

Strukturna komponenta relacionog modela

Atribut i domen

Atribut u relacionom modelu odgovara onome što nazivamo podatak.

definicija

Atribut je imenovana vrsta svojstava koja se ne može rastavljati na dijelove bez gubitka svakog značenja.

Navedena definicija podrazumjeva prost atribut. Atribut sastavljen od više atributa nazivamo složen atribut. Takav atribut možemo rastaviti na jednostavnije attribute koji mogu biti isto složeni ili prosti. Pri tome ne dolazi do potpunog gubitka značenja već samo do promjene značenja.

2.1. Struktura relacionog modela

Primjeri atributa

Datum je složen atribut, pošto se može rastaviti na tri prosta atributa: dan u mjesecu, mjesec i godinu. Tim rastavljanjem se gubi prvobitno značenje datuma kao cjeline, ali dobijeni prosti atributi imaju svaki svoje značenje.

Adresa je složen atribut, koji se može rastaviti na mjesnu adresu i mjesto, koji su takođe složeni atributi. Mjesnu adresu možemo rastaviti na naziv ulice i broj, koji su prosti. Mjesto čine dva prosta atributa, poštanski broj i naziv mjesta.

Svako svojstvo podrazumjeva pored vrste i mogućnost opisa pomoću vrijednosti tog svojstva. Na primjer, za vrstu svojstva “visina u cm” možemo imati vrijednosti 187, 190 itd. Za svojstvo opštine mogu da se pojave imena samo opština koje postoje u RS. Ovo nas dovodi do pojma **domena**, drugog osnovnog pojma relacionog modela.

Definicija

Skup svih mogućih vrijednosti nekog atributa A_i naziva se domenom tog atributa i označava se sa D_i ili $\text{dom}(A_i)$.

Pojam domena je vrlo blizak pojmu tipa podataka u programiranju. Svakom atributu odgovara samo jedan domen. Ali zato jednom domenu može da odgovara više atributa, odnosno 2 ili više atributa mogu imati zajednički domen.

Primjer

Atributi „Visina u cm“ i „trajanje filma u min“ imaju isti domen, a to je skup pozitivnih cijelih brojeva.

Definisanje šeme relacije podrazumjeva, da su samim tim što su zadati atributi zadati i njihovi domen.

Tri bitne osobine šeme relacije:

- atributi moraju biti unikatni, odnosno različitih naziva
- redoslijed atributa nije bitan, u smislu značenja
- šema relacije mora da sadrži bar jedan atribut.

Nebitnost redoslijeda atributa je samo teorijska. Kada se navodi šema relacije moraju se atributi navesti nekim redoslijedom

$$R(A_1, A_2, A_3, \dots, A_n)$$

gdje „n“ predstavlja broj atributa.

U praksi, za naziv relacije možemo izabrati bilo koji pogodan naziv. Isto važi i za nazive atributa, pod uslovom da su nazivi unikatni.

U primjeru biblioteka

AUTOR(SIFA, IME)

CLAN(SIFC, IME)

2.1. Struktura relacionog modela

Kartezijanski (Dekartov) proizvod.

Neka je data kolekcija skupova D_1, D_2, \dots, D_n (ne neophodno različitih).

Dekartov proizvod ovih n skupova

$D_1 \times D_2 \times \dots \times D_n$

je skup svi mogućih uređenih n -torki

$\langle d_1, d_2, \dots, d_n \rangle$, tako da je

$d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$.

Primer 1: $A = \{1, 2, 3, 4\}$, $B = \{4, 6, 8\}$

$A \times B = \{ \langle 1, 4 \rangle, \langle 1, 6 \rangle, \langle 1, 8 \rangle, \langle 2, 4 \rangle, \langle 2, 6 \rangle, \langle 2, 8 \rangle, \langle 3, 4 \rangle, \langle 3, 6 \rangle, \langle 3, 8 \rangle, \langle 4, 4 \rangle, \langle 4, 6 \rangle, \langle 4, 8 \rangle \}.$

Relacija.

Relacija definisana na “n” skupova je podskup Dekartovog proizvoda tih “n” skupova.

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

Podskup sadrži one n-torke Dekartovog proizvoda koje zadovoljavaju zadatu relaciju.

Primer 2: Neka je na skupovima A i B iz Primera 1 zadata relacija R:

$$A \times B = \{ \langle a, b \rangle \mid a = b/2 \}$$

$$R = \{ \langle 2, 4 \rangle, \langle 3, 6 \rangle, \langle 4, 8 \rangle \}$$

Domen relacije.

Skupovi D_1, D_2, \dots, D_n se nazivaju domenima relacije R .

Stepen relacije.

Broj domena na kojima je definisana neka relacija se naziva stepen relacije.

Razlikujemo **unarne** (na jednom domenu), **binarne** (na dva domena) i **n-arne** relacije).

Kardinalnost relacije je broj n -torki u relaciji.

Kako je Dekartov proizvod skup uređenih
n-torki, redosled elemenata u jednoj n-torki je bitan.

Na primer, ako na domenima

$$I1 = \{001, 007, 035\},$$

$$C1 = \{\text{Miloš}, \text{Ana}\},$$

$$I2 = \{19, 22\}$$

definišemo relaciju R

$$R \subseteq I1 \times C1 \times I2 = \{ \langle 001, \text{Miloš}, 19 \rangle, \\ \langle 007, \text{Ana}, 19 \rangle, \langle 035, \text{Ana}, 22 \rangle \},$$

u njoj je bitno da prvi element trojke uzima vrednost
iz prvog, drugi iz drugog, a treći iz trećeg skupa.

Međutim, ako vrednostima elemenata u n-torkama pridružimo imena domena (semantička, da bismo interpretirali odgovarajuće podatke, dajući sa istim ciljem i ime relaciji), možemo učiniti redosled elemenata u n-torkama bezznačajnim.

$STUDENT \subseteq BRIND \times IME \times STAROST =$
 $\{ \langle BRIND:001, IME:Miloš, STAROST:19 \rangle,$
 $\langle IME: Ana, BRIND: 007, STAROST:19 \rangle,$
 $\langle BRIND: 035, STAROST: 22, IME: Ana \rangle \}$

Atribut relacije. Imenovani domen, sa imenom koje definiše ulogu domena u relaciji se naziva atribut relacije. Atributi relacije STUDENT su BRIND, IME i STAROST.

Koncept atributa omogućuje predstavljanje relacije kao tabele.

Relacija STUDENT se može predstaviti sa:

STUDENT	BRIND	IME	STAROST
	001	Miloš	19
	007	Ana	19
	035	Ana	22

Pošto je relacija skup, a svaka tabela nije, definišu se sledeći uslovi koje tabela mora da zadovolji da bi bila relacija:

- (1) Ne postoje duplikati vrsta tabele;
- (2) Redosled vrsta nije značajan;
- (3) Redosled kolona nije značajan.

Pored toga, da bi se mogao definisati jednostavan skup operacija nad relacijama, definiše se sledeći dodatni uslov:

(4) Sve vrednosti atributa u relacijama su atomske, ili drugim rečima, nisu dozvoljeni atributi ili grupe atributa "sa ponavljanjem", odnosno nije dozvoljeno da vrednosti nekih atributa u relaciji budu relacije (nisu dozvoljene "tabele u tabeli").

Ako relacija zadovoljava uslov (4) tada je ona u Prvoj normalnoj formi. Svaka relacija u relacionom modelu mora biti u prvoj normalnoj formi.

Termin "normalizovana relacija" se koristi za relacije u prvoj normalnoj formi. (Za ostale normalne forme mora se precizirati o kojoj normalnoj formi se radi.)

Primer 3. Relacija (tabela) STUDISP, zapisana u sledećem obliku, nije normalizovana, jer su NAZIVPRED i OCENA "grupa sa ponavljanjem"

BRIND	IME	NAZIVPRED	OCENA
001	Miloš	Baze podataka	9
		Matematika	7
		Fizika	10
007	Ana	Fizika	7
		Matematika	9

Relacija STUDISP se može dovesti u prvu normalnu formu na sledeći način (uvodeći redundansu podataka):

BRIND	IME	NAZIVPRED	OCENA
001	Miloš	Baze podataka	9
001	Miloš	Matematika	7
001	Miloš	Fizika	10
007	Ana	Fizika	7
007	Ana	Matematika	9

Nula vrednosti.

Termin "nula vrijednost" (koji ćemo obeležavati sa ?) se koristi da označi "još nepoznatu vrijednost" za neki atribut ili "neprimenljivo svojstvo" za neki objekat ili vezu koje predstavlja tabela.

Na primer, u relaciji

NASLOV(SIFN, NAZIV, SIFO) unosimo podatke za naslov poznate šifre i naziva, ali ne znamo kojoj oblasti pripada. Atribut SIFO dobija "nula vrednost" (?) jer je njegovo ime "još nepoznato" i odgovarajući podatak će biti unet u bazu kada se sazna.

Pojava nula vrednosti koje su "neprimenljivo svojstvo" u nekoj relacionoj bazi podataka ukazuje na to da takva relaciona baza nije dobro projektovana.

Međutim, nula vrednosti kao "još nepoznate vrednosti" mogu se pojavljivati u bazi podataka.

Ključevi.

Činjenica da su sve n -torke u relaciji različite, govori da postoji jedan atribut ili više atributa zajedno (u krajnjem slučaju svi zajedno) čije vrednosti jedinstveno identifikuju jednu n -torku u relaciji (jednu vrstu u tabeli).

Taj atribut ili ta grupa atributa se nazivaju ključem relacije

(jedan atribut - prost ključ,

grupa atributa - složen ključ).

Ključ relacije se strožije definiše na sledeći način:

Ključ relacije R je takva kolekcija K njenih atributa koja zadovoljava sledeća dva uslova:

- Osobina jedinstvenosti. Ne postoje bilo koje dve n -torke sa istom vrednošću K .
- Osobina neredundantnosti. Ako se bilo koji atribut izostavi iz K , gubi se osobina jedinstvenosti.

4.2.1 Super ključ

Iz definicije relacije proizilazi da se u relaciji nikada ne mogu javiti dvije identične n-torke. Svakoj vrijednosti skupa atributa A_1, A_2, \dots, A_n šeme relacije $R=(A_i)$ odgovara jedna n-torka u relaciji r , odnosno skup vrijednosti a_1, a_2, \dots, a_n .

primjer

DRZI(SIFK, SIFC, DATUM)

Relacija koja sadrži: knjige koje drže članovi od kog datuma

Definicija

Super ključ neke šeme relacije R je svaki podskup atributa X te šeme koji ima osobinu da jednoznačno određuje n -torke u relaciji.

Iz osobine unikatnosti n -torke proizilazi da svaka šema relacije ima bar jedan super ključ. U krajnjem slučaju, kada postoji samo jedan super ključ, u njegov sastav ulaze svi atributi šeme relacije.

4.2.2 Kandidat ključ

Relacija JEAUTOR

JEAUTOR(SIFA, SIFN, KOJI)

Iz osobine unikatnosti n-torki u relaciji, je jasno da cijela šema, odnosno kombinacija atributa

SIFA, SIFN, KOJI predstavlja super ključ. Možemo zaključiti da je i kombinacija SIFA, SIFN super ključ, pošto se činjenica da je neki autor napisao neki naslov evidentira samo jednom n-torkom u relaciji. Međutim atributi SIFA i SIFN pojedinačno, a pogotovo koji, nisu super ključevi. Jedan autor može napisati više naslova, naslov može imati više autora. A mnogi autori će biti prvi, drugi itd za naslove koje su napisali. Pitanje je koji su super ključevi minimalni?

Definicija

Kandidat ključ neke šeme relacije R je svaki podskup atributa X te šeme koji ima osobinu da jednoznačno određuje n -torke relacije, a da ni jedan njegov pravi podskup nema tu osobinu.

Drugim riječima, kandidat ključ neke šeme relacije R je svaki njen super ključ koji ima osobinu minimalnosti. Takvih super ključeva može biti više za datu relaciju. Oni mogu biti bez ili sa zajedničkim atributima. Svaka šema relacije ima bar jedan super ključ.

primjer

OBLAST(SIFO, NAZIV)

kandidat i jedan i drugi atribut

NASLOV(SIFN, NAZIV, SIFO)

postoji kandidat ključ SIFN i NAZIV

4.2.3 Primarni ključ

U situacijama kada neka relacija ima više kandidat ključeva, za potrebe identifikacije n-torki relacije i realizaciju direktnog pristupa moramo da izaberemo jedan od njih.

Definicija

Primarni ključ neke šeme relacije R je jedan izabrani kandidat ključ, ili jedini kandidat ključ ako nema izbora. Svaka šema relacije ima jedan i samo jedan primarni ključ.

U situacijama kada neka šema relacije ima više kandidat ključeva, izbor PK se vrši po slijedećim kriterijumima:

- Prednost u odnosu na kandidat ključeve koji su poslijedica nekog dogovorenog ograničenja uvijek imaju kandidat ključevi koji su poslijedica suštinskog ograničenja (dogovorno ograničenje može da se promjeni, a suštinsko nikada).
- za kandidat ključeve koji su jedinstveni po prvom kriterijumu, prednost imaju oni koji zauzimaju manje memorijskog prostora u datoteci koja odgovara relaciji.

Primjer

Relacija OBLAST(SIFO, NAZIV),

PK SIFO jer je kraće.

Relacija DRZI(SIFK, SIFC, DATUM)

PK SIFK,

a kandidat ključ su atributi SIFK, SIFC.

Prednost se daje atributu SIFK, pošto knjiga van biblioteke može biti kod jednog člana, a dogovorno ograničenje pozajmice na jednu knjigu može biti ukinuto nekim novim pravilima biblioteke.

Primjer

Uobičajeno je da se u šemi relacije primarni ključevi naznače punom linijom, a drugi kandidat ključ se može podvući isprekidano.

OBLAST(SIFO, NAZIV),

Primarni ključ u tabeli nudi:

- Polju primarnog ključa je uvijek pridružen indeks koji znatno ubrzava, upite, pretraživanja i sortiranja.
- Prilikom unosa novog zapisa mora se popuniti polje primarnog ključa.
- Prilikom unosa novog zapisa u polje primarnog ključa nije moguće unijeti duplu vrijednost.
- Ako se ne zada drugačije Access uvek prikazuje sadržaj tabele po redoslijedu primarnog ključa.
- Preporučljivo je da polje primarnog ključa bude što kraće, kako bi se brže izvršavale operacije nad bazom.

4.2.4 Strani ključ (FOREIGN KEY)

Za objašnjavanje pojma stranog ključa potrebne su dvije relacije.

Primjer

NASLOV(SIFN, NAZIV, SIFO) i

OBLAST(SIFO, NAZIV).

Situacija da atribut SIFO u relaciji NASLOV, ima vrijednost nepostojeće oblasti iz relacije OBLAST, nije prihvatljiva.

Moguća je situacija da se nabavlja nova knjiga koja ima nalov, dodaje joj se šifra, a da se ne zna kojoj oblasti pripada. Tada se ta knjiga upisuje u bazu, a atributo SIFO se dodjeljuje NULL vrijednost koja se naknadno može unijeti.

Definicija

Strani ključ u šemi relacije R je svaki njen podskup atributa za koji važi ograničenje vrijednosti u relaciji r na sljedeće dvije vrijednosti:

- vrijednost primarnog ključa u nekoj relaciji (ciljnoj relaciji)
- vrijednost NULL.

Strani ključevi se mogu naglašavati iskošenim slovima.

Strani ključevi se mogu naglašavati masnim slovima:

CLAN (SIFC, **IME**)

POZAJMICA (SIFP, **SIFC**, **SIFK**, DANA)

KNJIGA (SIFK, **SIFN**)

REZERVACIJA (**SIFN**, **SIFC**, DATUM)

NASLOV (SIFN, NAZIV, **SIFO**)

JE_AUTOR (**SIFA**, **SIFN**, KOJI)

OBLAST (SIFO, NAZIV)

DRZI (**SIFK**, **SIFC**, DATUM)

AUTOR (SIFA, **IME**)

JE_REZERVISANA (**SIFK**, **SIFC**, DATUM)

Mogu se javiti situacije:

- jedna šema relacije može da sadrži više stranih ključeva.

POZAJMICA (SIFP, **SIFC**, **SIFK**, DANA)

- strani ključevi mogu biti u sastavu primarnog ključa.

- JEAUTOR(**SIFA**, **SIFN**, KOJI) i

REZERVACIJA(**SIFN**, **SIFC**, DATUM)

- strani ključ može biti istovremeno i primarni ključ u cjelini.

DRZI(**SIFK**, **SIFC**, DATUM)

Jednakost naziva atributa je poželjna sa gledišta preglednosti šeme relacione baze podataka, ali to nije uvijek moguće postići.

U suštini ono što povezuje SK i PK jeste jednakost značenja, ali jednakost naziva može, ali i ne mora biti zadovoljena.

Uz to, sve se može dešavati unutar jedne te iste šeme relacije.

Primjer

Posmatrajmo šemu relacije RADNIK kojom se pored zaposlenih evidentira i odnos nadređenosti u nekoj firmi, odnosno ko je kome šef:

RADNIK(SIFR, IME, ADRESA, **SIFNAD**)

SIFNAD predstavlja šifru šefa. Jedini radnik u firmi koji nema nadređenog je direktor. U ovoj šemi relacije SIFNAD je SK, pošto mora imati ili vrijednost postojeće šifre ili vrijednost NULL.

Može se uočiti:

- sve se dešava unutar jedne relacije koja sadrži oba učesnika u vezi PK i SK.
- SK i PK nemaju niti mogu da imaju isti naziv, ali zato imaju isto značenje identifikatora radnika.

Primjer baza *Rekurzija Jedan prema Vise*

4.2.5 Identifikacioni integritet

Identifikacioni integritet proizilazi iz osobine unikatnosti n -torki u relaciji i svodi se na formulaciju odgovarajućeg uslova.

Uslov Identifikacioni integriteta

Ni jedan atribut šeme relacije R koji je u sastavu primarnog ključa nikada ne smije imati NULL vrijednost u relaciji r .

Ovo proizilazi iz toga, kada bi dozvolili da dio primarnog ključa primi NULL vrijednost, mogla bi nastupiti situacija da dvije ili više n-torki (redova u tabeli) u relaciji postanu identične.

Primjer

Jeautor

<u>SIFA</u>	<u>SIFN</u>	KOJI
AP0	RBP0	1
AP0	PP00	1
DM0	RK00	2

Ako bi jedan dio primarnog ključa bio NULL nije problem, ali ako su dvije vrijednosti NULL to nije dozvoljeno u relacionom modelu.

Jeautor

<u>SIFA</u>	<u>SIFN</u>	KOJI
AP0	NULL	1
AP0	NULL	1
DM0	RK00	2

Zaključak

Kada je SK u sastavu PK ili ga čini u cjelini, NULL vrijednost nije dozvoljena.

Referencijalni integritet

- Referencijalni integritet je pravilo koje obezbjeđuje očuvanje veza između tabela u sistemu za upravljanje relacionim bazama podataka.
- Referencijalni integritet sprečava da ažurirate podatke u bazi podataka na način koji bi narušio veze između tabela.
- Tabela u kojoj je primarni ključ obično se zove *roditelj* a,
- Tabela u kojoj je spoljašnji ključ obično se zove *dete*.
- Referencijalni integritet obezbjeđuje da se nikada ne pojavi *siročće*, odnosno zapis u tabeli *dete* kome ne odgovara ni jedan zapis u tabeli *roditelj*.

Suština referencijalnog integriteta je u ograničenju stranog ključa. Uslov referencijalnog integriteta koji se odnosi na svako moguće stanje relacije baze podataka je statičke prirode i u suštini odgovara definiciji stranog ključa.

Uslov referencijalnog integriteta

Svaki podskup atributa šeme relacije R koji predstavlja strani ključ može u relaciji r imati ili vrijednost primarnog ključa u cijeloj relaciji ili vrijednost NULL.

Sa gledišta izmjena u relaciji koja sadrži strani ključ to podrazumjeva da važe slijedeća ograničenja:

- ne može se unijeti n-torka sa vrijednošću SK koja nije jednaka nekoj vrijednosti PK u cijeloj relaciji ili NULL vrijednost.

- ne može se izmjeniti n-torka tako da vrijednost SK ne bude jednaka nekoj vrijednosti PK u cijeloj relaciji ili NULL vrijednost.

Ono što je karakteristično za vezu „strani ključ-primarni ključ“ jeste to da referencijalni integritet podrazumjeva i ograničenja izmjena u cijeloj relaciji.

Dodavanje nove n-torke u cijeloj relaciji nikada ne može da naruši referencijalni integritet, pošto tada nastaje nova vrijednost PK.

Međutim, uklanjanje n-torke dovodi uvijek, a izmjena n-torke ponekad do nestanka jedne vrijednosti PK iz cijele relacije.

Ako bi se ta operacija izvršavala bezuslovno, to bi narušilo referencijalni integritet u slučaju da bilo gdje u bazi podataka postoje vrijednosti SK koje su jednake nestaloj vrijednosti PK.

Iz tih razloga se nameće potreba za specifikacijom u smislu:

kako treba postupiti u navedenim situacijama da bi referencijalni integritet bio očuvan.

Takva specifikacija se naziva dinamičkom specifikacijom referencijalnog integriteta.

Dinamička specifikacija referencijalnog integriteta

Neka je r ciljna relacija nad šemom R sa primarnim ključem X i neka je s relacija nad šemom S koja sadrži strani ključ Y .

Tada se posebno za svaku od operacija

- DELETE -uklanjanje n -torke,
- UPDATE-izmjena vrijednosti PK n -torke)

navodi jedna od slijedeće tri akcije u relaciji sa stranim ključem:

- RESTRICTED
- CASCADES
- NULLS

- **RESTRICTED**: operacija se ne izvršava, ako u relaciji *s* postoji vrijednost SK koja odgovara nestaloj vrijednosti PK iz ciljne relacije, a u suprotnom se izvršava.

- **CASCADES**: operacija se uvijek izvršava, za slučaj uklanjanja *n*-torke iz ciljne relacije *r*, iz relacije *s* se uklanjaju sve *n*-torke u kojima SK ima vrijednost koja odgovara nestaloj vrijednosti PK. Za slučaj izmjene vrijednosti PK u *n*-torci ciljne relacije *r*, u svim *n*-torkama relacije *s* u kojima SK ima vrijednost koja odgovara nestaloj vrijednosti PK, ta vrijednost se mijenja u novu vrijednost PK.

NULLS: operacija se uvijek izvršava, u relaciji *s* se u svim *n*-torkama u kojima SK ima vrijednost koja odgovara nestaloj vrijednosti PK ta vrijednost se mijenja u NULL vrijednost.

NULLS mogućnost se ne smije navoditi za strane ključeve koji su u sastavu primarnog ključa ili ga čine u cjelini.

Za slučaj relacione baze podataka, dinamička specifikacija referencijalnog integriteta se navodi za svaki strani ključ svake šeme relacije posebno. Opšta formula takve jedne specifikacije je

operacija OF *CiljnaŠema* akcija *StraniKljuč*

gdje operacija može biti: DELETE ili UPDATE.

A akcija može biti: RESTRICTED, CASCADES ili NULLS.

Primjer:

DELETE OF CLAN RESTRICTED

UPDATE OF CLAN CASCADES

Rijetka je situacija da jedna šema relacije ima više od jednog stranog ključa, u odnosu na istu cjelinu.

Zato se u navedenoj formuli naznaka *StraniKljuč* može izostaviti.

U rijetkim situacijama sa dva ili više SK u odnosu na istu ciljnu relaciju, **akcija** je obično ista za sve SK, što je dodatni razlog za izostavljanje naznake stranog ključa.

Primjer1

Posmatrajmo par šema relacija za mrežu željezničkih pruga:

STANICA (SIFS, NAZIV)

UDALJENOST (SIFSPOC, SIFSZAD, KM)

Šema relacije UDALJENOST sadrži dva strana ključa, u odnosu na istu ciljnu relaciju. To su šifra početne stanice i šifra zadnje stanice. Prirodno je da šema relacija UDALJENOST za oba strana ključa veže iste dinamičke specifikacije referencijalnog integriteta, i to:

DELETE OF STANICA RESTRICTED

UPDATE OF STANICA CASCADES

Primjer2

Za dio šeme BP Biblioteka date su dinamičke specifikacije referencijalnog integriteta i neophodni komentari:

NASLOV (SIFN, NAZIV, SIFO)

KNJIGA (SIFK, SIFN)

DELETE OF NASLOV RESTRICTED

Ne dozvoljava se uklanjanje naslova ako postoje knjige sa tim naslovom.

UPDATE OF NASLOV CASCADES

Promjena SIFN u NASLOVU, treba da automatski izazove promjenu u relaciji KNJIGA.

CLAN (SIFC, IME)

KNJIGA (SIFK, SIFN)

POZAJMICA (SIFP, SIFC, SIFK, DANA)

DELETE OF CLAN NULLS

Želimo da zadržimo podatke za knjige ako se briše neki član iz baze.

UPDATE OF CLAN CASCADES

Izmjena člana se vrši automatski u svim vezanim relacijama.

DELETE OF KNJIGA NULLS

Želimo da zadržimo podatke za članove, ako se knjiga briše.

UPDATE OF KNJIGA CASCADES

Izmjena podataka o knjizi, se vrši automatski u svim vezanim relacijama.

CLAN (SIFC, IME)

KNJIGA (SIFK, SIFN)

DRZI (SIFK, SIFC, DATUM)

DELETE OF KNJIGA RESTRICTED

Ne dozvoljava se ako je knjiga kod člana

UPDATE OF KNJIGA CASCADES

Izmjena podataka o knjizi, se vrši automatski u svim vezanim relacijama.

DELETE OF CLAN RESTRICTED

Ne dozvoljava se ako član drži neku knjigu.

UPDATE OF CLAN CASCADES

Izmjena podataka o CLANU, se vrši automatski u svim vezanim relacijama.

Za dinamičku specifikaciju referencijalnog integriteta je usvojeno da se podrazumjeva najoštrija RESTRICTED akcija, ako se ništa ne navede.

Treba naglasiti da se operacija, u cijeloj relaciji neke relacije baze podataka, sprovodi samo, ako ni u jednoj od relacija sa stranim ključevima nije nastupila RESTRICTED situacija odbijanja izmjene.

4.3 Manipulativna komponenta relacionog modela

Manipulativnu komponentu relacionog modela čini održavanje podataka i korišćenje podataka.

Za izmjenu sadržaja relacione BP koriste se 3 operacije:

- INSERT ubacivanje nove n-torke u relaciju
- DELETE uklanjanje n-torke iz relacije
- UPDATE izmjena vrijednosti jednog atributa u n-torci relacije.

Forma pisanja za ove operacije

- ubacivanje:

INSERT *Relacija, (Vrijednost1, ... , Vrijednost n)*

kao vrijednost se mogu zadati konstante ili varijable čije se vrijednosti učitavaju ili

- izračunavaju

Uklanjanje:

DELETE *Relacija, Uslov*

uslov može biti bilo kakav logički izraz

- izmjena

UPDATE *Relacija, Uslov, Atribut = Vrijednost*

Navedene tri operacije se mogu koristiti unutar raznih kontrolnih struktura:

- uslovi,
- grananja,
- Petlje.

Primjer

Za relacionu bazu podataka BIBLIOTEKA iz priloga A posmatrajmo slučaj održavanja kada neki član vraća pozajmljenu knjigu.

Prema opisu načina rada biblioteke. Kao prvo, treba očitati a zatim brisati podatke o tome da član drži knjigu kod sebe i evidentirati podatke o obavljenoj pozajmici. Nakon toga treba videti da li postoji rezervacija za naslov sa te knjige . Ako ona postoji, treba opslužiti prioritetsnu rezervaciju tako što će se evidentirati da je vraćena knjiga rezervisana za određenog člana.

Prvo treba da uočimo sa kojim relacijama se obavljaju koje elementarne operacije. To se, uz neophodne napomene, može provesti sledećim tabelarnim pregledom gde su sa znakom “?” označene operacije koje se eventualno izvršavaju:

Relacija	Operacije	Napomene
DRZI	Uvid, DELETE	Na osnovu SifK vracene knjige se ocitava SIFC i DATUM.
KNJIGA	Uvid	Na osnovu SifK vracene knjige se ocitava njeno SifN.
POZAJMICA	INSERT	Evidentira se pozajmica.
REZERVACIJA	Uvid, ? DELETE	Na osnovu SifN vracene knjige proverava se da li za taj naslov postoji neopsluzena rezervacija. Ako postoji, ositava se za kog je clana i rezervacija se brise.
JE_REZERVISANA	? INSERT	Ako je opsluzena rezervacija clana, evidentira se koji je to clan i koja knjiga

4.3.2 Korišćenje podataka

Manipulativna komponenta korišćenja podataka je dio relacionog modela podataka gdje njegov formalni karakter najviše dolazi do izražaja. Osnovu za taj dio relacionog modela predstavlja uža oblast Matematike poznata pod nazivom

"Predikatski račun I reda".

Iz te osnove izvedene su dve varijante formalnog predstavljanja korišćenja relacione baze podataka:

- relacionala algebra;
- relacioni račun.

