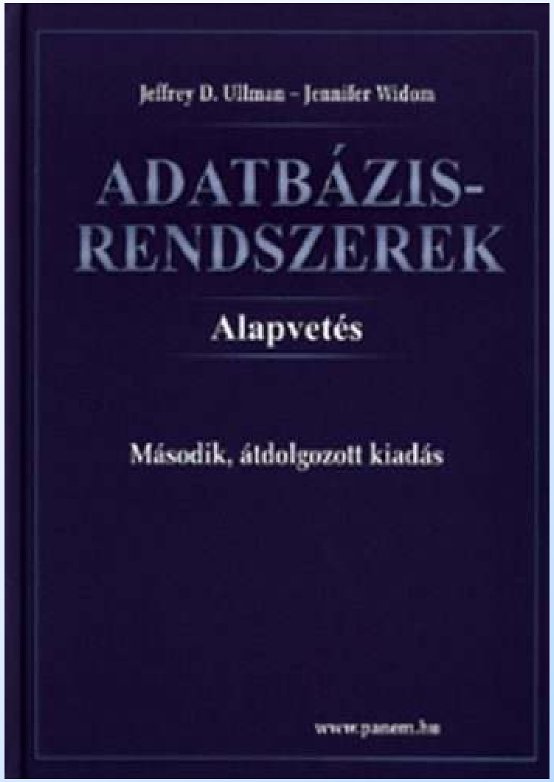
Relációs adatbázisok tervezése
5.rész (Többértékű függőségek)



Ullman-Widom: Adatbázisrendszerek

Alapvetés. Második, átdolgozott kiadás,

Panem, 2009

3.6. Többértékű függőségek

- Negyedik normálforma

(Jeffrey D. Ullman, 2007 slides alapján
Dr. Kiss Attila előadásainak felhasználásával)

Többértékű függőségek

- **Dolgozó adatbázis: Név(N), Diploma(D), Telefon(T)**
- **R=NDT**

N	D	T
Kovács	{programozó, közgazdász}	{1234567, 7654321, 1212123}
Szabó	{programozó, jogász}	{1234123, 1234512}

0. normálforma: az értékek halmazok is lehetnek.

Többértékű függőségek

- **Átírás 1. normálfomára (az értékek atomi értékek)**

N	D	T
Kovács	{programozó, közgazdász}	{1234567, 7654321, 1212123}
Szabó	{programozó, jogász}	{1234123, 1234512}

N	D	T
Kovács	programozó	1234567
Kovács	programozó	7654321
Kovács	programozó	1212123
Kovács	közgazdász	1234567
Kovács	közgazdász	7654321
Kovács	közgazdász	1212123
Szabó	programozó	1234123
Szabó	programozó	1234512
Szabó	jogász	1234123
Szabó	jogász	1234512

- Adott névhez diplomák halmaza és telefonszámok halmaza tartozik, egymástól függetlenül.
- Név $\rightarrow\rightarrow$ Diploma
- Név $\rightarrow\rightarrow$ Telefon

30 értéket tárolunk (redundancia)!

Többértékű függőségek

- **Dekomponáljuk 2 táblára veszteségmentesen:**

N	D
Kovács	programozó
Kovács	közgazdász
Szabó	programozó
Szabó	jogász

N	T
Kovács	1234567
Kovács	7654321
Kovács	1212123
Szabó	1234123
Szabó	1234512

18 értéket tárolunk (csökkent a redundancia)

- A 2 tábla összekapcsolása visszaadná az eredeti (redundáns) táblát, vagyis veszteségmentes lenne a dekompozíció.
- A funkcionális függőség speciális többértékű függőség, például Név $\rightarrow$ Telefon esetén 1 elemű halmaz (1 telefonszám) tartozik minden névhez, azaz Név $\rightarrow\rightarrow$ Telefon.

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 $X \rightarrow\rightarrow Y$ 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
$\exists$ u	x	y1	z2

Többértékű függőségek

- **Hasonló utat járunk be, mint a funkcionális függőségek esetén:**
  - **implikációs probléma**
  - **axiomatizálás**
  - **levezethető függőségek hatékony meghatározása (lezárás helyett a séma particiója (másképpen függőségi bázisa))**
  - **veszteségmentes dekompozíció**
  - **4. normálforma**
  - **veszteségmentes 4NF dekompozíció előállítása**
- **Mivel kijön majd, hogy minden 4NF egyben BCNF is, amire nincs egyszerre függőségőrző és veszteségmentes dekompozíció, így 4NF-re sincs mindig.**

Többértékű függőségek

- **Axiomatizálás**

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

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 (további levezetési szabályok):**
  1. $X \to\to Y$ és $X \to\to V$ esetén $X \to\to YV$.
  2. $X \to\to Y$ és $WX \to\to V$ esetén $WX \to\to V-WY$.
  3. $X \to\to Y$ és $XY \to V$ esetén $X \to V-Y$.
  4. $X \to\to Y$ és $X \to\to V$ esetén $X \to\to Y \cap V$

     és $X \to\to V-Y$

     és $X \to\to Y-V$.

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→→Y és Y→→V-ből nem következik, hogy X→→V. (A szokásos tranzitivitás nem igaz általában!)**
- **Bizonyítás: A következő r tábla kielégíti az X→→AB-t, AB→→BC-t, de nem elégíti ki az X→→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→→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 $D \vdash R1 \cap R2 \to\to R1 - R2$.

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 $Y=\varnothing$ választással $X \to\to \varnothing$ függőséget kapjuk, és alkalmazzuk a komplementer szabályt, azaz $Z=R-X\varnothing$, 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

  $D \vdash X \to\to Y$ 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 megállunk,

  egyébként

  van olyan nem triviális $X \to\to Y$, 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, amíg minden tag 4NF-ben nem lesz. **ALGORITMUS VÉGE.**

- 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.*

- $XY \cap (R-Y) = X \to\to Y = XY-(R-Y)$, 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.

Definition of MVD

- A *multivalued dependency* (MVD) on R, X ->-> Y, says that if two tuples of R agree on all the attributes of X, then their components in Y may be swapped, and the result will be two tuples that are also in the relation.
- i.e., for each value of X, the values of Y are independent of the values of R-X-Y.

Example: MVD

Drinkers(name, addr, phones, beersLiked)

- A drinker's phones are independent of the beers they like.
  - name->->phones and name ->->beersLiked.
- Thus, each of a drinker's phones appears with each of the beers they like in all combinations.
- This repetition is unlike FD redundancy.
  - name->addr is the only FD.

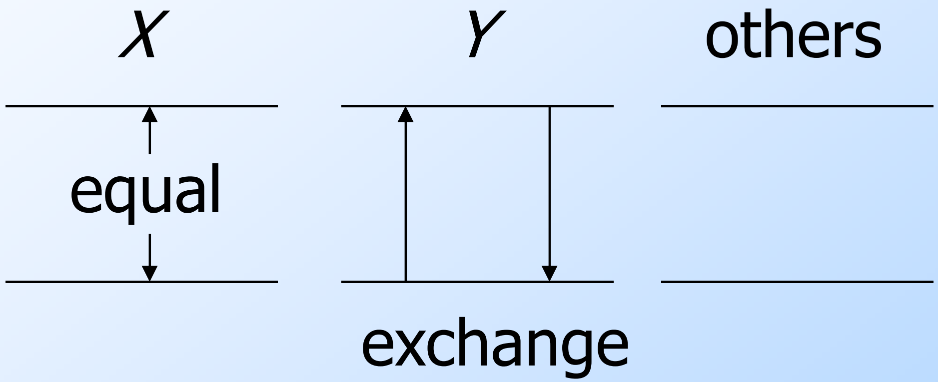
Tuples Implied by name->->phones

If we have tuples:

name	addr	phones	beersLiked
sue	a	p1	b1
sue	a	p2	b2
sue	a	p2	b1
sue	a	p1	b2

Then these tuples must also be in the relation.

Picture of MVD X ->-> Y



MVD Rules

- Every FD is an MVD (*promotion*).
  - If $X \rightarrow Y$, then swapping Y's between two tuples that agree on X doesn't change the tuples.
  - Therefore, the "new" tuples are surely in the relation, and we know $X \rightarrow\rightarrow Y$.
- *Complementation* : If $X \rightarrow\rightarrow Y$, and Z is all the other attributes, then $X \rightarrow\rightarrow Z$.

Splitting Doesn't Hold

- Like FD's, we cannot generally split the left side of an MVD.
- But unlike FD's, we cannot split the right side either --- sometimes you have to leave several attributes on the right side.

Example: Multiattribute Right Sides

Drinkers(name, areaCode, phone, beersLiked, manf)

- A drinker can have several phones, with the number divided between areaCode and phone (last 7 digits).
- A drinker can like several beers, each with its own manufacturer.

Example Continued

- Since the areaCode-phone combinations for a drinker are independent of the beersLiked-manf combinations, we expect that the following MVD's hold:

    name ->-> areaCode phone

    name ->-> beersLiked manf

Example Data

Here is possible data satisfying these MVD's:

name	areaCode	phone	beersLiked	manf
Sue	650	555-1111	Bud	A.B.
Sue	650	555-1111	WickedAle	Pete's
Sue	415	555-9999	Bud	A.B.
Sue	415	555-9999	WickedAle	Pete's

But we cannot swap area codes or phones by themselves. That is, neither name->->areaCode nor name->->phone holds for this relation.

Fourth Normal Form

- The redundancy that comes from MVD's is not removable by putting the database schema in BCNF.
- There is a stronger normal form, called 4NF, that (intuitively) treats MVD's as FD's when it comes to decomposition, but not when determining keys of the relation.

4NF Definition

- A relation *R* is in *4NF* if: whenever *X* ->-> *Y* is a nontrivial MVD, then *X* is a superkey.
  - *Nontrivial MVD* means that:
    1. *Y* is not a subset of *X*, and
    2. *X* and *Y* are not, together, all the attributes.
  - Note that the definition of "superkey" still depends on FD's only.

BCNF Versus 4NF

- Remember that every FD $X \text{->} Y$ is also an MVD, $X \text{->->} Y$.
- Thus, if R is in 4NF, it is certainly in BCNF.
  - Because any BCNF violation is a 4NF violation (after conversion to an MVD).
- But R could be in BCNF and not 4NF, because MVD's are "invisible" to BCNF.

Decomposition and 4NF

- If X ->-> Y is a 4NF violation for relation R, we can decompose R using the same technique as for BCNF.
  1. XY is one of the decomposed relations.
  2. All but $Y - X$ is the other.

Example: 4NF Decomposition

Drinkers(name, addr, phones, beersLiked)

FD: name -> addr

MVD's: name ->-> phones

name ->-> beersLiked

- Key is {name, phones, beersLiked}.
- All dependencies violate 4NF.

Example Continued

- Decompose using name -> addr:

1. Drinkers1(<u>name</u>, addr)
   - In 4NF; only dependency is name -> addr.
2. Drinkers2(<u>name</u>, <u>phones</u>, <u>beersLiked</u>)
   - Not in 4NF. MVD's name ->-> phones and name ->-> beersLiked apply. No FD's, so all three attributes form the key.

Example: Decompose Drinkers2

- Either MVD name ->-> phones or name ->-> beersLiked tells us to decompose to:
  - Drinkers3(name, phones)
  - Drinkers4(name, beersLiked)

Reasoning About MVD's + FD's

- Problem: given a set of MVD's and/or FD's that hold for a relation *R*, does a certain FD or MVD also hold in *R* ?
- Solution: Use a tableau to explore all inferences from the given set, to see if you can prove the target dependency.

Why Do We Care?

1. 4NF technically requires an MVD violation.
   - Need to infer MVD's from given FD's and MVD's that may not be violations themselves.
2. When we decompose, we need to project FD's + MVD's.

Example: Chasing a Tableau With MVD's and FD's

- To apply a FD, equate symbols, as before.
- To apply an MVD, generate one or both of the tuples we know must also be in the relation represented by the tableau.
- We'll prove: if A->->BC and D->C, then A->C.

The Tableau for A->C

Goal: prove that $c_1 = c_2$.

A	B	C	D
a	b_1	$\cancel{c_1}$ c_2	d_1
a	b_2	c_2	d_2
a	b_2	c_2	d_1

Use A->->BC (first row's D with second row's BC).

Use D->C (first and third row agree on D, therefore agree on C).

Example: Transitive Law for MVD's

- If A->->B and B->->C, then A->->C.
  - Obvious from the complementation rule if the Schema is *ABC*.
  - But it holds no matter what the schema; we'll assume *ABCD*.

The Tableau for A->->C

Goal: derive tuple (a, b_1, c_2, d_1).

A	B	C	D
a	b_1	c_1	d_1
a	b_2	c_2	d_2
a	b_1	c_2	d_2
a	b_1	c_2	

d_1

Use A->->B to swap *B* from the first row into the second.

Use B->->C to swap *C* from the third row into the first.

Rules for Inferring MVD's + FD's

- Start with a tableau of two rows.
  - These rows agree on the attributes of the left side of the dependency to be inferred.
  - And they disagree on all other attributes.
  - Use unsubscripted variables where they agree, subscripts where they disagree.

Inference: Applying a FD

- Apply a FD X->Y by finding rows that agree on all attributes of *X*. Force the rows to agree on all attributes of *Y*.
  - Replace one variable by the other.
  - If the replaced variable is part of the goal tuple, replace it there too.

Inference: Applying a MVD

- Apply a MVD X->->Y by finding two rows that agree in *X*.
  - Add to the tableau one or both rows that are formed by swapping the *Y*-components of these two rows.

Inference: Goals

- To test whether U->V holds, we succeed by inferring that the two variables in each column of V are actually the same.
- If we are testing U->->V, we succeed if we infer in the tableau a row that is the original two rows with the components of V swapped.

Inference: Endgame

- Apply all the given FD's and MVD's until we cannot change the tableau.
- If we meet the goal, then the dependency is inferred.
- If not, then the final tableau is a counterexample relation.
  - Satisfies all given dependencies.
  - Original two rows violate target dependency.