

# NORMALIZACIJA BAZE PODATKOV S POMOČJO TEORIJE GRAFOV

Jože Nemeč, Univerza v Mariboru, Visoka kmetijska sola, Vrbanska 30, 62000 Maribor  
Janez Grad, Univerza v Ljubljani, Ekonomska fakulteta, Kardeljeva ploščad 17, 61000 Ljubljana

## POVZETEK

Določitev višjih normalnih oblik baz podatkov je pogosto zamuden proces. Delo pri normalizaciji baz si lahko olajšamo, če uporabimo teorijo grafov. V članku so navedene lastnosti grafov baz podatkov in lastnosti matrik, ki te grafe opisujejo. Na osnovi teh lastnosti je prikazan algoritem za določitev baze podatkov brez odvečnih povezav.

## ABSTRACT

*DATABASE NORMALIZATION PROCESS BASED ON THE GRAPH THEORY - Determination of the higher order normal forms for database is frequently a time-consuming process. We can make the work easy by applying graph theory. In the paper the necessary characteristics of database graphs and the corresponding matrices which describe the graphs are analyzed. The algorithm for designing database involving no redundancy, based on these characteristics is also presented.*



Ključna gesla: relacijske baze, normalizacija, tretja normalna oblika

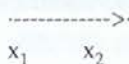
## UVOD

Proces izgradnje relacijskih baz poteka praviloma v več fazah. Tako moramo opraviti snemanje stanja v organizaciji, zatem moramo določiti povezave med podatki in na koncu te povezave še normalizirati. Iskanje normaliziranih oblik relacijskih baz je zahteven postopek. Idealno je, če lahko ta postopek v čim večji meri avtomatiziramo tako, da lahko pri izgradnji ustreznega modela uporabimo tudi računalnik. V preteklosti so bili že izdelani taki postopki. V (Vetter, Maddison, 1981) je prikazana izgradnja baze, kjer na osnovi ugotavljanja odvečnih povezav pridemo do ustreznega modela. Drugi algoritem je prikazan v (Salzberg, 1986), popravek tega algoritma je naveden v (Atkins, 1988) in (Diederich, 1991). Nekoliko modificirano obliko tega algoritma pa najdemo v (Diederich, Milton, 1988).

V pričujočem prispevku bomo skušali prikazati metodo, ki nam omogoča določitev optimalne baze podatkov s pomočjo teorije grafov.

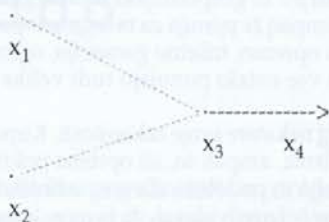
## GRAFI IN RELACIJE V BAZI PODATKOV

V informacijskem sistemu imamo praviloma opravka s podatki vrste entitet, ki so predstavljene s svojimi atributi. Denimo, da prikažemo te attribute kot vozlišča grafa. Povezave med posameznimi atributi pa prikažemo z usmerjenimi vejami. Povezava prikazana na sliki 1 prikazuje povezavo med dvema atributoma.



Slika 1: Medsebojno povezana atributa  $x_1$  in  $x_2$ .

Atribut  $x_1$  na sliki 1 je ključ relacije. Tako je v relaciji BARVA-IZ (IZDELEK, BARVA) ključ izdelek, barva pa je odvisen atribut. V primeru, da sestavlja ključ relacije več atributov, govorimo o sestavljenem ključu (vozlišču). Primer takega sestavljenega ključa je relacija MNENJE-ŠT (ŠTUDENT, UČITELJ, MNENJE). Ključ te relacije sestavljata atributa ŠTUDENT in UČITELJ, saj mnenje ni odvisno le od atributa ŠTUDENT ali le od atributa UČITELJ. Zato take relacije ne moremo razstaviti na več elementarnih relacij. Med atributi, ki sestavljajo ključ, in sestavljenim ključem vzpostavimo trivialno vejo, ki ima začetek v elementarnih atributih in konec v sestavljenem ključu. Trivialne veje bomo v grafu prikazali s črtasto črto. Gornjo relacijo lahko zato prikažemo s sliko 2.



Slika 2: Sestavljeni ključ relacije.

Na gornji sliki sestavljata atributa ŠTUDENT ( $x_1$ ) in UČITELJ ( $x_2$ ) sestavljeni ključ relacije  $x_3$ . Oba atributa, ki sestavljata sestavljeni ključ, sta z njim povezana s trivialnima vejama.

Denimo, da smo pri izgradnji baze podatkov dobili sliko o podatkih v informacijskem sistemu. Ker je ta slika nastala na osnovi potreb različnih služb, imamo opravka z

množico atributov in različnih povezav med atributi. Na osnovi teh podatkov lahko narišemo graf z množico vozlišč (atributov) in usmerjenih vej (relacij) med njimi. Privzemimo, da so v grafu vse netrivialne in trivialne veje naše baze. V tem primeru se da dokazati, da mora graf relacijske baze podatkov zadoščati pogojema:

- Nobena krožna pot v grafu ne sme vključevati vozlišča, ki predstavlja sestavljeni ključ neke relacije.
- Če obstaja pot z začetkom v sestavljenem vozlišču (ključu) in koncem v vozlišču  $x_i$ , tedaj ne sme obstajati pot, ki ima začetek v vozlišču, ki sestavlja sestavljeno vozlišče in konec v vozlišču  $x_i$ .

Dokaz za ti trditvi in za vse nadaljnje trditve v tem poglavju bo lahko bralec našel v članku istih avtorjev, ki bo izšel v reviji Informatica. V primeru pa, da je relacijska baza podatkov v tretji normalni obliki, mora zadoščati še naslednji zahtevi:

- Med dvema vozliščema sme obstajati samo ena pot.

Upoštevajmo sedaj še pojem povezanosti grafa. Usmerjeni graf je

- a) krepko povezan, če za poljubni vozlišči  $x_i$  in  $x_j$ , ki nista istovetni, velja, da obstaja pot, ki vodi iz vozlišča  $x_i$  v  $x_j$ , in pot, ki vodi iz vozlišča  $x_j$  v  $x_i$ ;
- b) enostransko povezan, če za poljubni vozlišči  $x_i$  in  $x_j$ , ki nista istovetni, velja, da obstaja pot, ki ima začetek v enem izmed obeh vozlišč in konec v drugem, lahko pa sta vozlišči povezani s potmi v obe smeri;
- c) šibko povezan, če med poljubnima vozliščema, ki nista istovetni, obstaja zaporedje poljubno usmerjenih vej;
- d) nepovezan, če ni šibko povezan.

Za graf baze podatkov v tretji normalni obliki velja:

- Graf baze podatkov v tretji normalni obliki je šibko povezan graf.

Oglejmo si sedaj še lastnosti podgrafov baze podatkov. Denimo, da imamo tak podgraf, da so v njem vsa vozlišča krepko povezana. Tak podgraf bomo poslej imenovali krepka komponenta danega grafa. Seveda ima lahko nek graf več krepkih komponent. Dokazati se da, da velja trditve:

- Graf baze podatkov v tretji normalni obliki ne sme imeti krepkih komponent.

Definirajmo sedaj matriko povezav grafa:

Vsakemu grafu baze podatkov lahko priredimo kvadratno matriko povezav vozlišč  $P = |p_{ij}|$  v kateri je

- a)  $p_{ij} = p$ , če obstaja  $p$  vej, ki imajo začetek v  $i$ -tem in konec v  $j$ -tem vozlišču in
- b)  $p_{ij} = 0$ , če ne obstaja nobena veja z začetkom v  $i$ -tem in koncem v  $j$ -tem vozlišču.

Oglejmo si lastnosti matrike  $P$ . Izračunajmo najprej višje potence matrike  $P$ . Dokazati se da, da ima ta matrika naslednje lastnosti:

- Element  $r_{ij}$   $k$ -te potence matrike povezav  $R=P^k$  je enak številu poti z dolžino  $k$ , ki vodijo od  $i$ -tega k  $j$ -temu vozlišču.
- Če je element  $p_{ij}$  matrike  $P$  baze podatkov v tretji normalni obliki enak 1, je element  $p_{ji}$  te matrike enak 0. Diagonalni elementi  $p_{ii}$  matrike  $P$  so enaki nič.
- V grafu baze podatkov, ki je v tretji normalni obliki, lahko element  $r_{ij}$   $k$ -te potence matrike povezav  $R=P^k$  zavzame le vrednosti 0 ali 1. Če je element  $r_{ij}$  matrike  $P^k$  enak 1, so vsi elementi  $a_{ij}$  matrik  $P, P^2, P^3, \dots, P^{k-1}$  enaki 0.

Tvorimo sedaj matriko  $D^k$  za katero velja, da je:

- a)  $d_{ij}^k = 0$ , če so vsi elementi  $a_{ij}$  matrik  $P, P^2, \dots, P^k$  enaki nič in
- b)  $d_{ij}^k = 1$ , če je vsaj en element  $a_{ij}$  matrik  $P, P^2, \dots, P^k$  od nič različen.

Dokazati se da, da veljajo trditve:

- Vedno obstoji tako število  $m$ , da veljajo enakosti:
- $D = D^m = D^{m+k}$ ,  $k = 1, 2, 3, \dots$
- Matrika  $D$  določa tranzitivno zaprtje baze podatkov. V  $i$ -ti vrstici so od nič različni le elementi  $d_{ij}$ , ki določajo, da obstoji pot z začetkom v  $i$ -tem vozlišču in koncem v  $j$ -tem vozlišču.
- Za elemente matrike  $D$  baze podatkov v tretji normalni obliki velja, da je  $d_{ij}=0$ , če je  $d_{ji}=1$ .

V primeru, da baza podatkov vsebuje krožne poti, lahko matriko  $D$  razstavimo na vsoto ene simetrične in ene asimetrične matrike tako, da je

$$D = D^s + D^a$$

Elementi  $d_{ij}^s$  in  $d_{ij}^a$  teh matrik so določeni takole:

$$d_{ij}^s = 1 \text{ in } d_{ji}^a = 0, \text{ če je } d_{ij} = d_{ji} = 1$$

$$d_{ij}^s = 0 \text{ in } d_{ji}^a = d_{ij}, \text{ če je } d_{ij} \neq d_{ji}$$

S to razstavitvijo matrike  $D$  smo dosegli, da je element  $d_{ij}^s$  matrike  $D^s$  enak 1 le, če obstaja pot, ki gre iz  $i$ -tega v  $j$ -to vozlišče in pot, ki gre iz  $j$ -tega v  $i$ -to vozlišče. Element  $d_{ij}^a$  matrike  $D^a$  pa je enak 1 le, če obstaja pot, ki vodi iz  $i$ -tega v  $j$ -to vozlišče in ne obstaja pot, ki vodi iz  $j$ -tega v  $i$ -to vozlišče.

## POSTOPEK DOLOČITVE OPTIMALNE ZGRADBE BAZE PODATKOV

Skušajmo sedaj vse te lastnosti, ki jih graf baze podatkov mora imeti, izkoristiti za to, da določimo optimalno zgradbo baze podatkov. To bomo storili tako, da bomo s pomočjo matrik  $D, D^s, P, P^2, P^3, \dots, P^k$  določili potrebne relacije v relacijski bazi podatkov.

Denimo, da smo za dane podatke določili matriko povezav  $P$ . V tej matriki naj bodo tako netrivialne kot trivialne veje. S pomočjo te matrike lahko izračunamo zaporedne po-

tence  $P^2, P^3, \dots, P^k$  matrike  $P$ . Istočasno lahko tvorimo matrike  $D^1, D^2, \dots, D^k$ . Računanje višjih potenc matrike  $P$  končamo, ko sta matriki  $D^k$  in  $D^{k+1}$  enaki. V tem primeru je  $D = D^k$ .

Glede na obliko teh matrik moramo določiti tudi nadaljnje postopke. Za začetek si oglejmo primer, da lahko pišemo matriko  $D$  kot vsoto simetrične in asimetrične matrike. Denimo, da smo ta razcep lahko opravili. V tem primeru je:

$$D = D^s + D^a$$

Denimo, da je v gornji simetrični matriki element  $d_{ij}$  od nič različen. Oglejmo si sedaj  $i$ -to vrstico te matrike. Razen elementa v  $j$ -tem stolpcu so lahko elementi z vrednostjo 1 še v nekaj stolpcih. Vsa vozlišča, ki ustrezajo tem stolpcem, so povezana v obeh smereh. To pa pomeni, da predstavljajo krepko povezan graf. Tak graf pa ni v tretji normalni obliki.

Tvorimo sedaj podgraf iz vozlišč, ki ga sestavljajo vsa vozlišča (atributi), katerih elementi so v  $i$ -ti vrstici matrike  $D^s$  od nič različni. V matriki  $D$  takega podgrafa so vsi elementi enaki 1. To pomeni, da ima baza podatkov, ki jo predstavlja tak graf, krožne poti, ki jih moramo odstraniti. To storimo tako, da gledamo podgraf, ki ga sestavljajo samo krepko povezana vozlišča. Z odvzemom ene veje prekinemo vsaj eno krožno pot. V primeru, da smo posameznim relacijam določili pogostnost uporabe, lahko to črtanje avtomatiziramo tako, da postopoma črtamo veje z najmanjšo pogostnostjo uporabe. V primeru, da pogostnosti uporabe relacij nismo določili, se moramo odločiti sami, katero vejo moramo v dani fazi postopka črtati.

Postopek črtanja vej v krožnih poteh pričnemo tako, da poiščemo tista vozlišča, ki so v matriki  $P$  obojestransko povezana. Za ta vozlišča velja, da je

$$P_{ij} = P_{ji}$$

To pomeni, da obstaja tako veja, ki vodi iz  $i$ -tega vozlišča k  $j$ -temu vozlišču, kot veja, ki vodi od  $j$ -tega vozlišča k  $i$ -temu vozlišču. Med tema vejama izberemo tisto, ki ima manjšo pogostnost uporabe, in jo črtamo. Ko smo na ta način prekinili vse veje, ki sestavljajo krožne poti z dvema vozliščema, nadaljujemo še s črtanjem ostalih odvečnih vej.

Po odvzemu vsake posamezne veje analiziramo matriko povezav danega podgrafa na enak način kot analiziramo matriko celotnega grafa. Postopek odzemanja vej ponavljamo tako dolgo, dokler ne odstranimo vseh možnih krožnih poti. Matrika  $D^s$  nam torej omogoča določiti tisti del modela baze podatkov, ki ima odvečne krožne povezave.

Posebej pa moramo opozoriti, da v nobenem podgrafu, ki vsebuje krožne poti, ne sme biti sestavljenih vozlišč. Če v nekem podgrafu s krožnimi potmi obstaja kakšno sestavljeno vozlišče, moramo zgradbo sestavljenega ključa ponovno preveriti.

Ko smo dosegli, da so prekinjene vse krožne poti, bo matrika  $D$  asimetrična. Sedaj moramo preveriti še, ali obstajajo kakšne tranzitivne povezave v našem grafu. Če take povezave obstajajo, graf ni v tretji normalni obliki. Eksisten-

co tranzitivnih vej ugotovimo tako, da primerjamo matriko  $P^k$  z matriko  $D^{k-1}$ . Če je v obeh matrikah element  $a_{ij}$  enak 1, tedaj med  $i$ -tim in  $j$ -tim vozliščem obstajata dve poti. Prav tako so v grafu tranzitivne veje, če je element  $a_{ij}$  matrike  $P^k$  večji od nič. Z določitvijo tranzitivnih poti (postopek določevanja poti bomo opisali pozneje) lahko na eni izmed poti odstranimo končno vejo. Tudi ta postopek lahko avtomatiziramo, če smo določili, katero vejo (relacijo) na danih poteh bomo najmanjkrat uporabljali. Vendar moramo omeniti, da v tranzitivnih poteh ne sme biti trivialnih vej. Če so v kakšni tranzitivni poti tudi trivialne veje, tedaj sestavljena vozlišča niso korektno določena. V tem primeru moramo korektnost sestavljenih vozlišč preveriti. Ko pri določitvi matrik  $D$  ne najdemo več tranzitivnih poti, je postopek določitve povezav v optimalni bazi podatkov končan.

### Naslednji korak je določitev zgradbe baze na osnovi danih povezav.

Najprej črtamo v matriki  $P$  vse trivialne veje. Zatem pa tvorimo matrike  $P, P^2, P^3, \dots, P^k$  in  $D$ . Kot smo že ugotovili, mora biti matrika  $D$  baze podatkov v tretji normalni obliki asimetrična. Oglejmo si, kakšen pomen ima ta matrika. Denimo, da so vsi elementi  $j$ -te vrstice te matrike enaki 0. Tedaj je to vozlišče (atribut) lahko le končno vozlišče vej ali poti v grafu. Iz tega vozlišča ne moremo doseči nobenega drugega vozlišča. Analizirajmo še primer, ko obstajajo v  $i$ -ti vrstici in stolpcu od nič različni elementi. Tedaj so v grafu veje, ki imajo konec v  $i$ -tem vozlišču in veje, ki imajo začetek v  $i$ -tem vozlišču.

Podobno lahko ugotovimo, da če so vsi elementi  $i$ -tega stolpca enaki nič, to vozlišče (atribut) ni končno vozlišče nobene veje. Iz takega vozlišča lahko dosežemo nekatera druga vozlišča. Denimo, da smo določili vsa vozlišča, ki imajo v pripadajočih stolpcih vse elemente enake nič. Vsa vozlišča s to lastnostjo določajo neko podmnožico vozlišč  $K$  našega grafa. Vzemimo neko vozlišče  $x_j$  grafa, ki ni element množice  $K$ . V tem primeru obstaja v množici  $K$  vsaj eno vozlišče  $x_i$ , ki je povezano z vozliščem  $x_j$  z neko vejo ali potjo. Vozlišča, ki niso v množici  $K$ , so namreč končna vozlišča vsaj ene veje. Začetno vozlišče veje s koncem v vozlišču  $x_j$  je ali vozlišče iz množice  $K$  ali pa je vozlišče, ki je istočasno začetno in končno vozlišče vej. V drugem primeru postopek iskanja predhodnih vej nadaljujemo tako dolgo, dokler ne pridemo do vozlišča iz množice  $K$ . V teoriji grafov imenujemo vozlišča iz množice  $K$  bazna vozlišča. Bazna vozlišča v grafu imajo to lastnost, da predstavljajo vhod v graf. Iz množice baznih vozlišč grafa lahko vedno dosežemo vsa druga vozlišča v grafu.

V bazi podatkov nam bazna vozlišča grafa predstavljajo primarne ključne relacij. Razen teh nastopajo v bazi še drugi ključni, ki jih moramo določiti s pomočjo matrike  $P$  in njenih potenc.

Oglejmo si zaporedje matrik  $P, P^2, P^3, \dots, P^k$  in njihovo povezavo z matriko  $D$ . Vzemimo, da smo s pomočjo matrike  $D$  določili končna vozlišča poti. Denimo, da je  $i$ -to vozlišče

(atribut) končno vozlišče neke poti. V tem primeru so vsi elementi v  $i$ -ti vrstici enaki nič. Tvorimo sedaj zaporedje, ki ga določajo elementi  $i$ -tega stolpca matrik  $P^k$ . Glede na število od nič različnih elementov v  $i$ -tem stolpcu matrike  $P^k$  ločimo tri možnosti:

- to število je enako 1,
- to število je večje od 1,
- to število je enako 0.

Oglejmo si vse tri primere. V prvem primeru obstaja v  $i$ -tem stolpcu matrike  $P$  natanko en element, katerega vrednost je enaka 1. Denimo, da je ta element v  $j_1$ -ti vrstici. Denimo, da je še v nekaj naslednjih zaporednih matrikah  $P^k$  po en od nič različen element v  $i$ -tem stolpcu. Na ta način dobimo zaporedje indeksov

$$j_1, j_2, j_3, \dots, j_n$$

To je zaporedje končnih vozlišč na poti od  $j_n$ -tega vozlišča k  $i$ -temu vozlišču. Vozlišča na tej poti si torej sledijo v naslednjem vrstnem redu:

$$j_n, j_{n-1}, j_{n-2}, \dots, j_1, i$$

Vsaka veja med dvema sosednjima vozliščema na tej poti definira eno relacijo. Vozlišče na začetku veje je ključ te relacije. Postopek končamo, ko je  $j_n$ -to vozlišče začetno vozlišče neke poti. V tem primeru so v vseh naslednjih matrikah  $P^k$  elementi v  $i$ -tem stolpcu enaki nič in imamo zato v tem primeru opravka s tretjim primerom.

Oglejmo si še drugo možnost. Denimo, da je v  $i$ -tem stolpcu matrike  $P^k$  en sam element od nič različen. Označimo vozlišče, ki ustreza temu elementu z  $x_k$ . Nadalje naj bo v  $i$ -tem stolpcu matrike  $P^{k+1}$   $m$  od nič različnih elementov.

Označimo sedaj vozlišča, ki ustrezajo tem elementom z:

$$x_{k1}, x_{k2}, x_{k3}, \dots, x_{km}$$

Vsako izmed navedenih vozlišč definira eno pot. Vse te poti potekajo od vozlišča  $x_k$  do vozlišča  $x_i$  po istih vejah. Do vozlišča  $x_k$  pa pridejo tako, da je vozlišče  $x_k$  končno vozlišče vej, ki imajo začetek v vozliščih  $x_{ki}$ .

Postopek nadaljujemo tako dolgo, dokler ne dosežemo vseh začetnih vozlišč poti. Ta začetna vozlišča poti so lahko le glavni ključ. Če smo odstranili vse trivialne veje, so začetna vozlišča poti tudi sestavljeni ključ.

Ob koncu tega postopka dobimo množico poti. Vsaka pot pa nam definira neko zaporedje vozlišč (atributov). Zapišimo vozlišča v posameznih poteh. Na ta način dobimo naslednjo množico zaporedij vozlišč na posameznih poteh:

$$1. \text{ pot: } x_{1,1}, x_{1,2}, x_{1,3}, \dots, x_{1,m1}$$

$$2. \text{ pot: } x_{2,1}, x_{2,2}, x_{2,3}, \dots, x_{2,m2}$$

$$3. \text{ pot: } x_{3,1}, x_{3,2}, x_{3,3}, \dots, x_{3,m3}$$

.

.

.

$$k\text{-ta pot: } x_{k,1}, x_{k,2}, x_{k,3}, \dots, x_{k,mk}$$

Prvi člen vsakega zaporedja je začetno vozlišče (glavni ključ ali sestavljeni ključ). Zadnji člen vsakega zaporedja je nek atribut. Po dva sosednja člena v isti vrsti definirata eno relacijo v bazi. Ključ te relacije je tisto vozlišče (atribut), ki je bližje začetku vrstice. Ob tem pripomnimo, da se lahko posamezna vozlišča (atributi) v zgornjih zaporedjih pojavljajo večkrat. Vendar se vsako vozlišče (atribut) lahko v posamezni vrstici pojavi samo enkrat.

Sedaj lahko tvorimo končni model relacijske baze v naslednjih korakih:

- Vzemimo vsa različna vozlišča, ki se pojavijo kot prvi člen gornjih zaporedij.
- Vsakemu vozlišču pripišemo pripadajoča vozlišča (atribute), ki nastopajo kot drugi člen zaporedja.
- Zatem v vsakem zaporedju črtamo prvi člen. Prav tako črtamo vsa tista zaporedja, ki jim je po črtanju prvega člena ostal le še en sam člen.
- Ponavljamo gornje tri korake tako dolgo, dokler ne črtamo vsa vozlišča. Ob tem moramo paziti, da lahko kot ključ nastopa vsako vozlišče (atribut) samo enkrat.

Določanje relacij, ki nastopajo v bazi podatkov lahko na ta način v veliki meri poenostavimo, saj nam posamezne korake lahko opravi računalnik. Zlasti je to pomembno tedaj, ko ima naša baza podatkov veliko atributov in veliko povezav med njimi. Tak primer pa srečamo že v vsaki malo obsežnejši organizaciji.

V veliki organizaciji ima matrika povezav zelo velike dimenzije. Število operacij, ki jih moramo opraviti, da dobimo matriko  $D$ , je zato zelo veliko. Opišimo še postopke, ki nam to matriko in število operacij zmanjšajo. Za določitev, katera vozlišča smemo odstraniti, uporabimo naslednji postopek.

Če je vsota elementov matrike  $P$  v  $i$ -ti vrstici nič, v  $i$ -tem stolpcu pa ena, je to končno vozlišče zadnje veje na neke poti. Istočasno ta veja ni v nobeni drugi poti. Zato lahko  $i$ -to vrstico in stolpec črtamo. Na ta način dobimo novo matriko povezav  $P'$ .

Če je vsota elementov matrike  $P$  v  $j$ -ti vrstici ena, v  $j$ -tem stolpcu pa nič, je to začetno vozlišče prve veje na neke poti. Ta veja ni v nobeni drugi poti. Zato lahko  $i$ -to vrstico in stolpec črtamo. Na ta način dobimo novo matriko povezav  $P''$ .

Postopek ponavljamo tako dolgo, dokler ne pridemo do reducirane matrike  $P_r$ , v kateri ne moremo črtati nobenega vozlišča več. Podgraf, ki ustreza tej matriki, vsebuje sedaj vse možne veje, ki jih lahko odstranimo.

Z reducirano matriko  $P_r$  lahko sedaj opravimo postopek odzemanja vej, ki smo ga opisali na začetku. Ker ima matrika  $P_r$  znatno manjše dimenzije kot prvotna matrika  $P$ , je seveda število operacij temu primerno manjše.

Denimo, da smo sedaj določili, katere veje moramo odstraniti in tako dobili novo reducirano matriko povezav  $P$ , ki teh vej ne vsebuje. Kot smo že ugotovili, potrebujemo za določitev posameznih poti višje potence te matrike. V matriki  $P$  pa je veliko takih vrstic in stolpcev, ki imajo vse člene enake nič. V teh primerih so tudi vse istoležne vrstice

in stolpci v matrikah višjih potenc enaki nič. Če to pri izračunu upoštevamo, se nam število operacij pri množenju matrik bistveno zmanjša.

Prav tako lahko zmanjšamo število matrik, ki jih moramo med obdelavo hraniti v pomnilniku. Ugotovili smo namreč, da sme biti element  $a_{ij}$  različen od nič le v eni potenci  $P^k$  matrike  $P$ . Denimo, da smo v matriki  $P$  odstranili tudi vse trivialne veje. Če to upoštevamo, lahko tvorimo matriko  $Z$ , katere elemente določimo tako, da izenačimo vse njene izven diagonalne elemente z nič, diagonalne elemente pa z 1. Ob izračunu matrike  $P^k$  pa postavimo, da je element  $z_{ij}$  matrike  $Z$  enak  $k$ , če je element  $P^k$  enak 1. Na ta način dobimo v stolpcih zaporedja števil. V stolpcih matrike  $Z$ , ki ustrezajo posameznim končnim vozliščem poti, števila od 1 do  $m$  določajo obratni vrstni red indeksov končnih vozlišč na poti od začetnega vozlišča  $k$  izbranemu končnemu vozlišču. Namesto vseh potenc matrike  $P$  lahko hranimo le matriko  $Z$ .

## ZAKLJUČEK

V pričujočem članku smo pokazali, da morajo imeti grafi, ki predstavljajo bazo podatkov v tretji normalni obliki, določene lastnosti. Prav tako morajo tudi matrike, ki te grafe opisujejo, ustrezati nekaterim zahtevam. Lastnosti matrik, ki prikazujejo bazo podatkov, pa lahko uspešno uporabimo pri določitvi algoritma, s katerim lahko konstruiramo bazo podatkov v tretji normalni obliki.

Z uvedbo kriterijev, ki nam določijo, katere veje smemo v danem trenutku črtati, lahko postopek določevanja tretje normalne oblike relacijske baze podatkov avtomatiziramo. S tem pa nam odpade zamudno nealgoritmčno načrtovanje ustreznega konceptualnega modela baze.

## LITERATURA:

- (1) Atkins John, A Note on Minimal Covers, SIGMOD Record, Vol. 17, št. 4, december 1988, str 16-21
- (2) Diederich Jim, Minimal Cover Revisited: Correct and Efficient Algorithms, SIGMOD Record, Vol. 20, št 1, marec 1991, str 12-13
- (3) Diederich Jim, Milton Jack, New Methods and Fast Algorithms for Database Normalization, ACM Transactions on Database Systems, Vol. 13, št. 3, september 1988, str 339-365
- (4) Nemeč Jože, Grad Janez, Graphs and the Third Normal Form for Relational Database, članek je bil poslan v objavo reviji Informatica
- (5) Salzberg Betty, Third Normal Form Made Easy, Sigmod Record, Vol. 15, št. 4, december 1986, str 2-18
- (6) Stout F. Quentin, Woodworth A. Patricia, Relational Databases, The American Mathematical Monthly, Vol. 90, 1983, str. 101-118
- (7) Vetter M., Maddison R.N., Database Design Methodology, Prentice Hall, Englewood Cliffs, 1981

## Navodila avtorjem

Prispevke pošiljajte v predpisani obliki na naslov Slovensko društvo Informatika, 61000 Ljubljana, Vožarski pot 12, s pripisom za revijo Uporabna informatika. Če je možno, naj bo članek lektoriran. V uredništvu bomo opravili korekturo in se po presoji posvetovali z avtorjem, da članek tudi lektoriramo. Prispevek naj bo v obsegu največ avtorska pola (30.000 znakov) za strokovne članke in približno 2 do 3 tiskane strani za druge prispevke. Vsak strokovni članek naj ima na začetku povzetek v slovenskem in v angleškem jeziku. Pošljite ga na disketi in odtisnjene na papirju. Napisan je lahko v kateremkoli urejevalniku besedil, vendar naj bo na disketi tudi kopija v ASCII formatu. Na disketi označite, kateri urejevalnik ste uporabili, in ime datoteke. Datoteke imenujte s svojim priimkom, n. pr. Novak.doc ali Novak.txt. Slike, ki ste jih izdelali z grafičnim programom, označite podobno. Na natisnjem izvodu članka naj bo jasno vidno, kam sodi posamezna slika. Lahko priložite tudi originalne predloge, ki jih na hrbtni strani označite s številkami, tako kot v natisnjem besedilu. Pišite v razmaku vrstic 1, brez posebnih ali poudarjenih črk ali podčrtovanja, za ločilom na koncu stavka napravite samo en prazen prostor, ne uporabljajte zamika pri odstavkih.

Za vsa vprašanja se obračajte na tehnično urednico Katarino Puc, 61000 Ljubljana, Ulica Gubčeve brigade 120, tel. 1271-579