# 11.előadás: Adatbázisok
## dr. Hajas Csilla (ELTE IK)
## http://sila.hajas.elte.hu/

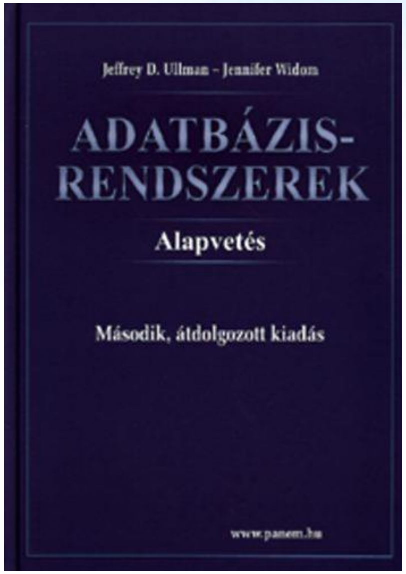
# Normalizálás (VM BCNF felbontás)

A mai témakörök a Tankönyvben:

3.2.8.Függőségek vetülete

3.3. Boyce-Codd normálforma (BCNF) és egy algoritmus BCNF-felbontásra

3.4. Veszteségmentes felbontás (VM), Chase-teszt a VM ellenőrzésére, függőségek megőrzése a vetületekben



# (Tk.3.3.1.) Motiváció: Bevezető példa

- Mi lenne, ha a Sörivók és a Sörök, meg a köztük lévő Szeret illetve aKedvencSör kapcsolatokat az eddigi több tábla helyett egy táblába egyesítve tárolnánk?

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

| név | cím | sör | gyártó | aKedvenc |
|---|---|---|---|---|
| Janeway | Voyager | Bud | A.B. | WickedAle |
| Janeway | Voyager | WickedAle | Pete's | WickedAle |
| Spock | Enterprise | Bud | A.B. | Bud |

- Redundancia, módosítási anomália, törlési anomália, beillesztési anomália
- Cél: redundancia csökkentése és az anomáliák megszüntetése.

# (Tk.3.3.2.) Relációk felbontása

Az anomáliák megszüntetésének útja a relációk felbontása (dekompozíciója). R felbontása azt jelenti, hogy R attribútumait szétosztjuk két vagy több reláció sémáját alakítjuk ki belőlük:

- Definíció: $d=\{R_1,\dots,R_k\}$ az (R,F) dekompozíciója, ha nem marad ki attribútum, azaz $R_1\cup\dots\cup R_k=R$.

  Az adattábla felbontását projekcióval végezzük.
- Megj.: Tk.3.3.2. Relációk felbontása c. részhez, lásd 91.o. kiegészítés: a felbontás nemcsak két relációra, hanem több új relációsémára történhet!
- Példa: R=ABCDE felbontása d={AD,BCE,ABE}

  ez pl. 3 tagú felbontás $R_1$=AD, $R_2$=BCE, $R_3$=ABE

# (Tk.3.2.8.) Funkc.függőségek 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* ->*C*, *C* ->*D*, *D* ->*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 -> 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!

# FF-i halmaz vetülete (definíció)

- Tegyük fel, hogy adott az R reláció (itt R jelöli az R relációsémát=attribútumok halmazát és az R séma feletti r előfordulást=táblát sorok halmazát) és adott F funkcionális függőségi halmaz.
- Vegyük R egy vetítését R1-re: $r_1 = \Pi_{R1}(r)$, ahol R1 az R reláció sémájának néhány attribútuma.
- Mely függőségek állnak fenn az R1 vetületben?
- Erre a választ az F funkcionális függőségek R1-re való vetülete adja, azok a függőségek, amelyek
  - (1) az F-ből levezethetők és
  - (2) csak az R1 attribútumait tartalmazzák.

# Függőségek vetülete

- Definíció: Függőségek vetülete

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

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

- Példa: *R*=*ABC*, *A* ->*B* és *B* ->*C* FF-kel.
  Nézzük meg az R1=*AC*-re való vetületet

# 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 $X \rightarrow A$-t minden $A$-ra $X^+ - X$ -ből.
- Dobjuk ki $XY \rightarrow A$ -t, ha $X \rightarrow A$ is teljesül

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

# 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: $A \to C$.

# 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 (10ea)
  - Többértékű függőség (12ea)
- II. Normalizálás: „jó” sémákra való felbontás
  - Funkcionális függ. -> **BCNF (11ea)**
  - Funkcionális függ. -> 3NF (12ea)
  - Többértékű függ. -> 4NF (12ea)
- III. Felbontás tulajdonságai: „jó” tulajdonságok
  - **Veszteségmentesség (11ea)**
  - **Függőségőrzés (11ea)**

# (Tk.3.3.3.) Boyce-Codd normálforma

- Definíció:

*R* reláció Boyce-Codd normálformában,

BCNF-ban van, ha

- minden *X* ->*Y* nem-triviális FF-re *R*-ben (nem-triviális, vagyis *Y* nem része *X*-nek)
- az *X* szuperkulcs (vagyis az *X*-ben szereplő attribútumok funkcionálisan meghatározzák a reláció összes többi attribútumát: *X* -> *R* )

# Példa: nincs BCNF-ban

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 Boyce-Codd normálformában.

# egy másik példa: nincs BCNF-ban

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.

Emiatt az *Sörök* reláció nincs BCNF-ban.

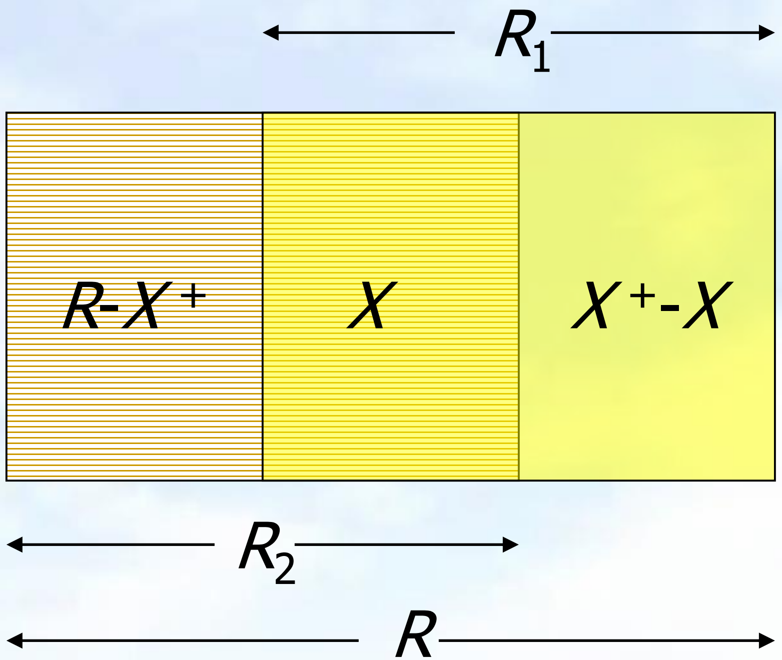
# Példa BCNF-ra

- Lemma: Azt állítjuk, hogy bármely két attribútumból álló reláció BCNF-ban van.
- Bizonyítás: Tegyük fel, hogy ezek az attribútumok A és B.
- Nincs nem-triviális függőség. Ekkor a BCNF feltétel mindenképpen érvényes, hiszen csak a nem triviális függőségek sérthetik meg a feltételt. (Egy kulcs: {A,B}.)
- A->B fennáll, de B->A nem áll fenn. Ekkor egy kulcs: A, és minden nem triviális függőség tartalmazza A-t a bal oldalon (valójában a bal oldal csak A lehet). Tehát nem sérül a BCNF feltétel. (Szimmetrikusan uígy A és B csere)
- Mindkét A->B és B->A fennáll. Ekkor mindkét attribútum, A is és B is kulcs. Bármelyik függőséget nézzük, a két attribútum közül az egyik a bal oldalon áll, emiatt nem sértheti meg a BCNF feltételt.

# (Tk.3.3.4.) BCNF-re való felbontás ---1

- Adott *R* reláció és *F* funkcionális függőségek.
- 1.) Ellenőrizzük, hogy R BCNF-ban van-e.
  - Ha igen, akkor készen vagyunk, és {R} a válasz.
  - Van-e olyan *X ->Y* FF, ami sérti a BCNF-t?
- 2.) Ha vannak BCNF-t megsértő függőségek, akkor X->Y legyen egy ilyen.
  - Kiszámítjuk $X^+$-t:
  - Legyen $R_1 = X^+$ ez valódi részhalmaza R-nek, nem szerepel benne az összes attribútum, mivel *X* nem szuperkulcs (az X->Y sértette a BCNF-t).

# BCNF-re való felbontás ---2




- Folyt. 2.) $R$-t helyettesítsük az alábbiakkal:
  - $R_1 = X^+$ az egyik relációséma és
  - $R_2 = X \cup (R - X^+)$ ebben legyenek benne X attribútumai és azok az R-beli attribútumok, amelyek nincsenek $X^+$-ban.

# BCNF-re való felbontás ---3

- 3.) Az F-beli funkcionális függőségeket vetítsük le az $R_1$ és $R_2$ reláció-sémákra, a vetületek legyenek rendre $F_1$ és $F_2$
- 4.) Rekurzívan folytassuk a felbontást $R_1$ -re illetve $R_2$ -re ennek az algoritmusnak a felhasználásával, amíg a kapott séma a BCNF-nak megfelelő nem lesz. A végeredmény a dekompozíciók eredményeinek uniója lesz.

Megj: Az eljárás biztosan véget ér, mert minden alkalommal a kapott relációk kevesebb attribútumot tartalmaznak és a 2 attribútumos már mindig BCNF.

# 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(név, 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>)

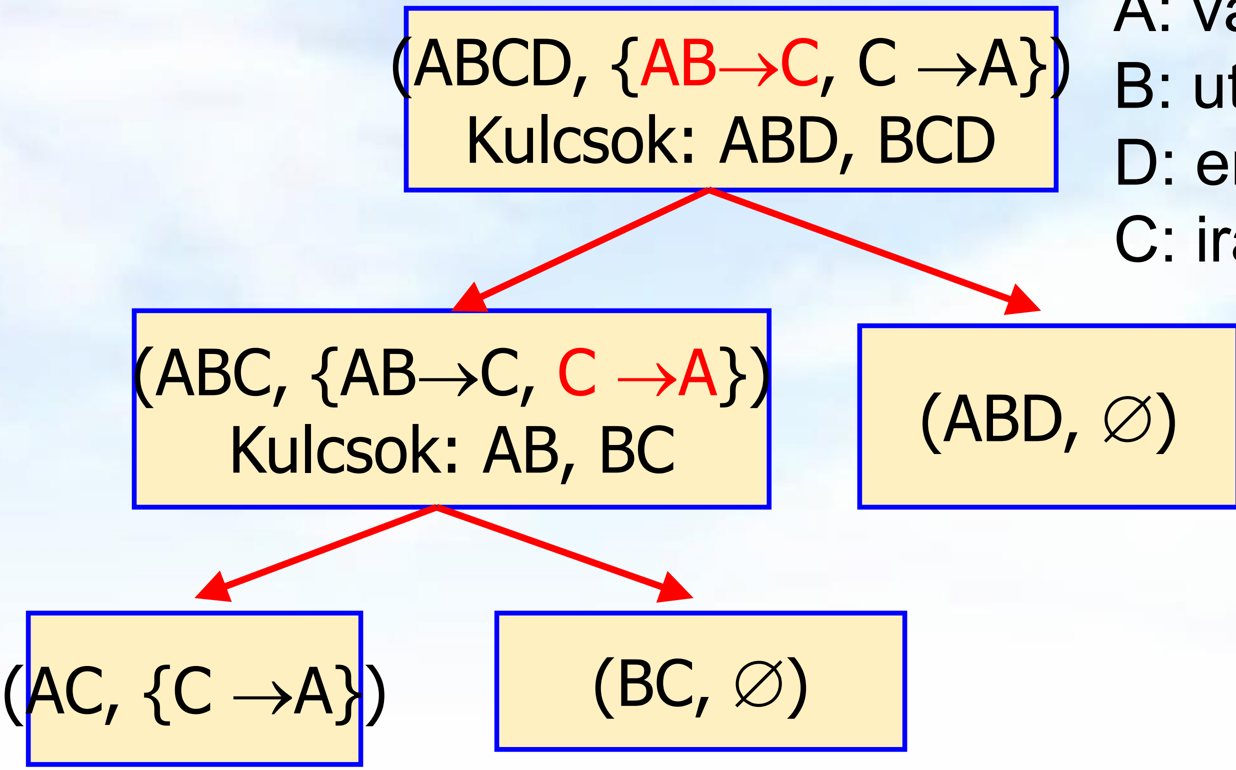
# Példa: BCNF dekompozíció ---4

- 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.

# Példa: BCNF-ra való felbontás

R=ABCD, F={AB→C, C →A}

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




Tehát d=(AC,BC,ABD) veszteségmentes BCNF dekompozíció. (∅ 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!

# (Tk.3.4.) Felbontásra vonatkozó elvárások

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

(2) Veszteségmentes legyen a felbontás, vagyis vissza tudjuk állítani az eredeti relációt a dekompozícióval kapott relációk soraiból.

(3) Függőségek megőrzése a vetületekben (FŐ)

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

# (Tk.3.4.1.)Veszteségmentes szétvágás

- 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 |

# Egy példa: Nem veszteségmentes

- Megjegyzés: Könnyen belátható, hogy $r \subseteq \Pi_{R_1}(r) \bowtie \ldots \bowtie \Pi_{R_k}(r)$ mindig teljesül.
- Példa: Vizsgáljuk meg R(A,B,C) relációt, amelyre a B->A és B->C függőségek egyike sem teljesül. A szétvágás után keletkező relációk természetes összekapcsolásával 4 sort, vagyis két hamis sort kapunk, amelyek nem voltak az eredeti R-ben.

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 |

# (Tk.3.4.2.) Chase-teszt VM ellenőrzése

- 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 ---4

- 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 = { $B \rightarrow AD$ }, 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 |

$B \rightarrow AD$ ⇒

| 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.

# (Tk.3.4.4.) Függőségek megőrzése függőségőrző felbontás

- Függőségőrző felbontás: a dekompozíciókban érvényes függőségekből következzen az eredeti sémára kirótt összes függőség, vagyis
- Definíció: Adott (R,F) séma esetén $d=(R_1,\ldots,R_k)$ felbontás akkor és csak akkor függőségőrző, ha minden F-beli függőség levezethető a vetületi függőségekből:

  minden $X \to Y \in F$ esetén

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

# Emlékeztető: Függőségőrző felbontás Itt mit értünk függőségek vetületén? (Tk.3.2.8.) Függőségek vetítése

- Definíció: Funkcionális függőségek vetítése

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

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

- Példa-1: *R=ABC, A ->B* és *B ->C* FF-kel.
  Nézzük meg az R1=*AC*-re való vetületet

# Példa-1: Függőségek vetítése

- $ABC$, $A \to B$ 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: $A \to C$.

# Példa-2: Függőségőrző felbontás

- 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?
- *AB* -re projekció: *A ->B* és *B ->A* is.
- *BC* -re projekció: *B ->C* és *C ->B* is.
- $\Pi_{R1}(F) \cup \ldots \cup \Pi_{Rk}(F) \not\vdash C \to A$.

# Függőségőrzés - veszteségmentesség

- 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→B}, d={AC,BC} veszteségmentes, de nem függőségőrző.

# A harmadik normálforma motiváció

- Bizonyos FF halmazok esetén a felbontáskor elveszíthetünk függőségeket.
- *AB ->C* és *C ->B*.
  - Példa: *A* = utca, *B* = város, *C* = irányítószám.
- Két kulcs van: {*A*,*B* } és {*A*,*C* }.
- *C ->B* megsérti a BCNF-t, tehát *AC*, *BC*-re dekomponálunk.
- A probléma az, hogy*AC* és *BC* sémákkal nem tudjuk kikényszeríteni *AB ->C* függőséget.

# 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.

# A 3NF és BCNF felbontások

- A dekompozícióknak két fontos tulajdonsága:
  1. Veszteségmentes összekapcsolás: ha a részsémákra vetített relációkból természetes összekapcsolással az eredetit kapjuk vissza.
  2. Függőségek megőrzése: a vetített 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)*

# Kérdés/Válasz

- Köszönöm a figyelmet! Kérdés/Válasz?
- Feladatok: Tankönyv 3.fejezetében az egyes szakaszok után található gyakorló feladatok
- Mai 11.előadáson: Feldolgoztuk a Tankönyv 3.3.szakaszát és a 3.4.szakasz nagy részét, folytatás, a következő 12.előadáson tárgyaljuk:
  - 3.2.7.Minimális bázis (3NF-ra felbontásánál)
  - 3.5. A harmadik normálforma
  - 3.6. Többértékű függőségek