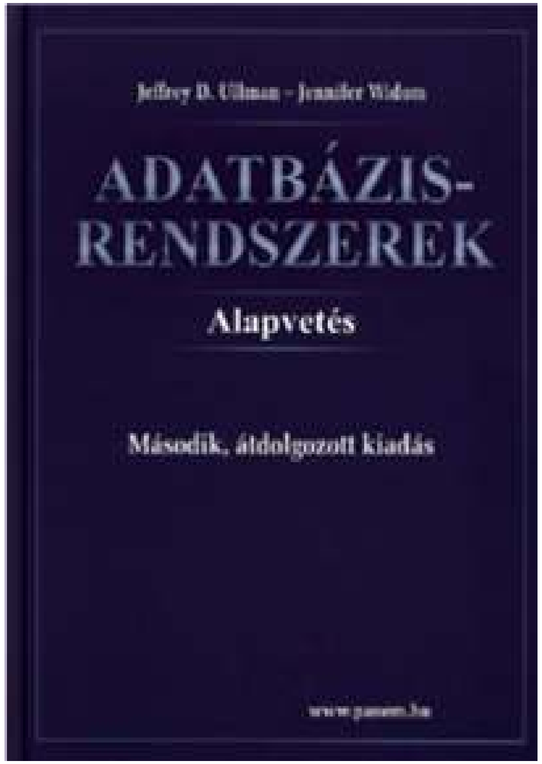
Relációs adatbázisok tervezése ---3

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 (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)

Függőségek megőrzése

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

Függőségek megőrzése (definíció)

- Definíció: Függőségek vetülete

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

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

- 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 $X \rightarrow Y \in F$ esetén

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

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* ->*C* és *C* ->*B*.
  - Példa1: *A* = utca, *B* = város, *C* = irányítószám.
  - Példa2: *A* = oktató, *B* = időpont, *C* = kurzus.
- 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.

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 elhyagunk egy vagy több attribútumot, akkor az eredmény már nem marad bázis.

Minimális bázist kiszámító algoritmus

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

Mohó algoritmus minimális bázis előállítására

1. Jobb oldalak minimalizálása:

   $X \rightarrow A1, \ldots, Ak$ 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→A,X→B,...,Y→C,Y→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 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 $G:=\{X\to A, X\to B, \ldots, Y\to C, Y\to D, \ldots.\}$ 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.

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.