

Operacije nad relacijama mogu da se grupišu u:

- **Konvencionalne skupovne operacije**, unija, presek, razlika i Dekartov proizvod. Relacija  $n$ -tog stepena se definiše kao podskup Dekartovog proizvoda  $n$  skupova (domena relacije) i predstavlja skup  $n$ -torki. Zbog toga su osnovne skupovne operacije i operacije relacione algebre;
- **Specijalne relacione operacije** selekcija, projekcija, spajanje i deljenje;
- **Dodatne operacije relacione algebre**, operacije koje su se kasnije dodavale originalnoj relacionoj algebri da bi se povećala njena "moć" kao upitnog jezika;
- **Operacije ažuriranja baze**;
- **Operacije u prisustvu nula vrednosti**;

Nisu sve originalne i dodatne operacije primitivne. Na primer, operacije spajanja i deljenje se mogu izvesti iz ostalih operacija, kao što će se pokazati kasnije. Isto tako, skup relacionih operacija se može proširiti i drugačije interpretirati u prisustvu nula vrednosti, što će takođe biti diskutovano u posebnom delu ovog poglavlja.

Skup relacionih operacija treba da bude **zatvoren**: rezultat koji se dobija njihovom primenom je takođe relacija (tabela) i može biti ponovo argument operacija relacione algebre. Ova osobina je od posebnog značaja za razvoj jezika zasnovanih na relacionoj algebri. Radi lakšeg prikazivanja primera i za definiciju operacija ažuriranja baze podataka uvešćemo i operaciju dodele  $A:=B$  kojom se vrednost promenljive  $B$  koja predstavlja relaciju dodeljuje promenljivoj  $A$  koja je takođe relacija.

### 3.2.1.1. Konvencionalne skupovne operacije u relacionoj algebri

Relacija  $n$ -tog stepena se definiše kao podskup Dekartovog proizvoda  $n$  skupova (domena relacije) i predstavlja skup  $n$ -torki. Zbog toga su osnovne skupovne operacije i operacije relacione algebre. Međutim, očigledno je da se operacije unije, preseka i diferencije ne mogu primeniti na sve relacije, odnosno nemaju uvek semantički značaj. Na primer, unija relacija Student i Predmet formalno treba da da relaciju čije su  $n$ -torke elementi bilo relacije Student bilo relacije Predmet, a tako dobijen rezultat nije relacija u smislu u kome su relacije ovde definisane, odnosno ovakva operacija unije ne bi bila "zatvorena". Zbog toga se definiše sledeći **uslov kompatibilnosti** relacija  $R_1$  i  $R_2$  za izvođenje operacija unije, preseka i diferencije:

**Relacije  $R_1$  i  $R_2$  moraju imati isti broj atributa (isti stepen), a njihovi odgovarajući atributi moraju biti definisani nad istim domenima da bi se nad njima mogle izvršiti operacije unije, preseka i diferencije.**

U literaturi se ovakve relacije nazivaju relacije kompatibilne za uniju (union compatible relations) ili, prosto, kompatibilne relacije. Koristi se takođe i termin "relacije istog tipa".

Primeri operacija unije, preseka i diferencije prikazaće se na relacijama:

$S_1$

Brlnd	MLB	Ime	Starost	ŠifSmer
152/97	16309723331981	Ana	19	01
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02

$S_2$

Brlnd	MLB	Ime	Starost	ŠifSmer
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02
003/94	23456786418976	Miloš	22	01

1. **Unija.** Date su relacije  $R_1$  i  $R_2$  koje zadovoljavaju navedeni uslov kompatibilnosti. Rezultat operacije unije

$$R_3 := R_1 \cup R_2$$

je relacija  $R_3$  koja sadrži sve n-torke koje se pojavljuju bilo u  $R_1$  bilo u  $R_2$ .

Primer:  $S_3 := S_1 \cup S_2$

$S_3$

Brlnd	MLB	Ime	Starost	ŠifSmer
152/97	16309723331981	Ana	19	01
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02
003/94	23456786418976	Miloš	22	01

2. **Diferencija.** Date su relacije  $R_1$  i  $R_2$  koje zadovoljavaju navedeni uslov kompatibilnosti. Rezultat operacije diferencije

$$R_3 := R_1 - R_2$$

su n-torke relacije  $R_1$  koje nisu istovremeno i n-torke relacije  $R_2$ .

Primer:  $S_4 := S_1 - S_2$

$S_4$

Brlnd	MLB	Ime	Starost	ŠifSmer
152/97	16309723331981	Ana	19	01

3. **Presek.** Date su relacije  $R_1$  i  $R_2$  koje zadovoljavaju navedeni uslov kompatibilnosti. Rezultat operacije preseka

$$R_3 := R_1 \cap R_2$$

je relacija  $R_3$  koja sadrži n-torke koje se pojavljuju u obe relacije  $R_1$  i  $R_2$ .

Primer:  $S_5 := S_1 \cap S_2$

$S_5$				
Brlnd	MLB	Ime	Starost	ŠifSmer
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02

4. **Dekartov proizvod.** Dekartov proizvod se može primeniti na bilo koje dve relacije. Rezultat ove operacije

$$R_3 := R_1 \times R_2$$

je relacija  $R_3$  čije su n-torke svi "parovi" koje čine jedna n-torka relacije  $R_1$  i jedna n-torka relacije  $R_2$ .

Dekartov proizvod ćemo prikazati na primeru datom na slici 3.2. Dekartov proizvod se često naziva i "nekontrolisano spajanje" jer, očigledno, može da posluži za spajanje tabela. "Nekontrolisano" je zbog toga što se u rezultatu javljaju sve kombinacije n-torki polaznih relacija, za razliku od operacije spajanja (o kojoj će se kasnije govoriti) koja u rezultatu daje samo kombinacije n-torki koje zadovoljavaju neki uslov.

$R_1$			$R_2$	
C	D	E	A	B
1	2	3	a1	b1
2	4	6	a2	b2
			a3	b3

  

$R_3 = R_1 \times R_2$				
C	D	E	A	B
1	2	3	a1	b1
1	2	3	a2	b2
1	2	3	a3	b3
2	4	6	a1	b1
2	4	6	a2	b2
2	4	6	a3	b3

Slika 3.2. Dekartov proizvod

Operacije unije, preseka i Dekartovog proizvoda su komutativne i asocijativne, što ne važi za operaciju razlike.

### 3.2.1.2. Specijalne relacione operacije

Pored skupovnih operacija u relacionom modelu se definišu i sledeće specifične operacije: projekcija, selekcija, spajanje i deljenje.

5. **Projekcija.** Operacija projekcije je unarna operacija koja iz neke relacije selektuje skup navedenih atributa, odnosno "vadi vertikalni podskup" iz odgovarajuće tabele. Formalno se može definisati na sledeći način:

Neka je  $R(A_1, A_2, \dots, A_n)$  relacija, a  $X$  podskup njenih atributa. Označimo sa  $Y$  komplement  $\{A_1, A_2, \dots, A_n\} - X$ . Rezultat operacije projekcije relacije  $R$  po atributima  $X$  je

$$\pi_X(R) = \{x \mid \exists y, \langle x, y \rangle \in R\}.$$

Primer: Posmatrajmo tabelu Građanin i njenu projekciju  $Pr_1$ . Kada se iz neke relacije selektuje podskup atributa preko operacije projekcije u rezultatu bi se mogli pojaviti duplikati n-torki. Operacija projekcije podrazumeva da se ovi duplikati eliminišu, da bi se zadovoljila definicija relacije, kako je to u navedenom primeru i urađeno. Na Slici 3.3. data je relacija Građanin i rezultat operacije projekcije

$$Pr_1 := \pi_{\text{Ime, Starost}}(\text{Građanin})$$

Građanin

MLB	Ime	Starost	MestoRođ
16309723331981	Ana	19	Beograd
13975673331981	Mirko	21	Valjevo
11145276418976	Zoran	20	Beograd
23243723331981	Ana	19	Niš
2222223331981	Mirko	21	Beograd
11145276418976	Zoran	20	Novi Sad
23456786418976	Miloš	22	Beograd

 $Pr_1$ 

Ime	Starost
Ana	19
Mirko	21
Zoran	20
Miloš	22

Slika 3.3. Operacija projekcije

6. **Selekcija (Restrikcija).** Selekcija je takođe unarna operacija koja iz date relacije selektuje n-torke koje zadovoljavaju zadati uslov ("vadi horizontalni podskup" tabele). Formalno se definiše na sledeći način: Data je relacija  $R(A_1, A_2, \dots, A_n)$  i predikat  $\Theta$  definisan nad njenim atributima. Rezultat operacije selekcije

$$\sigma_{\Theta}(R) = \{x \mid x \in R \text{ AND } \Theta(x)\}$$

je skup n-torki  $x$  relacije  $R$  koje zadovoljavaju predikat (uslov)  $\Theta$

Primer: Iz relacije Građanin (Slika 3.3.) prikaži građane koji su stariji od 20 godina i rođeni su u Beogradu.

$$Pr_2 := \sigma_{\text{Starost} > 20 \text{ AND MestoRođ} = \text{"Beograd"}}(\text{Građanin})$$

 $Pr_2$ 

MLB	Ime	Starost	MestoRođ
2222223331981	Mirko	21	Beograd
23456786418976	Miloš	22	Beograd

Originalno se ova operacija nazivala **restrikcija**, čime se želelo da kaže da uslov ograničava koje n-torke mogu da se jave u rezultatu. Sada se češće koristi naziv **selekcija**.

7. **Spajanje (Join)**. Spajanje je binarna operacija koja spaja dve relacije na taj način da se u rezultatu pojavljuju oni parovi n-torki jedne i druge relacije koji zadovoljavaju uslov zadat nad njihovim atributima. Formalno se definiše na sledeći način:

Date su relacije  $R_1(A_1, A_2, \dots, A_n)$  i  $R_2(B_1, B_2, \dots, B_m)$  i predikat  $\Theta$  definisan nad njihovim atributima. Obeležimo sa  $X$  i  $Y$  skupove atributa relacija  $R_1$  i  $R_2$ , respektivno. Rezultat operacije spajanja ovih relacija (tzv. **teta spajanje**) je

$$R_1[x\Theta]R_2 = \{ \langle x, y \rangle \mid x \in R_1 \text{ AND } y \in R_2 \text{ AND } \Theta(x, y) \}.$$

Oznaka  $x\Theta$  za operaciju spajanja ukazuje na činjenicu, očiglednu iz definicije teta spajanja, da ova operacija nije primitivna operacija relacione algebre, već se može izvesti uzastopnom primenom operacije Dekartovog proizvoda ( $\times$ ) nad relacijama koje se spajaju i selekcije po predikatu  $\Theta$  nad tako dobijenom relacijom.

Ako je predikat  $\Theta$  definisan sa  $A_k = B_j$ , s tim da su i atributi  $A_k$  i  $B_j$  definisani nad istim domenima, tada se takvo spajanje naziva **ekvispajanje**. Primer ekvispajanja ćemo prikazati na relacijama Građanin (Slika 3.3.) i Student koju ćemo, delimično modifikovanu, ponovo dati.

Student

BrInd	MLB	Smer
152/97	16309723331981	01
223/95	13975673331981	01
021/94	11145276418976	02
003/94	23456786418976	01

$Pr_3 := \text{Građanin} [\text{Građanin.MLB} = \text{Student.MLB}] \text{ Student}$

 $Pr_3$ 

MLB	Ime	Starost	MestoRođ	MLB	BrInd	Smer
16309723331981	Ana	19	Beograd	16309723331981	152/87	01
13975673331981	Mirko	21	Vajjevo	13975673331981	223/95	01
11145276418976	Zoran	20	Beograd	11145276418976	021/94	02
23456786418976	Miloš	22	Beograd	23456786418976	003/94	01

Očigledno je da se u rezultatu ekvispajanja uvek pojavljuju dve iste kolone, u gornjem primeru dve iste kolone MLB. Ako se jedna od te dve kolone izbaci, takvo spajanje se naziva **prirodno spajanje**. Prirodno spajanje podrazumeva i da su atributi po kojima se relacije spajaju istoimeni. Oznaka za operaciju prirodnog spajanja je \*

$Pr_4 := \text{Građanin} * \text{Student}$

Pr<sub>4</sub>

MLB	Ime	Starost	MestoRođ	BrInd	Smer
16309723331981	Ana	19	Beograd	152/87	01
13975673331981	Mirko	21	Valjevo	223/95	01
11145276418976	Zoran	20	Beograd	021/94	02
23456786418976	Miloš	22	Beograd	003/94	01

Na Slici 3.4. prikazana su još dva primera spajanja relacija pod složenijim uslovima.

C	D	E
1	2	3
2	4	6

A	B
1	1
2	2
3	3

$$R_3 := R_1 [R_1.C > R_2.A] R_2$$

C	D	E	A	B
2	4	6	1	1

$$R_4 := R_1 [R_1.C < R_2.A \text{ AND } R_1.D > R_2.B] R_2$$

C	D	E	A	B
2	4	6	3	3

Slika 3.4. Primeri spajanja relacija

8. **Deljenje.** Deljenje je operacija pogodna za upite u kojima se javlja reč "svi" ("sve", "sva"). Formalno se definiše na sledeći način:

Neka su  $A(X, Y)$  i  $B(Z)$  relacije gde su  $X$ ,  $Y$  i  $Z$  skupovi atributa takvi da su  $Y$  i  $Z$  jednakobrojni, a odgovarajući domeni su im jednaki. Rezultat operacije deljenja

$$A[Y \div Z]B = R(X)$$

gde  $n$ -torka  $x$  uzima vrednosti iz  $A.X$ , a par  $\langle x, y \rangle$  postoji u  $A$  za sve vrednosti  $y$  koje se pojavljuju u  $B(Z)$ .

Primer (Slika 3.5.): Iz relacija Predmet i Prijava prikaži brojeve indeksa studenata koji su položili sve predmete.

Predmet		Prijava	
ŠifPred		Brnd	ŠifPred
P1		152/97	P1
P2		152/97	P2
P3		021/94	P1
		003/94	P3
		152/97	P3

Prijava [ Prijava.ŠifPred + Predmet.ŠifPred] Predmet

Brnd
152/97

Slika 3.5. Primer za operaciju deljenja

Operacija deljenja nije primitivna operacija relacione algebre, već se može izvesti pomoću drugih operacija na sledeći način:

$$A(X, Y) [Y \div Z]B(Z) = \pi_X A - \pi_X ((\pi_X A \times B) - A)$$

Objašnjenje:

- $\pi_X A$  daje sve n-torke koje mogu da učestvuju u rezultatu,
- $(\pi_X A \times B)$  daje relaciju u kojoj se za svaku vrednost z iz B pojavljuju parovi  $\langle x, z \rangle$  sa svim vrednostima x,
- $((\pi_X A \times B) - A)$  ne sadrži ni u jednom paru  $\langle x, z \rangle$  one vrednosti x za koje u relaciji A, kao vrednosti y, postoje sve vrednosti z.

Rezultat je, znači, relacija koja sadrži one n-torke x za koje postoje u paru  $\langle x, y \rangle$ , kao vrednosti y, sve vrednosti z.

Relaciona algebra je proceduralni jezik. Pomoću operacija relacione algebre sačinjava se procedura koja dovodi do odgovora na postavljeni upit, kao što je to pokazano u prethodnom primeru. I ako je proceduralni jezik, relaciona algebra je znatno moćnija od klasičnih programskih jezika koji su takođe proceduralni. Razlog za to je što je operand relacione algebre relacija (cela tabela), a operand operacija sa datotekama u klasičnim jezicima je rekord (vrsta tabele).

Može se pokazati da se bilo koja relacija izvodljiva iz skupa datih relacija (bilo koji upit) može dobiti procedurom od tri koraka:

- (1) Dekartov proizvod svih relacija koje se koriste.
- (2) Selekcija n-torki koje zadovoljavaju predikat selekcije.
- (3) Projekcija rezultata po atributima koji se prikazuju.

Ova procedura se može iskazati jednim opštim izrazom relacione algebre

$$\pi_{A_1, A_2, \dots, A_n} (\sigma_{\Theta} (R_1 \times R_2 \times \dots \times R_m)).$$

Međutim, ovakvo izvođenje operacija bilo bi veoma "skupo" jer bi rezultat Dekartovog proizvoda relacija  $R_1, R_2, \dots, R_m$  bio relacija sa  $k_1 \cdot k_2 \cdot \dots \cdot k_m$  n-torki, gde su sa  $k_i$  označene kardinalnosti odgovarajućih relacija. Zbog toga se i definiše

10. **Skalarna računanja u relacionoj algebri.** Skup osnovnih operacija relacione algebre je *relaciono kompletan*: bilo koja relacija (tabela) iz skupa datih relacija (tabela), odnosno relacione baze podataka, se može izvesti preko ovih operacija. Međutim, ove operacije ne obuhvataju nikakva skalarna računanja preko kojih bi se mogle formirati vrednosti atributa relacija. Zbog toga se uvode sledeća dva dodatna iskaza:

- **EXTEND**, koji uključuje tzv. "horizontalno računanje" u relacionu algebru, odnosno formira vrednost atributa relacije preko aritmetičkog izraza nad vrednostima ostalih atributa iste n-torke;
- **SUMMARIZE**, koji uključuje tzv. "vertikalno računanje" u relacionu algebru, odnosno formira novu relaciju kao projekciju polazne po nekim atributima, proširenu za atribut koji je rezultat nekog "vertikalnog", agregirajućeg, aritmetičkog izraza nad vrednostima odgovarajućih atributa.

**EXTEND.** Ovaj iskaz se formalno definiše na sledeći način:

EXTEND term ADD skalarni-izraz AS atribut

gde je

term :: bazna-relacija | izvedena-relacija | izraz-relacione-algebre

Neformalno, iskaz EXTEND proširuje datu relaciju (term) sa novim atributom koji je rezultat nekog sračunavanja nad postojećim atributima posmatrane n-torke. Na primer, posmatrajmo relaciju Proizvod(ŠP, NazP, Težina)

**Proizvod**

ŠP	NazP	Težina
001/2a	Zavrtanj	5
127/3b	Klin	2

Ako je težina data u funtama, može se relacija proširiti sa atributom TežKg koji daje težinu u kilogramima preko iskaza:

EXTEND Proizvod ADD (Težina \* 0.453) AS TežKg

**Proizvod**

ŠP	NazP	Težina	TežKg
001/2a	Zavrtanj	5	2.265
127/3b	Klin	2	0.506

Novi atribut, dobijen iskazom EXTEND se može koristiti u operacijama projekcije, selekcije i drugim operacijama relacione algebre. Na primer, moguće je napisati upit

$\sigma_{TežKg > 2.0}$  (EXTEND Proizvod ADD (Težina \* 0.453) AS TežKg )

**SUMMARIZE.** Ovaj iskaz se formalno definiše na sledeći način:

SUMMARIZE term BY (lista-atributa) ADD agregirajući-izraz AS atribut



Opšti primer ovog iskaza koji služi za njegovo objašnjenje je:

SUMMARIZE A BY ( $A_1, A_2, \dots, A_n$ ) ADD izraz AS Z

$A_1, A_2, \dots, A_n$  su atributi relacije A. Rezultat iskaza SUMMARIZE je relacija sa atributima  $A_1, A_2, \dots, A_n, Z$ . N-torke ove relacije se dobijaju kao n-torke projekcije relacije A po atributima  $A_1, A_2, \dots, A_n$  proširene za Z koje se sračunava po datom izrazu nad svim n-torkama A koje imaju iste vrednosti  $A_1, A_2, \dots, A_n$ .

Kao primer, posmatrajmo ponovo relaciju Prijava

Prijava

Brlnd	ŠifPred	Datum	Ocena
152/97	P1	18.3.97	9
152/97	P2	21.4.96	8
021/94	P1	?	10
003/94	P3	30.5.97	6
152/97	P3	1.7.97.	10

Relacija sa srednjim ocenama svih studenata može se dobiti iskazom

SUMMARIZE Prijava BY (Brlnd) ADD AVG(Ocena) AS ProsOc

Rezultat je

Brlnd	ProsOc
152/97	9.0
021/94	10.0
003/94	6.0

Agregacioni izraz AVG sračunava srednju vrednost ocena za jednog studenta. Moguće je definisati i druge agregacione izraze, odnosno funkcije: SUM (zbir vrednosti), MAX (maksimalna vrednost), MIN (minimalna vrednost), COUNT (broj n-torki) itd. Vertikalna sračunavanja se mogu interpretirati i na sledeći način: Grupišu se n-torke polazne relacije po istoj vrednosti nekog atributa (u prethodnom primeru Brlnd), na svaku grupu se primeni odgovarajuća agregaciona funkcija (u prethodnom primeru AVG), rezultat se prikazuje u posebnoj relaciji u kojoj svakoj grupi odgovara jedna n-torka.

11. **Operacije poređenja nad relacijama.** Relaciona algebra nije originalno uvela operacije poređenja nad relacijama: jednakost ili nejednakost dve relacije, da li je jedna relacija podskup (nadskup) ili pravi podskup (nadskup) druge. Očigledno je da ove operacije vraćaju logičke vrednosti TRUE ili FALSE. Ove operacije je moguće uvesti samim tim što su relacije skupovi. One mogu biti pogodne za neke upite koji bi se inače veoma složeno iskazivali. Preko njih se takođe jednostavno mogu iskazati neka pravila integriteta, o čemu će se kasnije detaljno govoriti.

### 3.2.1.4. Operacije održavanja baze podataka

Operacije održavanja (ažuriranja) baze podataka treba da omoguće dodavanje n-torke u neku relaciju, izbacivanje n-torke iz neke relacije ili promenu vrednosti atributa relacije. Drugim rečima, operacije održavanja treba da obezbede promenu i pamćenje stanja u bazi podataka. Očigledno je da je za potrebe održavanja baze podataka neophodno definisati i operaciju dodele relacija.

Ako je definisana operacija dodele, tada se dodavanje n-torki u bazu podataka može izvršiti preko operacije unije, na primer na sledeći način:

Student := Student  $\cup$  (<BrInd:003/94>, <MLB:23456786418976>, <Ime:Miloš>, <Starost:22>, <ŠifSmer:01>)

Relacija Student pre ove operacije je

Student				
BrInd	MLB	Ime	Starost	ŠifSmer
152/97	16309723331981	Ana	19	01
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02

a posle nje

Student				
BrInd	MLB	Ime	Starost	ŠifSmer
152/97	16309723331981	Ana	19	01
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02
003/94	23456786418976	Miloš	22	01

Moguće je da se umesto direktnog navođenja n-torki koje se ubacuju, formira izraz koji daje relaciju kompatibilnu sa onom koja se ažurira. Na primer, ako je data relacija

$S_2$				
BrInd	MLB	Ime	Starost	ŠifSmer
342/95	14795673331321	Mirko	25	01
021/94	11145276418976	Zoran	20	02
003/94	23456786418976	Miloš	22	01

tada se može izvršiti sledeće ažuriranje:

Student := Student  $\cup$  ( $\sigma_{\text{ŠifSmer} = 01}(S_2)$ )

Na sličan način, preko operacije diferencije, može se definisati operacija izbacivanja n-torki iz baze:

$$\text{Student} := \text{Student} - (\langle \text{BrInd}:125/97 \rangle, \langle \text{MLB}:16309723331981 \rangle, \langle \text{Ime}: \text{Ana} \rangle, \langle \text{Starost}:19 \rangle, \langle \text{ŠifSmer}:01 \rangle)$$

Isto tako moguće je napisati i

$$\text{Student} := \text{Student} - (\sigma_{\text{ŠifSmer} = 01}(\text{Student}))$$

koja iz relacije Student izbacuje sve n\_torke u kojima je ŠifSmer = 01.

Operacija ažuriranja vrednosti nekog atributa se obavlja na taj način što se prvo operacijom diferencije izbace n-torke čiji se atribut menja, a zatim vrate preko operacije unije sa izmenjenom vrednošću atributa. Na primer,

$$\text{Student} := \text{Student} - (\langle \text{BrInd}:125/97 \rangle, \langle \text{MLB}:16309723331981 \rangle, \langle \text{Ime}: \text{Ana} \rangle, \langle \text{Starost}:19 \rangle, \langle \text{ŠifSmer}:01 \rangle) \cup (\langle \text{BrInd}:125/97 \rangle, \langle \text{MLB}:16309723331981 \rangle, \langle \text{Ime}: \text{Ana} \rangle, \langle \text{Starost}:19 \rangle, \langle \text{ŠifSmer}:02 \rangle)$$

Student sa BrInd = 125/97 promenio je ŠifSmer sa 01 na 02.

Ovde je samo pokazano da se operacije unije i diferencije, zajedno sa operacijom dodele mogu da koriste kao operacije održavanja relacione baze podataka. Isto tako je moguće, što se češće i čini, definisati posebne operacije za ubacivanje i izbacivanje n-torki i promenu sadržaja nekog atributa relacije.

### 3.2.1.5. Operacije sa nula vrednostima

Navedene operacije relacione algebre nisu uzimale u obzir "nula vrednosti" koje mogu postojati u relacijama. Podrazumevalo se da se one u relacijama ne pojavljuju i da se vrednosti predikata sračunavaju na osnovu standardnih, dvovrednosnih tablica istinitosti. Pojavljivanje "nula vrednosti" u relacionoj bazi podataka zahteva da se proširi skup definisanih operacija, da bi se one na neki način uzele u obzir.

Očigledno je da u prisustvu nula vrednosti dvovrednosna logika u kojoj se neki predikat može da sračuna samo u T (istina) ili F (neistina) nije dovoljna. Na primer, za građane čija je starost nepoznata, predikat Građanin.Starost > 25, nema ni F ni T kao vrednost, već je vrednost tog predikata, za posmatranog građanina, nepoznata. Zbog toga je neophodno uvesti trovrednosnu logiku, sa vrednostima T, F, ?, gde ? označava nepoznatu vrednost, koja se često naziva i "možda" vrednost. (Građanin x je "možda" stariji od 25 godina je predikat koji uzima u obzir i "nula vrednost" atributa Starost). Pored toga treba jasno definisati kako se sračunavaju aritmetički izrazi gde neki od operandi dobija nula vrednost i šta nula vrednost predstavlja u nekom skupu.