

Matija Varga, mag. inf. univ. spec. oec.  
Srednja škola Sesvete  
Učiteljski fakultet Zagreb (vanjski suradnik)  
maavarga@gmail.com

UIDK 004.65  
Pregledni članak

# BERNSTEINOV ALGORITAM ZA VERTIKALNU 3NF NORMALIZACIJU SINTEZOM

## SAŽETAK

Ovim radom se prikazuje korištenje Bernsteinovog algoritma za vertikalnu 3NF normalizaciju sintezom. Cilj ovog rada je ponuditi algoritam za normalizaciju baze podataka te prikazati skup koraka koji vode na svodenje redundantnosti na najmanju moguću mjeru u cilju povećanja učinkovitosti rada s bazom podataka te navesti testove i algoritme za ispitivanje i dokazivanje reverzibilnosti (tj. dokazivanje da tijekom normalizacije nije došlo do gubitka informacije). U radu su prikazani primjeri kroz korake Bernsteinovog algoritma za vertikalnu 3NF normalizaciju sintezom te su predloženi test i algoritam za dokazivanje reverzibilnosti dekompozicije. Svrha ovog rada je također objasniti da su razlozi uvođenja normalnih formi: brže pretraživanje podataka, eliminacija redundancije i eliminacija anomalije brisanja, upisivanja i modifikacije te objasniti nastanak anomalija na temelju primjera.

**Ključne riječi:** funkcijska zavisnost, relacijska shema, normalizacija, normalizacija sintezom, dekompozicija, Bernsteinov algoritam, gubitak informacija.

## 1. Uvod

Kod baza podataka vrlo je bitno pridržavati se svojstva baze podataka a to su: (1) ne postoje dva jednaka redka, (2) redosljed redaka nije bitan, (3) ne postoje dva jednaka stupca, (4) redosljed stupaca nije bitan. Uzimajući u obzir relacijsku teoriju potrebno se je strogo pridržavati prvog i trećeg pravila (ne postoje dva jednaka redka i ne postoje dva jednaka stupca). Cilj svih algoritama za normalizaciju baze podataka je prikazati skup koraka koji vode na svodenje redundantnosti na najmanju moguću mjeru u

cilju povećanja učinkovitosti rada s bazom podataka. Jedan od poznatijih algoritama za normalizaciju baze podataka je Bernsteinov algoritam za vertikalnu 3NF normalizaciju sintezom čiji će koraci ovim radom biti prikazani i primijenjeni. Algoritam je pojam koji je osmislio arapski matematičar Muhamed ibn Musa Al Horezmi (u prijevodu: Muhamed sin Muse iz Horezma). Muhamed je napisao knjigu u kojoj je razradio postupke i pravila za provođenje aritmetičkih operacija s brojevima zapisanim u dekadskom obliku.<sup>1</sup> Normalizacija baze podata-

1 Algoritmi i programiranje računala. URL: [http://informatika.efos.hr/wp-content/uploads/2012/04/P3\\_Algoritmi-i-Programiranje](http://informatika.efos.hr/wp-content/uploads/2012/04/P3_Algoritmi-i-Programiranje)

ka je postupak kojim se nenormalizirana relacijska shema transformira uvođenjem novih ograničenja. Uvođenjem novih ograničenja baza podataka postaje normalizirana. U okviru normalizacije ne smije doći do nepovratnog gubitka podataka tj. informacija. Normalizacija je postupak transformiranja relacijske sheme iz jedne normalne forme u drugu. Teorija normalizacije objašnjava anomalije postojanjem redundantnih podataka u bazi podataka te donosi pravila sustavne eliminacije redundancije iz baze podataka. Potpuno normalizirana baza podataka nema redundancije i loži zalihosti te se u njoj pouzdano i jednostavno obavljaju operacije s podacima. Tijekom fizičkog modeliranja podataka se može uvesti kontrolirana redundancija ili zalihosti podataka u svrhu očuvanja integriteta podataka i postizanja povećane brzine pristupa podacima.

Teorija normalizacije ustanovila je više normalnih formi pojedine relacije, odnosno čitave baze podataka. Najpoznatije normalne forme su: prva normalna forma (1NF), druga normalna forma (2NF), treća normalna forma (3NF), Boyce-Coddova normalna forma (BCNF), četvrta normalna forma (4NF) i peta normalna forma (5NF). Prva normalna forma je najniža, odnosno najblaža, dok je peta najviša odnosno najstroža normalna forma. Svaka viša normalna forma obuhvaća ujedno i sve niže normalne forme.<sup>2</sup> Relacijska shema je u 3NF i BCNF ako je naravno u 1NF itd.

## 2. Bernsteinov algoritam za normalizaciju sintezom

Bernsteinov algoritam za vertikalnu normalizaciju sintezom je definiran 1976. godine te je prilagođen za računalnu obradu. Bernsteinovim algoritmom je izvedena modifikacija. Bernsteinov algoritam za normalizaciju polazi od skupa funkcijskih zavisnosti. Skup funkcijskih zavisnosti je nezalihan (neredundantan) ako ne postoji njegov podskup koji bi dao isti zatvarač. U postupku normalizacije sintezom traži se skup ili skupovi funkcijskih zavisnosti koji imaju minimalan broj funkcijskih zavisnosti u odnosu na bilo koji ekvivalentni skup, istovremeno su oni neredundantni ili nezalihan.

Normalizacija baze podataka se može izvršiti sintezom i dekompozicijom (razdiobom). Prikaz Bernsteinovog algoritma za normalizaciju sintezom:

Ulaz: Skup funkcijskih zavisnosti<sup>3</sup> F, zadan na R.

Izlaz: Komplet relacijskih shema<sup>4</sup> 3NF<sup>5</sup> za F.

### SINTEZA (F)

1. Nadite atribut<sup>6</sup> Z tako da vrijedi  $Z \notin R$ .
2. Konstruirajte funkciju zavisnost  $R \rightarrow Z$  i dodajte ju u F.
3. Nadite reducirani prstenasti pokrivač<sup>7</sup> G za prošireni F.
4. Neredundantni pokrivač.
5. Reducirani pokrivač.
6. Za svaku SFZ  $(X_1, X_2, X_3, \dots, X_n) \rightarrow Y$  u G konstruirajte relacijsku shemu  $R_i(X_1, X_2, \dots, X_n)$  s ključevima  $X_1, X_2, \dots, X_n$ .
7. Izbacite atribut Z iz relacijske sheme  $R_i$ .
8. Ispišite skup relacijskih shema u 3NF.

Koraci jedan i dva u Bernsteinovom algoritmu su potrebni da bi se osigurala reverzibilnost tj. dokaz da tijekom normalizacije nije došlo do gubitka podataka te na kraju informacije koja je ključna za krajnjeg korisnika poslovnog informacijskog sustava.

### 2.1. Zatvarač skupa atributa

Desna strana zavisnosti  $X \rightarrow Y$ , naziva se **zatvarač skupa** X i označava se s  $X^+$ . Zatvarač skupa atributa se može dovoljno brzo odrediti. Skup  $X^+$  nazivamo zatvaračem skupa F.

- 3 Neka je r(R) relacija i neka su X,Y podskupovi od R. Kažemo da u r, Y funkcijski zavisi o X i pišemo  $X \rightarrow Y$ .
- 4 Relacijska shema je model podataka zasnovan na teoriji skupova koji predstavlja formu po kojoj su pohranjeni podaci (a ne same podatke).
- 5 Treća normalna forma.
- 6 Atribut je svojstvo koje opisuje objekt.
- 7 Reducirani prstenasti pokrivač

### 2.2. Pokrivač skupa funkcijskih zavisnosti

**Pokrivač skupa** definira se pomoću pojma ekvivalencije<sup>8</sup> skupova funkcijskih zavisnosti. Dva skupa funkcijskih zavisnosti F i G zadanih na skupu atributa R međusobno su ekvivalentni ako vrijedi da su im zatvarači jednaki ( $F^+ = G^+$ ). Skup funkcijskih zavisnosti F smatramo pokrivačem skupa zavisnosti G ako vrijedi ( $F^+ = G^+$ ).

### 2.3. Sastavljena funkcijska zavisnost

Sastavljena funkcijska zavisnost zadana na skupu atributa imenovanim znakom R i ima oblik  $(X_1, X_2, X_3, \dots, X_n) \rightarrow Y$ . Sastavljena funkcijska zavisnost (SFZ) vrijedi, ako je  $X_1, X_2, \dots, X_n, Y$  podskup od R.

### 2.4. Prstenasti pokrivač

Za skup sastavljene funkcijske zavisnosti kažemo da je **prstenast** ako za bilo koja dva međusobno ekvivalentna podskupa atributa X i Z vrijedi da ne mogu biti članovi lijevih strana dviju različitih (SFZ) sastavljenih funkcijskih zavisnosti u F.

Neka je G nezalihan (neredundantan)<sup>9</sup> skup sastavljene funkcijske zavisnosti (SFZ). SFZ je **reduciran** ako niti jedan član u lijevoj strani ne sadrži niti jedan pomični atribut i ako desna strana SFZ ne sadrži niti jedan suvišni atribut. Skup sastavljene funkcijske zavisnosti G je reducirana ako su sve sastavljene funkcijske zavisnosti u G reducirane.

### 2.4.1. Primjer pronalaska prstenastog pokrivača

Potrebno je pronaći prstenasti pokrivač G za skup FZ  $F = \{AB \rightarrow D, AD \rightarrow E, AF \rightarrow C, D \rightarrow F, F \rightarrow B\}$ .

**1. korak** – pronaći klase ekvivalencije koje čine particiju u F:

$AB_F^+ = ABCDEF$

$AD_F^+ = ABCDEF$

$AF_F^+ = ABCDEF$

$D_F^+ = BDF$

$F_F^+ = BF$

**2. korak** – particija  $E_F$  ima tri podskupa:

$E_F(AB) = \{AB \rightarrow D, AD \rightarrow E, AF \rightarrow C\}$

$E_F(D) = \{D \rightarrow F\}$

$E_F(F) = \{F \rightarrow B\}$

**3. korak** – za svaki podskup tj. klasu ekvivalencije konstruiramo jednu SFZ. Lijeva strana se sastoji od lijevih strana FZ jedne klase, a desna je unija<sup>10</sup> desnih strana:

$(AB, AD, AF) \rightarrow CDE$

$(D) \rightarrow F$

$(F) \rightarrow B$

Primjer.

Unija je operacija sa skupovima koja se primjenjuje nad dvjema relacijama koje imaju istu shemu. Nakon primjene operacije unije novonastala relacija sadrži attribute sa skupom podataka koji se nalaze i u jednoj i u drugoj relaciji iz kojih je nastala treća relacija. Algoritam za operaciju unije glasi: ULAZ: Relacije m i n, IZLAZ:  $m \cup n$ .

1. Provjeriti da li je  $Sh(m) = Sh(n)$ .

\*Ako nije, ispisati pogrešku i stati

2. Otvoriti praznu relaciju  $m \cup n$  sa shemom  $Sh(m)$

3. Sve slogove iz r prepisati u  $m \cup n$

4. Za svaki slog iz s učiniti sljedeće:

Provjeriti nalazi li se taj slog i u m.

Ako se taj slog ne nalazi u r, dodati ga u  $m \cup n$

<sup>10</sup> Unija je operacija nad dvije relacije koje imaju istu shemu,

miranjet.pdf. (13.7.2012.).

2 Varga Mladen, Upravljanje podacima, Element, 1. Izdanje. Zagreb, 2012.g.

**4. korak** - ovako dobivene SFZ<sup>11</sup> su elementi pokrivača G, tj.

$$G = \{(AB, AD, AF) \rightarrow CDE, (D) \rightarrow F, (F) \rightarrow B\}$$

#### 2.4.2. Reducirani prstenasti pokrivač

Reducirani prstenasti pokrivač se dobije tako da prvo pronađemo neredundantan ili nezalihan pokrivač te nakon toga pronađemo L-reducirani pokrivač, L-reducirani prstenasti pokrivač, D-reducirani prstenasti pokrivač. Primjer: Zadan je skup FZ,  $F = \{A \rightarrow C, B \rightarrow C\}$ . Treba pronaći nezalihan (neredundantni) pokrivač skupa F.  $G_1 = \{A \rightarrow C, B \rightarrow C\}$ . Nakon toga promatramo funkcijsku zavisnost (FZ)  $A \rightarrow C, G' = \{B \rightarrow C\}, AG' = AC$ . Zavisnost  $A \rightarrow C$  nije redundantna ili zalihosna, a zavisnost  $B \rightarrow C$  isto nije redundantna ili zalihosna. Skup zavisnosti koje nisu redundantne ili zalihosne su  $G_2 = \{A \rightarrow C, B \rightarrow C\}$ . Slijedi lijevo reduciranje.  $G_3 = \{A \rightarrow C, B \rightarrow C\}$ .  $G_3 = \{A \rightarrow C, B \rightarrow C\}$  (oznaka za prazan skup), pa atribut A nije suvišan. Isto vrijedi i za atribut B, niti atribut B nije suvišan. Lijevo reducirani pokrivač skupa od  $G_3$  je  $G_4, G_4 = \{A \rightarrow C, B \rightarrow C\}$ . Lijevo reducirani prstenasti pokrivač je  $G_5 = \{(A, B) \rightarrow C\}$ . Približno D-reducirani prstenasti pokrivač  $G_6 = \{(A, B) \rightarrow C\}$ .

##### 2.4.2.1. Neredundantan pokrivač

Primjer pronalaska neredundantnog pokrivača: Zadan je skup FZ  $F: AB \rightarrow E, AB \rightarrow CF, AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ . Treba naći neredundantan pokrivač od F.

1.  $G: AB \rightarrow E, AB \rightarrow CF, AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .
2. Promatramo FZ  $AB \rightarrow E, G': AB \rightarrow CF, AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ . Tražimo ključ  $ABG' = ABCDEF$ , pa je zavisnost  $AB \rightarrow E$  redundantna. Nakon toga stavljam:  $G': AB \rightarrow CF, AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ . 1.  $G: AB \rightarrow E, AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .
3. Promatramo FZ  $AB \rightarrow E, G': AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .  $ABG' = ABCDEF$  iz čega zaključujemo da je  $AB \rightarrow E$  redundantna. Zato stavljam:  $G: AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .
4. Promatramo FZ  $AC \rightarrow F, G': AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .  $ACG' = ABCDE$  iz čega zaključujemo da  $AC \rightarrow F$  nije redundantna.  $G: AC \rightarrow F, AB \rightarrow E, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .

5. Promatramo FZ  $AB \rightarrow E, G': AC \rightarrow F, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .  $ABG' = ABCDEF$  iz čega zaključujemo da je  $AB \rightarrow E$  redundantna. Stoga stavljam:  $G: AC \rightarrow F, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .  $G: AC \rightarrow F, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .
6. Promatramo FZ  $AD \rightarrow B, G': AC \rightarrow F, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .  $ADG' = ADE$  iz čega zaključujemo da  $AD \rightarrow B$  nije redundantna.
7. Promatramo FZ  $B \rightarrow C, G': AC \rightarrow F, AD \rightarrow B, D \rightarrow E, C \rightarrow D$ .  $BG' = B$  iz čega zaključujemo da zavisnost  $B \rightarrow C$  nije redundantna.
8. Promatramo FZ  $D \rightarrow E, G': AC \rightarrow F, AD \rightarrow B, B \rightarrow C, C \rightarrow D$ .  $DG' = D$  iz čega zaključujemo da zavisnost  $D \rightarrow E$  nije redundantna.
9. Promatramo FZ  $C \rightarrow D, G': AC \rightarrow F, AD \rightarrow B, B \rightarrow C, D \rightarrow E$ .  $CG' = C$  iz čega zaključujemo da zavisnost  $C \rightarrow D$  nije redundantna. Neredundantan pokrivač od F je  $G: AC \rightarrow F, AD \rightarrow B, B \rightarrow C, D \rightarrow E, C \rightarrow D$ .

##### 2.4.2.2. L-reducirani pokrivač

Primjer: Zadan je skup FZ (funkcijski zavisnosti).  $F: AB \rightarrow DF, ADF \rightarrow BCE, AF \rightarrow BC, D \rightarrow F, F \rightarrow B$ . Treba naći lijevo reducirani pokrivač skupa F.  $G: AB \rightarrow DF, ADF \rightarrow BCE, AF \rightarrow BC, D \rightarrow F, F \rightarrow B$ .

1. Provjeravamo reduciranost FZ

$AB \rightarrow DF, A_G^+ = A, DF \not\subseteq A_G^+$  pa atribut A nije suvišan u ovoj FZ.  $B_G^+ = B, DF \not\subseteq B_G^+$  pa atribut B nije suvišan u ovoj FZ.  $G: AB \rightarrow DF, ADF \rightarrow BCE, AF \rightarrow BC, D \rightarrow F, F \rightarrow B$ .

2. Provjeravamo reduciranost FZ

$ADF \rightarrow BCE, DF_G^+ = BDF, BCE \not\subseteq DF_G^+$  pa atribut A nije suvišan.  $AF_G^+ = ABCDEF, BCE \subseteq AF_G^+$  pa je atribut D suvišan i imamo  $G: AB \rightarrow DF, AF \rightarrow BCE, AF \rightarrow BC, D \rightarrow F, F \rightarrow B$ .  $A_G^+ = A, BCE \not\subseteq A_G^+$  pa atribut F nije suvišan.

3. Provjeravamo reduciranost FZ

$AF \rightarrow BC, A_G^+ = A, BC \not\subseteq A_G^+$  pa atribut F nije suvišan.  $F_G^+ = BF, BC \subseteq F_G^+$  pa atribut A nije suvišan.

4. Provjeravamo reduciranost FZ

$D \rightarrow F, O_G^+ = O, F \not\subseteq O_G^+$  pa atribut D nije suvišan.  $G: AB \rightarrow DF, AF \rightarrow BCE, AF \rightarrow BC, D \rightarrow F, F \rightarrow B$ . 5. Provjeravamo reduciranost FZ  $F \rightarrow B, O_G^+ = O, F \not\subseteq O_G^+$  pa atribut F nije suvišan.

##### 2.4.2.3. D-reducirani pokrivač

$G: AB \rightarrow DF, AF \rightarrow BC, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .

1. Prvo provjeravamo desnu reduciranost FZ

$AB \rightarrow DF, G': AB \rightarrow D, AF \rightarrow BC, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .  $ABG' = ABCDEF, F \supseteq ABG'$  pa je atribut F suvišan i imamo  $G: AB \rightarrow D, AF \rightarrow BC, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .  $G: AB \rightarrow D, AF \rightarrow BC, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .

2. Prvo provjeravamo desnu reduciranost FZ

$AB \rightarrow DF, G': AB \rightarrow O, AF \rightarrow BC, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .  $ABG' = AB, D \not\subseteq ABG'$ , pa atribut D nije suvišan.  $G: AB \rightarrow D, AF \rightarrow BC, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .

3. Provjeravamo desnu reduciranost FZ

$AD \rightarrow BC, G': AB \rightarrow D, AF \rightarrow B, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .  $AFG' = ABCDEF, BC \subseteq AFG'$  pa je atribut C suvišan, tako da imamo.  $G: AB \rightarrow D, AF \rightarrow B, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .  $G: AB \rightarrow D, AF \rightarrow B, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .

4. Provjeravamo desnu reduciranost FZ

$AD \rightarrow BC, G': AB \rightarrow D, AF \rightarrow O, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .  $AFG' = ABCDEF, B \subseteq AFG'$  pa je atribut B suvišan i imamo  $G: AB \rightarrow D, AF \rightarrow O, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$  odnosno, ako izbacimo zavisnost s praznom desnom stranom.  $G: AB \rightarrow D, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .  $G: AB \rightarrow D, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .

5. Provjeravamo desnu reduciranost FZ

$AF \rightarrow BCE, G': AB \rightarrow D, AF \rightarrow BC, D \rightarrow F, F \rightarrow B$ .  $AFG' = ABCDEF, BCE \not\subseteq AFG'$  pa atribut E nije suvišan.  $G': AB \rightarrow D, AF \rightarrow BE, D \rightarrow F, F \rightarrow B$ .  $AFG' = ABDEF, BCE \not\subseteq AFG'$  pa atribut C nije suvišan.  $G: AB \rightarrow D, AF \rightarrow BCE, D \rightarrow F, F \rightarrow B$ .

6. Provjeravamo desnu reduciranost FZ

$AF \rightarrow BCE, G': AB \rightarrow D, AF \rightarrow CE, D \rightarrow F, F \rightarrow B$ .  $AFG' = ABCDEF, BCE \subseteq AFG'$  pa je atribut B suvišan i imamo  $G: AB \rightarrow D, AF \rightarrow CE, D \rightarrow F, F \rightarrow B$ .  $G: AB \rightarrow D, AF \rightarrow CE, D \rightarrow F, F \rightarrow B$ .

7. Provjeravamo desnu reduciranost FZ

$D \rightarrow F, G': AB \rightarrow D, AF \rightarrow CE, D \rightarrow O, F \rightarrow B$ .  $DG' = D, F \subseteq DG'$  pa atribut F nije suvišan.  $G: AB \rightarrow D, AF \rightarrow CE, D \rightarrow F, F \rightarrow B$ .

8. Provjeravamo desnu reduciranost FZ

$F \rightarrow B, G': AB \rightarrow D, AF \rightarrow CE, D \rightarrow F, F \rightarrow O$ .  $FG' = F, B \subseteq FG'$  pa atribut B nije suvišan. Dakle dobili smo da je desno reducirani pokrivač od polaznog skupa skup  $G: AB \rightarrow D, AF \rightarrow CE, D \rightarrow F, F \rightarrow B$ .

#### 2.5. Vertikalna normalizacija sintezom

Vertikalna normalizacija dijeli se na normalizaciju dekompozicijom i normalizacijom sintezom. U ovom radu prikazat će se normalizacija sintezom i normalizacija uz pomoć dekomponiranja (razdiobe)<sup>12</sup>. Normalizacija sintezom je novija tehnika koja sve više dolazi do izražaja. Polazište normalizacije sintezom je skup atributa i skup zavisnosti zadan na tom skupu. Formalnim postupkom konstruiraju se relacijske sheme u traženoj normalnoj formi. Razvijeni su postupci koji omogućavaju sintezu relacijske sheme u trećoj normalnoj formi. Zasad ne postoji pouzdan algoritam za formiranje relacijskih shema u višim normalnim formama.<sup>13</sup> Normalizacija sintezom omogućuje da polazeći od skupa atributa i skupa funkcijskih zavisnosti koji je na tom skupu atributa zadan konstruiramo skup relacijskih shema u 3NF. Kao rezultat normalizacije sintezom dobiva se 3NF.<sup>14</sup> 3NF podrazumijeva 2NF i 1NF. (R,F) može biti u 3NF, a da i dalje postoji redundancija ili zalihost. Primjer 3NF sinteze: Neka je zadana shema R: BCDEFI.  $F \rightarrow BC, BC \rightarrow I, ID \rightarrow EF, BCD \rightarrow EF$ . Potrebno je naći 3NF koja čuva informaciju.  $F \rightarrow BC, BC \rightarrow I, ID \rightarrow EF, BCD \rightarrow EF$ .

1. Vršimo desno razbijanje

$G: I \rightarrow B, I \rightarrow C, BC \rightarrow I, ID \rightarrow E, ID \rightarrow F, BCD \rightarrow E, BCD \rightarrow F$ .

2. Lijevo reduciramo G

- a)  $OG = O$ , pa su zavisnosti  $I \rightarrow B$  i  $I \rightarrow C$  lijevo reducirane.
- b)  $BG = B$   
 $CG = C$  pa je zavisnost  $BC \rightarrow I$  lijevo reducirana.
- c)  $IG = I$   
 $DG = D$  pa su zavisnosti  $ID \rightarrow E$  i  $ID \rightarrow F$  lijevo reducirane.  
 $BCG = IBC$

<sup>12</sup> Dekompozicija je postupak raščlanjivanja, razgradnje, razdiobe složenih struktura na jednostavnije. Prilikom dekomponiranja potrebno je pridržavati se pravila da svaki roditelj mora imati minimalno barem dva djeteta.

<sup>13</sup> Skočir, Z.; Matasić, I.; Vrdoljak, B. Organizacija obrade podataka. Izd. 1. Merkur A.B.D.; FER, Udžbenik Sveučilišta u Zagrebu, 2007.g. str. 48.

<sup>14</sup> Slavko Tkalac, Relacijski model podataka, DRIP, Biblioteka informacijsko društvo 11, Zagreb, 1993.g. str. 133.

<sup>11</sup> SFZ – sastavljena funkcijska zavisnost.

$BDG^+ = BD$

$CDG^+ = CD$  pa su zavisnosti  $BCD \rightarrow E$  i  $BCD \rightarrow F$  lijevo reducirane.

### 3. Izbacujemo redundantne (zalihosne) zavisnosti

a) Prvo provjeravamo zavisnost  $I \rightarrow B$

$G': I \rightarrow C, BC \rightarrow I, ID \rightarrow E, ID \rightarrow F, BCD \rightarrow E, BCD \rightarrow F$

$IG'^+ = IC$ . pa zavisnost  $I \rightarrow B$  nije redundantna

(zalihosna).

b) sada provjeravamo zavisnost  $I \rightarrow C$

$G': I \rightarrow B, BC \rightarrow I, ID \rightarrow E, ID \rightarrow F, BCD \rightarrow E, BCD \rightarrow F$

$IG'^+ = IB$ , pa zavisnost  $I \rightarrow C$  nije redundantna

(zalihosna).

c) Provjeravamo zavisnost  $BC \rightarrow I$

$G': I \rightarrow B, I \rightarrow C, ID \rightarrow E, ID \rightarrow F, BCD \rightarrow E, BCD \rightarrow F$

$BCG'^+ = BC$ , pa zavisnost  $BC \rightarrow I$

nije redundantna (zalihosna).

d) provjeravamo zavisnost  $ID \rightarrow E$

$G': I \rightarrow B, I \rightarrow C, BC \rightarrow I, ID \rightarrow F, BCD \rightarrow E, BCD \rightarrow F$

$IDG'^+ = IBCDEF$ , pa je zavisnost  $ID \rightarrow E$

redundantna (zalihosna) i imamo

$G: I \rightarrow B, I \rightarrow C, BC \rightarrow I, ID \rightarrow F, BCD \rightarrow E, BCD \rightarrow F$ .

e) provjeravamo zavisnost  $ID \rightarrow F$

$G': I \rightarrow B, I \rightarrow C, BC \rightarrow I, BCD \rightarrow E, BCD \rightarrow F$

$IDG'^+ = IBCDEF$ , pa je zavisnost  $ID \rightarrow F$

redundantna (zalihosna) i imamo

$G: I \rightarrow B, I \rightarrow C, BC \rightarrow I, BCD \rightarrow E, BCD \rightarrow F$

f) Sada provjeravamo zavisnost  $BCD \rightarrow E$

$G': A \rightarrow B, I \rightarrow C, BC \rightarrow I, BCD \rightarrow F$

$BCD_{G'}^+ = IBCDEF$ , pa zavisnost  $BCD \rightarrow E$

nije redundantna (zalihosna)

g) i na kraju provjeravamo zavisnost  $BCD \rightarrow F$

$G': I \rightarrow B, I \rightarrow C, BC \rightarrow I, BCD \rightarrow E$

$BCD_{G'}^+ = IBCDE$ , pa zavisnost  $BCD \rightarrow F$

nije redundantna (zalihosna)

$G: I \rightarrow B, I \rightarrow C, BC \rightarrow I, BCD \rightarrow E, BCD \rightarrow F$

### 4. Napravimo dekompoziciju

$I \rightarrow B$  i  $I \rightarrow C$  daju shemu  $\underline{IBC}$

$BC \rightarrow I$  daje shemu  $\underline{BCI}$

$BCD \rightarrow E$  i  $BCD \rightarrow F$  daju shemu  $\underline{BCDEF}$

pa imamo

$d(R): \underline{IBC}, \underline{BCI}, \underline{BCDEF}$

### 5. Pronađimo ključ od (R,F)

$Xk = D$

$D_{G'}^+ = D$

Vidimo da su ključevi od (R,F)

$K = ID, BCDA$  kako je  $BCD \subseteq \underline{BCDEF}$  nije potrebno dodavati novu komponentu.

$G: I \rightarrow B, I \rightarrow C, BC \rightarrow I, BCD \rightarrow E, BCD \rightarrow F$

$d(R): \underline{IBC}, \underline{BCI}, \underline{BCDEF}$

### 6. Prve dvije komponente su jednake, pa možemo

jednu maknuti i imamo

$d(R): \underline{IBC}, \underline{BCDEF}$

To je traženo 3NF rješenje.

Razlozi uvođenja normalnih formi su brže pretraživanje podataka, eliminacija redundancije ili zalihosti i eliminacija anomalija brisanja, upisivanja i modifikacije. Eliminacija redundancije ili zalihosti je vrlo bitna. To je slučaj ako je neka informacija memorirana u bazi podatka u više kopija. U tom slučaju uvijek postoji mogućnost da neka od njih ostane neažurirana, tada ćemo imati dvije suprotne tvrdnje što nikako nije dobro za krajnjeg korisnika informacije jer je u neizvjesnosti. Anomalija upisivanja javlja se u onim slučajevima kada su informacije o atributima jednog entiteta memorirane u bazi kao dio opisa nekog drugog entiteta. Na primjer u okviru opisa atributa jednog zaposlenika memorirane informacije o općini u kojoj zaposlenik boravi. Informacije u općini nije moguće unijeti u bazu tako dugo dok ne postoji barem jedan zaposlenik koji u toj općini boravi. S obzirom na to da se prikupljene informacije o općini odvijaju potpuno neovisno o prikupljanju podataka o zaposleniku, ovakvo rješenje u bazi podataka može imati za posljedicu gubitak podataka i informacija. Za anomaliju brisanja možemo reći da je inverzija anomalije dodavanja.

### 3. Primjena koraka Bernsteinovog algoritma

Primjer 1.

Neka je  $F = \{I \rightarrow C, B \rightarrow C\}$  skup zavisnosti zadan na  $R(IBC)$ . Atribut  $Z$  se ne nalazi u relacijskoj shemi. Konstruirat ćemo zavisnost  $IBC \rightarrow Z$  i dodati je u  $F$ . Na taj način dobili smo prošireni skup zavisnosti označen s:  $F' = \{I \rightarrow C, B \rightarrow C, IBC \rightarrow Z\}$ . Reducirani prstenasti pokrivač od  $F'$  je  $G = \{(I) \rightarrow C, (B) \rightarrow C, (IB) \rightarrow Z\}$ . Na osnovu toga konstruiraju se tri relacijske sheme:  $R_1(IC)$  s ključem  $I$ ,  $R_2(BC)$  s ključem  $B$  i

$R_3(IBZ)$  s ključem  $IB$ . Budući da je atribut  $Z$  dodan samo da bi se spriječilo izbacivanje novo konstruirane zavisnosti kao redundantne, možemo ga sad izbaciti. Relacijska shema  $R_3$  imala bi sad sljedeći izgled:  $R_3(IB)$  s ključem  $IB$ .

Primjer 2.

Bezuspješna primjena Bernsteinovog algoritma na skupu funkcijskih zavisnosti metodom "olovke i papira". Pronalazimo atribut  $V$  tako da vrijedi  $V \notin R$ .  $R(ABCDEF)$ .  $F = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD\}$ . Konstruiramo funkciju zavisnosti  $ABCDEF \rightarrow V$ , te je dodajemo u  $G$  (u skup funkcijskih zavisnosti).  $G = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $AB \rightarrow E$ .  $G' = \{CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $AB_{G'}^+ = ABCDEF$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $CD \rightarrow F$ .  $G' = \{AB \rightarrow E, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $CD_{G'}^+ = ABCDE$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $A \rightarrow C$ .  $A_{G'}^+ = A$ .  $G = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $B \rightarrow D$ .  $B_{G'}^+ = B$ .  $G = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD, ABCDEF \rightarrow V\}$ .  $C \rightarrow A$ .  $C_{G'}^+ = C$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, ABCDEF \rightarrow V\}$ .  $F_{G'}^+ = F$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD\}$ .  $ABCDEF_{G'}^+ = ABCDEFV$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD\}$ . Lijevo reducirani pokrivač:  $F = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD\}$ .  $AB \rightarrow E$ .  $A_{F'}^+ = AC$ ,  $E \not\subseteq A_{F'}^+$  (nije suvišan atribut).  $B_{F'}^+ = BD$ ,  $E \not\subseteq B_{F'}^+$  nije suvišan atribut.  $B$ .  $G' = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD\}$ .  $CD \rightarrow F$ .  $C_{G'}^+ = AC$ ,  $F \not\subseteq C_{G'}^+$ .  $D_{G'}^+ = BD$ ,  $F \not\subseteq D_{G'}^+$ .  $F \rightarrow AD$ .  $F_{G'}^+ = ABCDEF$ ,  $AD \subseteq F_{G'}^+$ .  $F = \{AB \rightarrow E, CD \rightarrow F, A \rightarrow C, B \rightarrow D, C \rightarrow A, D \rightarrow B, F \rightarrow AD\}$ . Prstenasti pokrivač:  $AB_{F'}^+ = ABCDEF$ ,  $CD_{F'}^+ = ABCDEF$ ,  $F_{F'}^+ = ABCDEF$  (Ekvivalentni ključevi).  $A_{F'}^+ = AC$ ,  $B_{F'}^+ = BD$ .  $E_{F'}(AB) = \{AB \rightarrow E, CD \rightarrow F, F \rightarrow AD\}$ ,  $E_{F'}(A) = \{A \rightarrow C\}$ ,  $E_{F'}(B) = \{B \rightarrow D\}$ ,  $E_{F'}(C) = \{C \rightarrow A\}$ ,  $E_{F'}(D) = \{D \rightarrow B\}$ .  $(AB, CD, F) \rightarrow ADEF$ ,  $(A) \rightarrow C$ ,  $(B) \rightarrow D$ ,  $(C) \rightarrow A$ ,  $(D) \rightarrow B$ .  $SFZ_1(AB, CD, EF, AD)$ ,  $SFZ_2(A, C)$ ,  $SFZ_3(B, D)$ .  $R1(AB, CD, EF, AD)$ ,  $R2(A, C)$ ,  $R3(B, D)$ . Treća normalna forma (3NF) =  $R(R1, R2, R3)$ .

### 4. Zaključak

Bernsteinov algoritam za vertikalnu normalizaciju sintezom prikazuje korake koji se moraju izvršiti kako bi se postigla normalizacija relacija baze podataka tj. kako bi se eliminirala zalihost ili redundancija, ali s ciljem da se ne izgubi cjelovitost podataka i ključnih vrijednosnih informacija koje su bitne za funkcioniranje poslovnog informacijskog sustava. Drugim riječima normalizacijom se nastoji sačuvati integritet podataka u bazi podataka. Vertikalna normalizacija sintezom ne polazi od relacijske sheme, već od skupa atributa i skupa zavisnosti zadanah na tom skupu atributa. Iako je Bernsteinov algoritam jasan, izvedba niza koraka koje sadrži algoritam nije nimalo jednostavna. Zbog velikog broja atributa i funkcijskih zavisnosti, postupak pronalaska zatvarača je težak i vremenski zahtjevan problem. Izbor postupaka vertikalne normalizacije sintezom ovisan je o raspoloživim resursima (potencijalima). Normalizacija sintezom zahtijeva korištenje određene programske potpore. Za prikaz Bernsteinovog algoritam odlučili smo se iz razloga što nije toliko poznata normalna forma, a postiže se normalizacija baze podataka. Nakon provedene vertikalne normalizacije sintezom Bernsteinovim algoritmom, potrebno je ispitati je li tijekom normalizacije došlo do gubitka informacije ili nije. Rissanenov test može poslužiti za utvrđivanje gubitka informacije, ali Rissanenov test nije nužan u dokazu reverzibilnosti. Rissanen (1977. godine) napominje da je reverzibilnost osigurana kada dekompozicije imaju barem jedan zajednički atribut te da zajednički atribut treba predstavljati ključ barem u jednoj dekompoziciji. Također za ispitivanje i dokazivanje reverzibilnosti može poslužiti algoritam tako zvani Aho Beri Ullman algoritam. Algoritmom Aho Beri Ullman moguće je dokazati da je polazni skup informacija sačuvan. Pomoću Rissanenovog kriterija nezavisnosti komponenata moguće je dokazati također da je polazni skup informacija sačuvan i to na daleko jednostavniji i brži način. U postupku normalizacije sintezom može se dobiti skup relacijskih shema koje su prema algoritmu Aho Beri Ullmana reverzibilne, a ne zadovoljavaju Rissanenov test reverzibilnosti. Dakle, može se zaključiti da Rissanenov test nije nužan za dokaz reverzibilnosti dekompozicije (raziobe).

## LITERATURA

1. Mateljan Vladimir, URL: seminar.foi.hr/mateljan\_12\_09.pps (13.7.2012.).
2. Skočir, Z.; Matasić, I.; Vrdoljak, B. Organizacija obrade podataka. Izd. 1. Merkur A.B.D. ; FER, Udžbenik Sveučilišta u Zagrebu, 2007.g.
3. Tkalac Slavko, Relacijski model podataka, DRIP, Biblioteka informacijsko društvo 11, Zagreb, 1993.g.
4. Tkalac Slavko, Relacijski model podataka, Informator, Informatička biblioteka, Zagreb, 1988.g.
5. Varga Mladen, Upravljanje podacima, Element, 1. Izdanje. Zagreb, 2012.g.
6. Algoritmi i programiranje računala. URL: [http://informatika.efos.hr/wp-content/uploads/2012/04/P3\\_Algoritmi-i-Programiranje1.pdf](http://informatika.efos.hr/wp-content/uploads/2012/04/P3_Algoritmi-i-Programiranje1.pdf). (13.7.2012.).

Matija Varga

## BERNSTEIN ALGORITHM FOR VERTICAL NORMALIZATION TO 3NF USING SYNTHESIS

### ABSTRACT

This paper demonstrates the use of Bernstein algorithm for vertical normalization to 3NF using synthesis. The aim of the paper is to provide an algorithm for database normalization and present a set of steps which minimize redundancy in order to increase the database management efficiency, and specify tests and algorithms for testing and proving the reversibility (i.e., proving that the normalization did not cause loss of information). Using Bernstein algorithm steps, the paper gives examples of vertical normalization to 3NF through synthesis and proposes a test and an algorithm to demonstrate decomposition reversibility. This paper also sets out to explain that the reasons for generating normal forms are to facilitate data search, eliminate data redundancy as well as delete, insert and update anomalies and explain how anomalies develop using examples.

**Keywords:** functional dependency, relation schema, normalization, normalization using synthesis, decomposition, Bernstein algorithm, loss of information.