionális függőségi halmazok lezárása (min.bázis)

3.4.4. Függőségek megőrzése

3.5. Harmadik normálforma és 3NF-szintetizáló algoritmus

Relációs sématervezés (vázlat)

- I. Függőségek: a sématervezésnél használjuk
  - Funkcionális függőség
  - Többértékű függőség
- II. Normalizálás: „jó” sémákra való felbontás
  - Funkcionális függőség -> BCNF
  - Funkcionális függőség -> 3NF
  - Többértékű függőség -> 4NF
- III. Felbontás tulajdonságai: „jó” tulajdonságok
  - Veszteségmentesség
  - Függőségőrző felbontás

Függőségek megőrzése

- Függőségőrző felbontás: a dekompozíciókban érvényes függőségekből következzen az eredeti sémára kirótt összes függőség.
- Milyen függőségek lesznek érvényesek a dekompozíció sémáiban?
- Példa: R=ABC, F= {A→B, B→C , C→A} vajon a d= (AB, BC) felbontás megőrzi-e a C→A függőséget?

Függőségek megőrzése (definíció)

- Definíció: Függőségek vetülete

  Adott (R,F), és $R_i \subseteq R$ esetén:

  $\Pi_{R_i}(F) := \{ X \to Y \mid F \vdash X \to Y ,\ XY \subseteq R_i \}$

- Definíció: Adott (R,F) esetén $d=(R_1,\ldots,R_k)$ függőségőrző dekompozíció akkor és csak akkor, ha minden F-beli függőség levezethető a vetületi függőségekből:

  minden $X \to Y \in F$ esetén

  $\Pi_{R_1}(F) \cup \ldots \cup \Pi_{R_k}(F) \vdash X \to Y$

Példa: függőségek vetülete

- ABC, $A \to B$ és $B \to C$ FF-kel.
  Nézzük meg az AC-re való vetületet:
  - $A^+ = ABC$; ebből $A \to B$, $A \to C$.
    - Nem kell kiszámítani AB^+ és AC^+ lezárásokat.
  - $B^+ = BC$; ebből $B \to C$.
  - $C^+ = C$; semmit nem ad.
  - $BC^+ = BC$; semmit nem ad.
- A kapott FF-ek: $A \to B$, $A \to C$ és $B \to C$.
- AC -re projekció: $A \to C$.

Függőségek megőrzése (tételek)

- A függőségőrzésből nem következik a veszteségmentesség:

  R=ABCD, F= {A→B,C→D}, d={AB,CD} függőségőrző, de nem veszteségmentes.

- A veszteségmentességből nem következik a függőségőrzés

  R=ABC, F= {AB→C,C→A}, d={AC,BC} veszteségmentes, de nem függőségőrző.

A 3normálforma -- motiváció

- Bizonyos FF halmazok esetén a felbontáskor elveszíthetünk függőségeket.
- $AB \to C$ és $C \to B$.
  - Példa1: A = utca, B = város, C = irányítószám.
  - Példa2: A = oktató, B = időpont, C = kurzus.
- Két kulcs van: $\{A,B\}$ és $\{A,C\}$.
- $C \to B$ megsérti a BCNF-t, tehát AC, BC-re dekomponálunk.
- A probléma az, hogyAC és BC sémákkal nem tudjuk kikényszeríteni $AB \to C$ függőséget.

A probléma megoldása: 3NF

- 3. normálformában (3NF) úgy módosul a BCNF feltétel, hogy az előbbi esetben nem kell dekomponálnunk.
- Egy attribútum elsődleges attribútum (prím), ha legalább egy kulcsnak eleme.
- *X* ->*A* megsérti 3NF-t akkor és csak akkor, ha *X* nem szuperkulcs és *A* nem prím.

Példa: 3NF

- Az előző példában *AB ->C* és *C ->B* FF-ek esetén a kulcsok *AB* és *AC*.
- Ezért *A*, *B* és *C* mindegyike prím.
- Habár *C ->B* megsérti a BCNF feltételét, de a 3NF feltételét már nem sérti meg.

Miért hasznos 3NF és BCNF?

- A dekompozícióknak két fontos tulajdonsága lehet:
  1. *Veszteségmentes összekapcsolás* : ha a projektált relációkat összekapcsoljuk az eredetit kapjuk vissza.
  2. *Függőségek megőrzése* : a projektált relációk segítségével is kikényszeríthetőek az előre megadott függőségek.
- Az (1) tulajdonság teljesül a BCNF esetében.
- A 3NF (1) és (2)-t is teljesíti.
- A BCNF esetén (2) sérülhet (*utca-város-irszám)*

Tk.3.2.7. Minimális bázis (definíció)

Egy relációhoz F minimális bázis, amikor az olyan függőségekből áll, amelyre három feltétel igaz:

1. F összes függőségének jobb oldalán egy attribútum van.
2. Ha bármelyik F-beli függőséget elhagyjuk, a fennmaradó halmaz már nem bázis.
3. Ha bármelyik F-beli funkcionális függőség bal oldaláról elhagyunk egy vagy több attribútumot, akkor az eredmény már nem marad bázis.

Minimális bázist kiszámító algoritmus

1. Kezdetben G az üreshalmaz.
2. Minden $X \to Y \in F$ helyett vegyük az $X \to A$ függőségeket, ahol $A \in Y - X$).
   Megj.: Ekkor minden G-beli függőség $X \to A$ alakú.
3. Minden $X \to A \in G$-re, amíg van olyan $B \in X$-re $A \in (X - B)^+$ a G-szerint, vagyis $(X - B) \to A$ teljesül, akkor $X := X - B$.
   Megjegyzés: E lépés után minden baloldal minimális lesz.
4. Minden $X \to A \in G$-re, ha $X \to A \in (G - \{ X \to A \})^+$, vagyis ha elhagyjuk az $X \to A$ függőséget G-ből, az még mindig következik a maradékból, akkor $G := G - \{X \to A\}$.
   Megjegyzés: Végül nem marad több elhagyható függőség

Mohó algoritmus minimális bázis előállítására

1. Jobb oldalak minimalizálása:

   $X \rightarrow A1, ..., Ak$ függőséget cseréljük le az

   $X \rightarrow A1, \ldots, X \rightarrow Ak$ k darab függőségre.

2. A halmaz minimalizálása:

   Hagyjuk el az olyan $X \rightarrow A$ függőségeket, amelyek a bázist nem befolyásolják, azaz

   while F változik

       if $(F-\{X \rightarrow A\})^* = F^*$ then $F := F-\{X \rightarrow A\}$;

3. Bal oldalak minimalizálása:

   Hagyjuk el a bal oldalakból azokat az attribútumokat, amelyek a bázist nem befolyásolják, azaz

   while F változik

     for all $X \rightarrow A \in F$

       for all $B \in X$

         if $((F-\{X \rightarrow A\}) \cup \{(X-\{B\}) \rightarrow A\})^* = F^*$ then $F := (F-\{X \rightarrow A\}) \cup \{X-\{B\}) \rightarrow A\}$

Normálformák (3NF)

- Az algoritmusban különböző sorrendben választva a függőségeket, illetve attribútumokat, különböző minimális bázist kaphatunk.
- $F=\{A\to B,\ B\to A,\ B\to C,\ A\to C,\ C\to A\}$

  $(F-\{B\to A\})^*=F^*$, mivel $F-\{B\to A\} \vdash B\to A$

  $F:=F-\{B\to A\}$

  $(F-\{A\to C\})^*=F^*$, mivel $F-\{A\to C\} \vdash A\to C$

  $F:=F-\{A\to C\}= \{A\to B, B\to C, C\to A\}$ minimális bázis, mert több függőség és attribútum már nem hagyható el.

- $F=\{A\to B,\ B\to A,\ B\to C,\ A\to C,\ C\to A\}$

  $(F-\{B\to C\})^*=F^*$, mivel $F-\{B\to C\} \vdash B\to C$

  $F:=F-\{B\to C\}=\{A\to B, B\to A, A\to C, C\to A\}$ is minimális bázis, mert több függőség és attribútum már nem hagyható el.

Normálformák (3NF)

- Az algoritmusban különböző sorrendben választva a függőségeket, illetve attribútumokat, különböző minimális bázist kaphatunk.
- $F=\{AB\to C,\ A\to B,\ B\to A\}$

  $(F-\{AB\to C\}\cup\{A\to C\})^*=F^*$, mivel

  $(F-\{AB\to C\})\cup\{A\to C\} \vdash AB\to C$ és $F \vdash A\to C$.

  $F:=(F-\{AB\to C\}\cup\{A\to C\})=\{A\to C, A\to B, B\to A\}$ minimális bázis, mert több függőség és attribútum már nem hagyható el.
- $F=\{AB\to C,\ A\to B,\ B\to A\}$

  $(F-\{AB\to C\}\cup\{B\to C\})^*=F^*$, mivel

  $(F-\{AB\to C\})\cup\{B\to C\} \vdash AB\to C$ és $F \vdash B\to C$.

  $F:=(F-\{AB\to C\}\cup\{B\to C\})=\{B\to C, A\to B, B\to A\}$ is minimális bázis, mert több függőség és attribútum már nem hagyható el.

Normálformák (3NF)

- Algoritmus függőségőrző 3NF dekompozíció előállítására:
- Input: (R,F)
  - Legyen $G:=\{X\rightarrow A, X\rightarrow B, ..., Y\rightarrow C, Y\rightarrow D, ...\}$ az F minimális bázisa.
  - Legyen S az R sémának G-ben nem szereplő attribútumai.
  - Ha van olyan függőség G-ben, amely R összes attribútumát tartalmazza, akkor legyen $d:=\{R\}$, különben legyen

    $d:=\{S, XA, XB, ..., YC, YD, ...\}$.

Normálformák (3NF)

- Algoritmus függőségőrző és veszteségmentes 3NF dekompozíció előállítására:
- Input: (R,F)
  - Legyen $G:=\{X\rightarrow A, X\rightarrow B, \ldots, Y\rightarrow C, Y\rightarrow D, \ldots\}$ az F minimális bázisa.
  - Legyen S az R sémának G-ben nem szereplő attribútumai.
  - Ha van olyan függőség G-ben, amely R összes attribútumát tartalmazza, akkor legyen d:={R}, különben legyen K az R egy kulcsa, és legyen

    d:={K,S,XA, XB,...,YC,YD,...}.

Normálformák (3NF)

- Algoritmus függőségőrző és veszteségmentes 3NF redukált (kevesebb tagból álló) dekompozíció előállítására:
- Input: (R,F)
  - Legyen $G:=\{X\rightarrow A, X\rightarrow B, \ldots, Y\rightarrow C, Y\rightarrow D, \ldots.\}$ az F minimális bázisa.
  - Legyen S az R sémának G-ben nem szereplő attribútumai.
  - Ha van olyan függőség G-ben, amely R összes attribútumát tartalmazza, akkor legyen d:={R}, különben legyen K az R egy kulcsa, és legyen

    d:={K,S,XAB...,...,YCD...,...}.

- Ha K része valamelyik sémának, akkor K-t elhagyhatjuk.

Miért működik?

3NF-szintetizáló algoritmus:

- Megőrzi a függőségeket: minden FF megmarad a minimális bázisból.
- Veszteségmentes összekapcsolás: a CHASE algoritmussal ellenőrizhető (a kulcsból létrehozott séma itt lesz fontos).
- 3NF: a minimális bázis tulajdonságaiból következik.

Ellenőrző kérdések

- Adott X-attr.hz, F-ff.hz. Attr.hz.lezártjának kiszámítása.
- Adott R-rel.séma, F-ff.hz. Kulcsok meghatározása.
- Adott R-rel.séma, F-ff.hz. BCNF-e? (def. alapján)
- Adott R-rel.séma, F-ff.hz. 3NF-e? (def. alapján)

típusú kérdésekhez lásd 1.) gyakorló feladatok :

Ellenőrző kérdések

1.) Adott R relációs séma és F funkcionális függőségek halmaza. Attribútum halmaz lezártjának kiszámolására tanult algoritmus felhasználásával határozza meg az adott séma kulcsait, és azt, hogy BCNF-ben vagy 3NF-ben van-e?

a.) Cím(Város, Utcahsz, Irányítószám)

röviden R(V, U, I), és

$F = \{I \rightarrow V, VU \rightarrow I\}$.

Ellenőrző kérdések

b.) Tankönyv 3.5.2. feladata: Órarend adatbázis

Jelölje röviden:

- K - Kurzus
- O - Oktató
- I - Időpont
- T - Terem
- D - Diák (hallgató)
- J - Jegy

Feltételek (funkcionális függőséggel megadva)

- $K \rightarrow O$ vagyis egy kurzust csak egy oktató tarthat
- $IT \rightarrow K$ nincs óraütközés, egy helyen egy időben egy kurzus lehet
- $IO \rightarrow T$ az oktatónak nincs óraütközése
- $ID \rightarrow T$ a diákoknak sincs óraütközése
- $KD \rightarrow J$ egy diák egy kurzust egy végső jeggyel zár

R=KOITDJ és F= $\{K \rightarrow O, IT \rightarrow K, IO \rightarrow T, ID \rightarrow T, KD \rightarrow J\}$

Ellenőrző kérdések

c.) Adott SzallításiInfo (SzallAzon, SzallNev, SzallCim, AruKod, TermekNev, MeEgys, Ar) reláció séma,

amit így is rövidithetünk R(S, N, C, K, T, M, A), és

a séma feletti funkcionális függőségek:
SzallAzon→{SzallNev, SzallCim},
AruKod→{TermekNev, MeEgys},
{SzallAzon, AruKod}→ Ar,

vagyis a röviden F = {S → NC, K → TM, SK → A}.

Ellenőrző kérdések

- Adott R, F. Bontsuk fel VM BCNF-ra
- Adott R, F. Bontsuk fel VM FŐ 3NF-ra
- Adott R, F és d dekompozíció. Chase algoritmussal döntsük el, hogy veszteségmentes-e a dekompozíció.

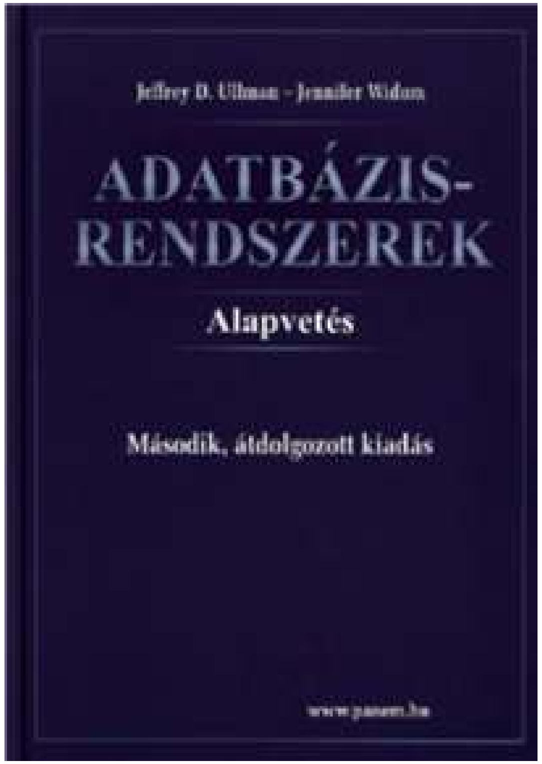
a.) Az 1a. feladat R sémáját szétvágjuk
IU, VU sémákra.

b.) Az 1b. feladat R sémáját szétvágjuk
KOIT, IDT, KDJ sémákra.

c.) Az 1c. feladat R sémáját szétvágjuk
SNC, KTMA sémákra.

Relációs adatbázisok tervezése
4.rész Többértékű függőségek

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



3.6. Többértékű függőségek,
- Negyedik normálforma
- Funkcionális és többértékű függőségek következtetése

Többértékű függőségek és 4NF

- Hasonló utat járunk be, mint a funkcionális függőségek esetén:
  - Definiáljuk a többértékű függőséget
  - implikációs probléma
  - axiomatizálás
  - levezethető függőségek hatékony meghatározása (lezárás helyett a séma particiója függőségi bázisa)
  - veszteségmentes dekompozíció
  - 4. normálforma
  - veszteségmentes 4NF dekompozíció előállítása

A TÉF definíciója

- A *többértékű függőség* (TÉF): az *R* reláció fölött *X* ->->*Y* teljesül: ha bármely két sorra, amelyek megegyeznek az *X* minden attribútumán, az *Y* attribútumaihoz tartozó értékek felcserélhetőek, azaz a keletkező két új sor R-beli lesz.
- Más szavakkal: *X* minden értéke esetén az *Y* -hoz tartozó értékek függetlenek az

  R-*X*-*Y* értékeitől.

Példa: TÉF

Sörivók(név, cím, tel, kedveltSörök)

- A sörivók telefonszámai függetlenek az általuk kedvelt söröktől.
  - név->->tel és név ->->kedveltSörök.
- Így egy-egy sörivó minden telefonszáma minden általa kedvelt sörrel kombinációban áll.
- Ez a jelenség független a funkcionális függőségektől.
  - itt a név->cím az egyetlen FF.

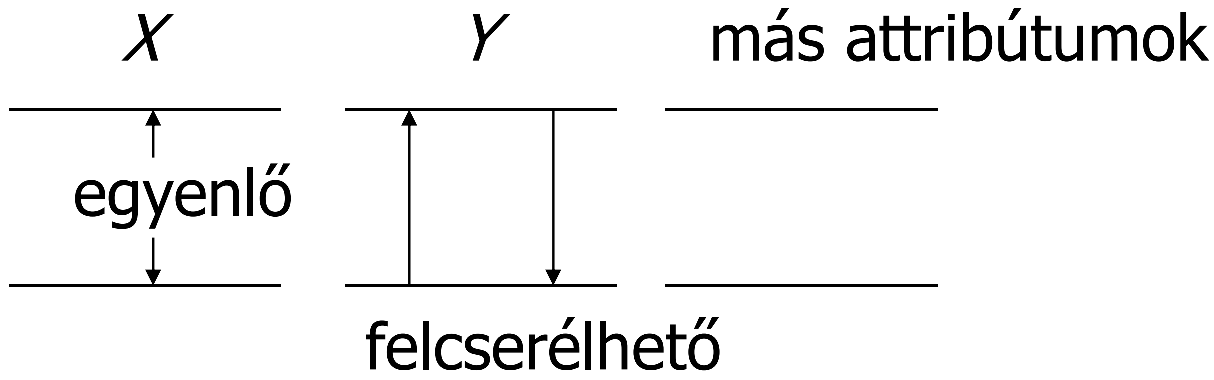
A név->->tel által implikált sorok

Ha ezek a soraink vannak:

név	cím	tel	kedveltSörök
sue	a	p1	b1
sue	a	p2	b2
sue	a	p2	b1
sue	a	p1	b2

Akkor ezeknek a soroknak is szerepelnie kell.

Az X ->-> Y TÉF képe



Többértékű függőségek

- **Definíció:** $X,Y \subseteq R$, $Z := R - XY$ esetén $X \rightarrow\rightarrow Y$ **többértékű függőség. (tf)**
- A függőség akkor teljesül egy táblában, ha bizonyos mintájú sorok létezése garantálja más sorok létezését.
- A formális definiciót az alábbi ábra szemlélteti.
- Ha létezik **t** és **s** sor, akkor **u** és **v** soroknak is létezniük kell, ahol az azonos szimbólumok azonos értékeket jelölnek.

	X	Y	Z
t	x	y1	z1
s	x	y2	z2
∃ u	x	y1	z2
∃ v	x	y2	z1

Többértékű függőségek

Definíció (Formálisan): Egy R sémájú r reláció kielégíti az $X \to \to Y$ függőséget, ha $t,s \in r$ és $t[X]=s[X]$ esetén létezik olyan $u,v \in r$, amelyre $u[X]=v[X]=t[X]=s[X]$, $u[Y]=t[Y]$, $u[Z]=s[Z]$, $v[Y]=s[Y]$, $v[Z]=t[Z]$.

Állítás: Elég az u,v közül csak az egyik létezését megkövetelni.

	X	Y	Z
t	x	y1	z1
s	x	y2	z2
∃ u	x	y1	z2

TÉF szabályok

- Minden FF TÉF.
  - Ha *X* -> *Y* és két sor megegyezik X-en, Y-on is megegyezik, emiatt ha ezeket felcseréljük, az eredeti sorokat kapjuk vissza, azaz: *X* ->-> *Y*.
- *Komplementálás* : Ha *X* ->-> *Y* és *Z* jelöli az összes többi attribútum halmazát, akkor

  X ->-> *Z*.

Nem tudunk darabolni

- Ugyanúgy, mint az FF-ek esetében, a baloldalakat nem „bánthatjuk” általában.
- Az FF-ek esetében a jobboldalakt felbonthattuk, míg ebben az esetben ez sem tehető meg.

Példa: többattribútumos jobboldal

Sörivók(név, tTársaság, tel, kedveltSörök, gyártó)

- Egy sörivónak több telefonja lehet, minden számot két részre otsztunk: tTársaság (pl. Vodafone) és a maradék hét számjegy.
- Egy sörivó több sört is kedvelhet, mindegyikhez egy-egy gyártó tartozik.

Példa folytatás

- Mivel a tTársaság-tel kombinációk függetlenek a kedveltSörök-gyártó kombinációtól, azt várjuk, hogy a következő FÉK-ek teljesülnek:

  név ->-> tTársaság tel

  név ->-> kedveltSörök gyártó

Példa adat

Egy lehetséges előfordulás, ami teljesíti az iménti FÉK-et:

név	tTásaság	tel	kedveltS	gyártó
Sue	30	555-1111	Bud	A.B.
Sue	20	555-1111	WickedAle	Pete's
Sue	70	555-9999	Bud	A.B.
Sue	70	555-9999	WickedAle	Pete's

Ugyanakkor sem a név->->tTársaság sem a név->->tel függőségek nem teljesülnek.

Többértékű függőségek

- Axiomatizálás

Funkcionális függőségek	Többértékű függőségek	Vegyes függőségek
A1 (reflexivitás): $Y \subseteq X$ esetén $X \rightarrow Y$.	A4 (komplementer): $X \rightarrow\rightarrow Y$ és $Z = R - XY$ esetén $X \rightarrow\rightarrow Z$.	A7 (funkcionálisból többértékű): $X \rightarrow Y$ esetén $X \rightarrow\rightarrow Y$.
A2 (tranzitivitás): $X \rightarrow Y$ és $Y \rightarrow Z$ esetén $X \rightarrow Z$.	A5 (tranzivitás): $X \rightarrow\rightarrow Y$ és $Y \rightarrow\rightarrow S$ esetén $X \rightarrow\rightarrow S - Y$.	A8 (többértékűből és funcionálisból funkcionális): $X \rightarrow\rightarrow Y$ és $W \rightarrow S$, ahol $S \subseteq Y$, $W \cap Y = \emptyset$ esetén $X \rightarrow S$.
A3 (bővíthetőség): $X \rightarrow Y$ és tetszőleges Z esetén $XZ \rightarrow YZ$.	A6 (bővíthetőség): $X \rightarrow\rightarrow Y$ és tetszőleges $V \subseteq W$ esetén $XW \rightarrow\rightarrow YV$.	

Többértékű függőségek

- Jelölés a továbbiakban:
  - F funkcionális függőségek halmaza
  - M többértékű függőségek halmaza
  - D vegyes függőségek (funkcionális és többértékű függőségek) halmaza
- Tétel (helyes és teljes axiómarendszerek):
  - A1,A2,A3 helyes és teljes a funkcionális függőségekre,
  - A4,A5,A6 helyes és teljes a többértékű függőségekre,
  - A1,A2,A3,A4,A5,A6,A7,A8 helyes és teljes a vegyes függőségekre.

Többértékű függőségek

- Állítás: X→→Y-ből nem következik, hogy X→→A, ha A∈Y. (A jobb oldalak nem szedhetők szét!)
- Bizonyítás: A következő r tábla kielégíti az X→→AB-t, de nem elégíti ki az X→→A-t. q.e.d.

X	A	B	C
x	a	b	c
x	e	f	g
x	a	b	g
x	e	f	c

X→→A esetén ennek a sornak is benne kellene lenni a táblában.

x	a	f	g

Többértékű függőségek

- Állítás: $X \to\to Y$ és $Y \to\to V$-ből nem következik, hogy $X \to\to V$. (A szokásos tranzitivitás nem igaz általában!)
- Bizonyítás: A következő r tábla kielégíti az $X \to\to AB$-t, $AB \to\to BC$-t, de nem elégíti ki az $X \to\to BC$-t. q.e.d.

X	A	B	C
x	a	b	c
x	e	f	g
x	a	b	g
x	e	f	c

$X \to\to BC$ esetén ennek a sornak is benne kellene lenni a táblában.

x	e	b	c

Többértékű függőségek

- A veszteségmentesség, függőségőrzés definíciójában most F funkcionális függőségi halmaz helyett D függőségi halmaz többértékű függőségeket is tartalmazhat.

- Így például d=(R1,...,Rk) veszteségmentes dekompozíciója R-nek D-re nézve, akkor és csak akkor, ha minden D-t kielégítő r tábla esetén $r=\Pi_{R1}(r)|\bowtie|\ldots|\bowtie|\,\Pi_{Rk}(r)$

- A következő tétel miatt a veszteségmentesség implikációs problémára vezethető vissza, így hatékonyan eldönthető.

- Tétel: A d=(R1,R2) akkor és csak akkor veszteségmentes dekompozíciója R-nek, ha
  $D \vdash R1 \cap R2 \rightarrow\rightarrow R1-R2$.

Többértékű függőségek

- A 4.normálforma definiálása előtt foglaljuk össze, hogy melyek a triviális többértékű függőségek, vagyis amelyek minden relációban teljesülnek.
- Mivel minden funkcionális függőség többértékű függőség is, így a triviális funkcionális egyben triviális többértékű függőség is.

1. $Y \subseteq X$ esetén $X \to\to Y$ triviális többértékű függőség.

- Speciálisan $Y=\emptyset$ választással $X \to\to \emptyset$ függőséget kapjuk, és alkalmazzuk a komplementer szabályt, azaz $Z=R-X\emptyset$, így az $X \to\to R-X$ függőség is mindig teljesül, azaz:

2. $XY=R$ esetén $X \to\to Y$ triviális többértékű függőség.

- A szuperkulcs, kulcs definíciója változatlan, azaz X szuperkulcsa R-nek D-re nézve, ha $D \vdash X \to R$.
- A minimális szuperkulcsot kulcsnak hívjuk.

Többértékű függőségek

- A 4.normálforma hasonlít a BCNF-re, azaz minden nem triviális többértékű függőség bal oldala szuperkulcs.
- Definíció: R 4NF-ben van D-re nézve, ha $XY \neq R$, $Y \not\subset X$, és

$$D \vdash X \to\to Y \text{ esetén } D \vdash X \to R.$$

- Definíció: d={R1,...,Rk} dekompozíció 4NF-ben van D-re nézve, ha minden Ri 4NF-ben van $\Pi_{Ri}(D)$-re nézve.
- Állítás: Ha R 4NF-ben van, akkor BCNF-ben is van.
- Bizonyítás. Vegyünk egy nem triviális $D \vdash X \to A$ funkcionális függőséget. Ha $XA=R$, akkor $D \vdash X \to R$, ha $XA \neq R$, akkor a $D \vdash X \to\to A$ nem triviális többértékű függőség és a 4NF miatt $D \vdash X \to R$. q.e.d.
- Következmény: Nincs mindig függőségőrző és veszteségmentes 4NF dekompozíció.

Többértékű függőségek

- Veszteségmentes 4NF dekompozíciót mindig tudunk készíteni a naiv BCNF dekomponáló algoritmushoz hasonlóan.
- Naiv algoritmus veszteségmentes 4NF dekompozíció előállítására:

  Ha R 4NF-ben van, akkor <u>megállunk</u>,

  egyébként

  van olyan nem triviális $X \rightarrow\rightarrow Y$, amely R-ben teljesül, de megsérti a 4NF-et, azaz X nem szuperkulcs.

  Ekkor R helyett vegyük az (XY,R-Y) dekompozíciót.

  A kettévágásokat addig hajtjuk végre, <u>amíg minden tag 4NF-ben nem lesz</u>.

  ALGORITMUS VÉGE.

Többértékű függőségek

- Az is feltehető, hogy X és Y diszjunkt, mert különben Y helyett az Y-X-et vehettük volna jobb oldalnak.
- $XY \neq R$, így *mindkét tagban csökken az attribútumok száma.*
- $XY \cap (R-Y) = X \rightarrow\rightarrow Y = XY-(R-Y)$, azaz a kéttagú dekompozícióknál bizonyított állítás miatt *veszteségmentes kettévágást kaptunk*.
- Legrosszabb esetben a 2 oszlopos sémákig kell szétbontani, amelyek mindig 4NF-ben vannak, mivel nem lehet bennük nem triviális többértékű függőség.

Negyedik normálforma

- A TÉF-ek okozta redundanciát a BCNF nem szünteti meg.
- A megoldás: a negyedik normálforma!
- A negyedik normálformában (4NF), amikor dekomponálunk, a TÉF-eket úgy kezeljük, mint az FF-eket, a kulcsok megtalálásánál azonban nem számítanak.

4NF definíció

- Egy *R* reláció *4NF* -ben van ha: minden *X* ->->*Y* nemtriviális FÉK esetén *X* szuperkulcs.
  - *Nemtriviális TÉF* :
    1. *Y* nem részhalmaza *X*-nek,
    2. *X* és *Y* együtt nem adják ki az összes attribútumot.
  - A szuperkulcs definíciója ugyanaz marad, azaz csak az FF-ektől függ.

BCNF versus 4NF

- Kiderült, hogy minden *X* -> *Y* FF
  X ->-> *Y* TÉF is.
- Így, ha *R* 4NF-ben van, akkor BCNF-ben is.
  - Mert minden olyan FF, ami megsérti a BCNF-t, a 4NF-t is megsérti.
- De *R* lehet úgy BCNF-ben, hogy közben nincs 4NF-ben.

Dekompozíció és 4NF

- H X ->->Y megsérti a 4NF-t, akkor R-t ugyanúgy dekomponáljuk, mint a BCNF esetén.
  1. XY az egyik dekomponált reláció.
  2. Az $Y – X$-be nem tartozó attribútumok a másik.

Példa: 4NF dekompozíció

Sörivók(<u>név</u>, cím, <u>tel</u>, <u>kedveltSörök</u>)

FF: név -> cím

FÉK-ek: név ->-> tel

név ->-> kedveltSörök

- Kulcs {név, tel, kedveltSörök}.
- Az összes függőség megsérti 4NF-et.

Példa folytatás

- Dekompozíció név -> cím szerint:

1. Sörivók1(<u>név</u>, cím)
   - Ez 4NF-beli; az egyetlen függőség név-> cím.
2. Sörivók2(<u>név</u>, <u>tel</u>, <u>kedveltSörök</u>)
   - Nincs 4NF-ben. A név ->-> tel és név ->-> kedveltSörök függőségek teljesülnek. A három attribútum együtt kulcs (mivel nincs nemtriviális FF).

Példa: Sörivók2 dekompozíciója

- Mind a név ->-> tel, mind a név ->-> kedveltSörök szerinti dekompozíció ugyanazt eredményezi:
  - Sörivók3(<u>név</u>, <u>tel</u>)
  - Sörivók4(<u>név</u>, <u>kedveltSörök</u>)