

Sistemi baza podataka

Sadržaj

SBP	1
Realizacija ograničenja šeme RBP putem SUBP	1
(Uvod) Realizacija integritetne komponente šeme BP	1
Specifikacija tipa ograničenja	2
Tipovi ograničenja u RMP	3
Specifikacija tipa ograničenja	3
Tipovi ograničenja u RMP	7
Specifikacija ograničenja	17

Realizacija ograničenja šeme RBP putem SUBP

Predavanje 0.

(Uvod) Realizacija integritetne komponente šeme BP

Cilj je da sva naša ograničenja organizujemo na nivou servera BP putem mehanizma SUBP, ideja je da prakticno zaštitimo sve naše podatke, tako da odakle god da im pristupimo da ne možemo da ih narušimo.

Kontrola ograničenja, implementiranih na nivou SUBP je centralna to u suštini znači da ni jedan softver ni program ne može spolja da ih naruši, korisnici nisu svesni ograničenja sve dok ih ne naruše. Primer prolazimo kroz aerodrom i mi bivamo iskontrolisani a da nismo ni svesni dok ne narušimo neko ograničenje, isto funkcioniše i SUBP. U slučaju da narušimo nešto SUBP će obaviti jednu od ove dve operacije:

- obavestice nas o greski (korisnicki program i samog korisnika) a potom prekinuti izvršavanje
- nekim aktivnim mehanizmom ce obaviti dalje operacije nad BP da bi tu BP vratio u konzistentno stanje

Implementacija ograničenja BP (na serveru BP), moramo da definiše parametre u vezi ograničenja (tj da ih saopštimo serveru),

parametri su:

- definiše se ograničenje datog tipa
- definišu se operacije tj. događaji koji mogu dovesti do narušavanja ograničenja
- za svaku operaciju je potrebno definisati akciju koju ćemo preduzeti da bi očuvali konzistentnost u slučaju narušavanja ograničenja
- svako ograničenje je vezano za određenja obeležja i određene šeme relacija šeme BP
- operacije(događaji) koji mogu dovesti do narušavanja ograničenja:
 - o Upis (nove torke u relaciju)
 - o Brisanje (neke postojeće torke)
 - o Modifikacija (postojeće torke u relaciji)

Pitanje: Zašto upiti nisu kritična operacija? Upiti ne menjaju stanje, samo konstatuju koji su to podaci u BP, ako su upiti ispravni sve ce biti ok ako nisu ispravni samo će pokazati da nisu ispravni.

Pitanje: Kada možemo da donesemo pretpostavku da će nam upiti vratiti uvek ispravne podatke ? Ako krenemo od početka i ako kontroliramo svaki upis, brisanje i modifikaciju, posledično svaki naš upit će vratiti konzistentno ispravne podatke. U životu ovo baš nije uvek tako pošto često preuzimamo nasleđene sisteme.

Akcije očuvanja konzistentnosti BP u slučaju pokušaja narušavanja ograničenja, vezuju se sa svaku operaciju koja može narušiti ograničenje, dele se na:

- Aktivne, obezbeđuju automatsko sprovođenje daljih operacija ažuriranja nad BP, koje će obezbediti očuvanje konzistentnosti BP
- Pasivne, sprečavaju operaciju koja bi narušila ograničenje
- Kombinovane, pod određenim uslovima pasivne a pod određenim uslovima aktivne

Ograničenja se implementiraju pomoću mehanizama SUBP, imaćemo način da definišemo ograničenje i moraćemo da saopštimo DBMSu kako smo definisali ograničenje. Kada imamo kritične operacije koje mogu narušiti ograničenje, onda opet moramo saopštiti DBMSu koje su to kritične operacije, i za svaku kritičnu operaciju moramo saopštiti koja akcija očuvanja konzistentnosti BP će se primeniti.

Kontrola ograničenja šeme BP, SUBP pokreće mehanizme kontrole važenja ograničenja

- automatski
- nakon izvođenja kritične operacije za ograničenje

Prednosti:

- Automatska kontrola implementiranih ograničenja na nivou SUBP
- Imamo obezbeđenu konzistentnost BP u svakom trenutku
- Standardizacija načina za implementaciju ograničenja (sql2, sql3, etc...)
- Kada postavimo implementaciju ograničenja na nivo servera baze podataka onda možemo da kažemo da nemamo dodatnih razloga zašto bi realizovali ograničenja unutar samih korisničkih programa na srednjem sloju, ili na klijentskom sloju. **Znači implementiramo ograničenja jednom, a ne milion puta na različitim aplikacijama.**

Pitanje: Zašto je ovo jako dobro? Rasterećujemo programere i olakšavamo samu izradu aplikacije kao i njeno održavanje koje jako puno košta.

Nedostaci:

- Povišen stepen zavisnosti šeme BP od proizvođača, tipa i verzije SUBP-a, ne podržavaju svi sistemi za upravljanje BP standarde na isti način, standardi generalno nisu idealni, to znači mi ako želimo da koristimo kapacitete DBMS-a za implementaciju ograničenja na nivou servera BP nužno ćemo istrčavati izvan onoga što definiše standard, znači ako želimo da migriramo bazu sa jednog na drugi DBMS moraćemo da preradimo te delove programskog koda, postajemo više zavisni od šeme BP i proizvođača tipa i verzije. **Zato je izbor DBMS-a strateski potez jedne kompanije!**
- Ostaje uvek potreba za realizacijom nekih ograničenja unutar programa, zbog obezbeđenja pogodnosti programa za upotrebu, i dvostruke implementacije i kontrole ograničenja. Recimo ako pravimo interfejs bolje pogodnosti (HCI priča) obično želimo da obavestimo korisnika da pravi grešku u trenutku kada je pravi a ne kada klikne submit. Onda imamo potrebu da ista ograničenja implementiramo i na frontu a ne samo na serveru BP.

Predavanje 1.

Specifikacija tipa ograničenja

Karakteristike tipa ograničenja u modelu podataka (MP)

- Oblast definisanosti, to je tip logičke strukture nad kojom se ograničenje definiše

- Oblast interpretacije, to je tip logičke strukture nad kojom se ograničenje interpretira
- Formalizam za zapisivanje, to je nekakav parametrizovani šablon na osnovu kog pišemo ograničenja
- Pravilo za interpretaciju odnosno validaciju
- Skup kritičnih operacija nad bazom podataka koje mogu dovesti do narušavanja ograničenja datog tipa, mi idemo sa pretpostavkom da imamo praznu bazu podataka
- Skup mogućih akcija, definiše se za svaku kritičnu operaciju

Specifikacija tipa ograničenja u MP

TipO(T(t), TOd, TOi, TFz, TPi)

- *TipO* - oznaka tipa ograničenja
 - *T(t)* - definicija tipa logičke strukture obeležja
 - » s uključenim kritičnim operacijama i mogućim akcijama
 - *TOd* - specifikacija oblasti definisanosti
 - *TOi* - specifikacija oblasti interpretacije
 - *TFz* - definicija formule za zapisivanje
 - *TPi* - definicija pravila za interpretaciju
- Sve komponente specifikacije TipO treba da budu specificirane formalno(u cilju njihove lakše programske implementacije). U ove potrebe potrebno je koristiti koncepte nekog konkretnog MP (mi ćemo koristiti primer relacionog modela podataka).
 - TipO – oznaka tipa ograničenja, to je nekakav niz znakova, ne postoje neka konkretna pravila kako zadajemo taj niz znakova ali očekujemo da taj niz znakova jedinstveno identifikuje svaki uvedeni tip ograničenja, formira se sasvim slobodno na osnovu naziva tipa ograničenja u datom MP

Tipovi ograničenja u RMP

Mogući tipovi ograničenja u RMP

- s pridruženim oznakama (*TipO*)

• ograničenje domena	(<i>DomCon</i>)	
• ograničenje vrednosti obeležja	(<i>AttValCon</i>)	
• ograničenje torke	(<i>TupleCon</i>)	
• prošireno ograničenje torke	(<i>ExTupleCon</i>)	
• ograničenje ključa	(<i>KeyCon</i>)	
• ograničenje jedinstvenosti	(<i>UniqueCon</i>)	ograničenje referencijalnog integriteta
• zavisnost sadržavanja	(<i>InCon</i>)	» prošireno, selektivno, selektivno i prošireno
• proširena zavisnost sadržavanja	(<i>ExInCon</i>)	» <i>RefInCon</i> , <i>ExRefInCon</i> , <i>SelRefInCon</i> , <i>SelExRefInCon</i>
• selektivna zavisnost sadržavanja	(<i>SelInCon</i>)	ograničenje inverznog referencijalnog integriteta
• selektivna proširena zavisnost sadrž.	(<i>SelExInCon</i>)	» prošireno, selektivno, selektivno i prošireno
		» <i>InvRfCon</i> , <i>ExInvRfCon</i> , <i>SelInvRfCon</i> , <i>SelExInvRfCon</i>
- Specijalne vrste zavisnosti sadržavanja (ograničenje referencijalnog integriteta i ograničenje inverznog referencijalnog integriteta)
- to su korisnički definisani, za koje kažemo da u njih spadaju svi koji prethodno nisu definisani, štos je u tome da što manje toga ostane u ovoj kategoriji a da što više ograničenja pokrijemo sa unapred definisanim.

Pitanje: Zašto imamo ovoliki spisak definisanih ograničenja? Rasterećujemo projektante, tj u suštini mi smo unapred prepoznali više situacija u realnom sistemu, i lakše ih modelujemo.

Pitanje: Zašto uopšte uvodimo pojam „tip ograničenja“ ? Zato što uvođenjem ovog pojma mi uvodimo znanje o tome kako treba specificirati, u koliko nemamo pojam tipa ograničenja svaki projektant može da radi šta hoće...

Specifikacija tipa ograničenja

Specifikacija tipa ograničenja u RMP

Tip ograničenja je za nas formalna struktura koja obuhvata sledećih 5 elemenata:

$$TipO(T(t), TOd, TOi, TFz, TPI)$$

– $T(t)$ – definicija tipa logičke strukture obeležja

- obavezna komponenta specifikacije
- strukturirana putem formule
 - $T(t) = \{(Role_1, Mult_1, AtStr_1, AtMult_1, \{(op_1^{i1}, sact_1^{i1}) \mid i_1 \geq 1\}), \dots, (Role_m, Mult_m, AtStr_m, AtMult_m, \{(op_m^{im}, sact_m^{im}) \mid i_m \geq 1\})\}$
 - Definicija strukture elementa

$$(Role_j, Mult_j, AtStr_j, AtMult_j, \{(op_j^{ij}, sact_j^{ij}) \mid i_j \geq 1\})$$

Nad definicijom tipa logičke strukture obeležja se kasnija ograničenja datog tipa definišu, moramo da je specificiramo (ne smemo da je preskočimo), to je nekakav skup n-torki, dozvoljeno je čak da $T(t)$ bude prazan skup.

Svaki tip logičke strukture obeležja se definiše se skupom uloga:

$$T(t) = \{(Role_1, Mult_1, AtStr_1, AtMult_1, \{(op_1^{i1}, sact_1^{i1}) \mid i_1 \geq 1\}), \dots, (Role_m, Mult_m, AtStr_m, AtMult_m, \{(op_m^{im}, sact_m^{im}) \mid i_m \geq 1\})\}$$

a svaku tu ulogu definišemo jednom ovakvom n-torkom:

$$(Role_j, Mult_j, AtStr_j, AtMult_j, \{(op_j^{ij}, sact_j^{ij}) \mid i_j \geq 1\})$$

to će ustvari biti uloga nekih šema relacija u našem ograničenju.

Definicija strukture elementa:

$$(Role_j, Mult_j, AtStr_j, AtMult_j, \{(op_j^{ij}, sact_j^{ij}) \mid i_j \geq 1\})$$

- $Role_j$ - niz znakova
 - » određuje moguću ulogu šeme relacije u ograničenju
 - » Δ - oznaka za "nedefinisano" (nebitnu) ulogu
 - » svi upotrebljeni $Role_j$ u istom $T(t)$, moraju imati jedinstvenu oznaku (naziv) uloge
- $Mult_j \in \{0, 1, n, *\}$
 - » specifikacija dozvoljenog broja šema relacija u ograničenju, s datom ulogom
 - » 0 – ni jedna, 1 – tačno jedna, n – tačno n, * – jedna ili više
- $AtStr_j \in \{set, array\}$
 - » da li se obeležja ograničenja date šeme relacije strukturiraju u skup (set), ili niz (array)
- $AtMult_j \in \{0, 1, *\}$
 - » specifikacija dozvoljenog broja obeležja šeme relacije u ograničenju, s datom ulogom
 - » 0 – ni jedno, 1 – tačno jedno, * – jedno ili više
- $\{(op_j^{ij}, sact_j^{ij}) \mid i_j \geq 1\}$
 - » skup svih kritičnih operacija i mogućih akcija
- $op_j^{ij} \in \{ins, upd, del\}$
 - » kritična operacija, koja može narušiti ograničenje
- $sact_j^{ij} \subseteq \{NoAction, Cascade, SetDefault, SetNull, <<UserDef>>\}$
 - » skup mogućih akcija očuvanja konzistentnosti BP pri pokušaju narušavanja ograničenja putem op_j^{ij}

Tipovi akcija očuvanja konzistentnosti

Pri očuvanju narušavanja ograničenja BP, akcije mogu biti:

- Pasivne akcije
 - NoAction (Restrict) , ova akcija zabranjuje sprovođenje operacije koja bi izazvala narušavanje kontrolisanog ograničenja. To znači npr. ako bi neko operacijom ažuriranja narušio neko ograničenje, ova akcija bi rekla zaustavi i poništi izvršavanje ove operacije.
- Aktivne akcije
 - Cascade, kaskadna propagacija operacije, npr. ako je u pitanju brisanje, kaskadno će se nastaviti brisanje kroz druge relacije.. isto i za modifikaciju itd...
 - SetNull, svodi kritične vrednosti atributa na nula vrednosti
 - SetDefault, svodi kritične vrednosti atributa na predefinisane inicijalne default vrednosti
 - <<UserDef>> , kaže napiši onu proceduru, onu logiku koja tebi kao projektantu odgovara za datu situaciju. Posebno su isprojektovane, isprogramirane i specificirane unapred određenom sintaksom.

Specifikacija tipa ograničenja u RMP

$$TipO(T(t), TOd, TOi, TFz, TPi)$$

– TOd – specifikacija oblasti definisanosti

- saglasno klasifikaciji tipova ograničenja prema **oblasti definisanosti** ograničenja
 - prema "širini" logičke strukture obeležja
 - povezana s definicijom elementa $T(t)$
 - $TOd \in \{0, 1, n, *\}$
 - » 0 – vanrelacioni tip ograničenja
 - » 1 – jednorelacioni tip ograničenja
 - » n, * – višerelacioni tip ograničenja

U okviru oblasti definisanosti mi govorimo koja je to širina te logičke strukture obeležja koja je povezana sa definicijom naših elemenata u tipu logičke strukture obeležja, pa govorimo o vanrelacionim, jednorelacionim i višerelacionim tipovima ograničenja.

Oblasti definisanosti tipa ograničenja u RMP

- vanrelacioni (bezrelacioni) tip ograničenja, ovo ograničenje se ne definiše nad šemama relacije šeme BP, nego se definiše nad predefinisanim ili korisnički definisanim domenima (tipovima)
 - $T(t) = \emptyset$
- jednorelacioni (unutarrelacioni, lokalni) tip ograničenja, ograničenje se definiše nad tačno jednom šemom relacije
 - $|T(t)| = 1$
 - $Mult_j = 1$
 - uloga jedine šeme relacije N_j u $T(t)$ je nebitna ($Role_j = \Delta$)
- višerelacioni tip ograničenja, ograničenje se definiše nad skupom šema relacija, koji sadrži bar dva člana
 - $|T(t)| \geq 1$
 - ne postoje posebna ograničenja na zadavanje mogućih uloga šema relacija $Role_j$ u strukturi $T(t)$
 - u skupu šema relacija ograničenja ovog tipa može se pojaviti ista šema relacije više od jedanput
 - » u tom slučaju, svaki put, sa različitom ulogom
 - » ovakva situacija je moguća samo kada je $|T(t)| > 1$

Specifikacija tipa ograničenja u RMP

$$TipO(T(t), TOd, TOi, TFz, TPi)$$

– TOi – specifikacija oblasti interpretacije

- saglasno klasifikaciji tipova ograničenja prema oblasti interpretacije
 - prema "dubini" potrebnog dela BP, nad kojim se ograničenje interpretira
 - $TOi \in \{v, t, r, m\}$
 - » v – tip ograničenja **vrednosti**
 - » t – tip ograničenja **torke**
 - » r – **relacioni** tip ograničenja (tip ograničenja **relacije**)
 - » m – **međurelacioni** tip ograničenja

Oblasti interpretacije tipa ograničenja u RMP

- tip ograničenja vrednosti, interpretira se nad tačno jedno vrednošću nekog budućeg obeležja.

- tip ograničenja torke, intepetira se nad jednom torkom bilo koje relacije.
- relacioni tip ograničenja, intepetira se nad skupom torki bilo koje relacije.
- međurelacioni tip ograničenja, intepetira se nad barem dve, bilo koje relacije, koje ne moraju biti međusobno različite. **Napomena** bilo koja relacija znači relacija iz baze podataka ili neka relacija koja predstavlja definisani pogled. (Pogled je relacija koja je nastala primenom izrazne relacione algebre nad jednom ili više drugih relacija, ili primenom operatora spajanja)

Specifikacija tipa ograničenja u RMP

$TipO(T(t), TOd, TOi, TFz, TPI)$

– TFz – definicija formule za zapisivanje

- gramatika za formiranje zapisa ograničenja datog tipa
 - zavisi od definicije elementa $T(t)$
 - koncepti, korišćeni u zapisu formule
 - » oznake šema relacija
 - » oznake obeležja
 - » oznake operatora
 - » različiti tipovi logičkih izraza
 - često se zadaje putem parametrizovanog šablona formule

Specifikacija tipa ograničenja u RMP

$TipO(T(t), TOd, TOi, TFz, TPI)$

– TPI – definicija pravila za interpretaciju

- predikatska formula
 - definiše kada je ograničenje zadovoljeno
 - čiji su operandi
 - » oznake relacija
 - » oznake torki i restrikcija torki
 - zavisi od definicije formule za zapisivanje TFz
 - interpretira se (validira), često, u svetu trovalentne logike
 - » true – ograničenje zadovoljeno
 - » false – ograničenje narušeno
 - » null – nepoznata interpretacija ograničenja

I na kraju sve to samo tabelarno prikazno:

$TipO$	naziv tipa	duži naziv tipa ograničenja
$T(t)$	$Role_1$	$Mult_1$ $AtStr_1$ $AtMult_1$
	op_1^1	$sact_1^1$ (skup svih dozvoljenih akcija)
	op_1^2	$sact_1^2$ (skup svih dozvoljenih akcija)
	...	
	$Role_m$	$Mult_m$ $AtStr_m$ $AtMult_m$
	op_m^1	$sact_m^1$ (skup svih dozvoljenih akcija)
	op_m^2	$sact_m^2$ (skup svih dozvoljenih akcija)
TOd	x	opis oblasti definisanosti (0, 1, n, *)
TOi	y	opis oblasti interpretacije (v, t, r, m)
TFz	parametrizovani šablon formule za zapisivanje	
TPI	specifikacija pravila za interpretaciju	

Tipovi ograničenja u RMP

Ograničenje domena

– $DomCon(\emptyset, 0, v, TFz, TPi)$

O ovom ograničenju bi trebalo sve već da znamo iz baza podataka 1 sada radimo samo rekapitulaciju.

- Tip logičke strukture - $T(t) = \emptyset$

Pre svega ograničenje domena je bez relaciono ograničenje (precrtana nula u zagradi kod DomCon) što znači nema nikakve posebne tipove logičke strukture obeležja nad kojim se definiše.

- Tip definisanosti TOd - *vanrelaciono ograničenje*

Ne definiše se ni na jednom jedinom šemom relacije (nula u zagradi kod DomCon)

- Tip interpretacije TOi - *ograničenje vrednosti*

Definiše se kao ograničenje vrednosti, za njegovu interpretaciju trebaće nam bilo koja vrednost.

- Šablon za tip zapisa TFz

- zavisi od vrste ograničenja domena

- za primitivni domen $TFz = niz_znakova$

- » predstavlja oznaku (naziv) primitivnog domena u RMP

- za korisnički definisani domen, TFz zavisi od izabranog pravila

- » nasleđivanja

- » tipa sloga

- » tipa skupa (kolekcije)

- » tipa izbora

- » tipa pokazivača (reference)

- u RMP, za korisnički definisani domen i pravilo nasleđivanja

- $TFz = id(D) = (Tip, Dužina, Uslov)$

- » $id(D)$ – ograničenje domena s nazivom D

- ostala pravila – proširenje RMP u O-R MP

- Pravilo za interpretaciju TPi

- prema definiciji u RMP

- interpretira se za svaku moguću vrednost d

- » $id(D)(d) = (Tip, Dužina, Uslov)(d) =$
 $Tip(d) \wedge Dužina(d) \wedge Uslov(d)$

TipO	DomCon	ograničenje domena
$T(t)$	\emptyset	
TOd	0	vanrelaciono ograničenje
TOi	v	ograničenje vrednosti
TFz	$id(D) = (Tip, Dužina, Uslov)$	
TPi	$id(D)(d) = (Tip, Dužina, Uslov)(d) =$ $Tip(d) \wedge Dužina(d) \wedge Uslov(d)$	

Vrednosti tip, dužina i uslov primenjujemo na bilo koje vrednosti, primenjujemo ih prvo kao proveru tipa pa proveru dužine pa proveru uslova. Ovo “and” nije ono and koje mi poznajemo nego “cond” (conditional and), “conditional and” ne zahteva da svaki operand u svakom trenutku bude interpretabilan, redosled pisanja je važan, npr ako leva strana vrati false desna strana ne mora uopšte da bude interpretabilna. To takodje znaci da ukoliko imamo da je neki od operandi izostavljen (npr. Uslov je izostavljen, tj. trouglic) to znaci da je on **true**!

Ograničenje vrednosti obeležja

Prvi put uvodimo sada tip logičke strukture obeležja koji neće biti prazan skup, ograničenje vrednosti obeležja definišemo kao jednorelaciono ograničenje. To znači da će biti definisano nad jednom šemom relacija u krajnjoj liniji i da će imati svoju formulu za zapisivanje i pravilo za interpretaciju.

– AttValCon($T(t)$, 1, v , TFz , TPi)

- Tip logičke strukture
 - $T(t) = \{(\Delta, 1, set, 1, \{(ins, sact_i), (upd, sact_u)\})\}$
 - » nebitna uloga, koristi se za tačno jednu šemu relacije i jedno obeležje
 - » $sact_i, sact_u = \{NoAction, SetNull, SetDefault, <<UserDef>>\}$
- Tip definisanosti TOd - *jednorelaciono ograničenje*
- Tip interpretacije TOi - *ograničenje vrednosti*
- Šablon za tip zapisa TFz
 - $TFz = \pi(N, A) = (id(D), NullSpec)$
 - » $NullSpec \in \{Null, NotNull\}$
 - » specifikacija dozvole ili zabrane nula vrednosti

TipO	AttValCon	ograničenje vrednosti obeležja
$T(t)$	Role, Δ	Mult, 1, AtStr, set, AtMult, 1
	ins	NoAction, SetNull, SetDefault, <<UserDef>>
	upd	NoAction, SetNull, SetDefault, <<UserDef>>
TOd	1	jednorelaciono ograničenje
TOi	v	ograničenje vrednosti
TFz	$\pi(N, A) = (id(D), NullSpec)$	
TPi	$\pi(N, A)(d) = (id(D), NullSpec)(d) = id(D)(d) \wedge NullSpec(d)$	

Tip logičke strukture obeležja se sastoji iz jedne jedine komponente, znači definišemo jednu jedinu ulogu I kažemo da je naziv te uloge nebitan (znak delta u zagradi kod $T(t)$), dalje kažemo da će se u instanci pod tom nebitnom ulogom pojavljivati samo jedna šema relacije (broj jedan u zagradi kod $T(t)$), I da će se pojavljivati skup od samo jednog atributa (set u zagradi kod $T(t)$), I da su kritične operacije ins I upd (insert I update), za kritičnu operaciju insert I update definišemo skup mogućih akcija (NoAction, SetNull, SetDefault, <<UserDef>>) tj definišemo šta se može primeniti u slučaju da neko naruši ograničenje.

Šablon zapisa formule je onaj koji već znamo iz BP1, ograničenje vrednosti obeležja ovde označeno kao $\pi(N, A)$

Definisemo kao par čija prva komponenta je ograničenje domena a druga komponenta je specifikacija ograničenja nula vrednosti $(id(D), NullSpec)$

- Pravilo za interpretaciju TPi
 - prema definiciji u RMP
 - interpretira se za svaku moguću vrednost d
 - » $\pi(N, A)(d) = (id(D), NullSpec)(d) = id(D)(d) \wedge NullSpec(d)$

Ograničenje torke

Ograničenje torke podrazumeva da moramo da imamo ograničenje svakog obeležja u šemi relacije i još dodatno ako postoji, zadajemo logički uslov, koji vrednosti obeležja u bilo kojoj torci moraju međusobno da zadovoljavaju.

– TupleCon($T(t)$, 1, t , TFz , TPi)

- Tip logičke strukture
 - $T(t) = \{(\Delta, 1, set, *, \{(ins, sact_i), (upd, sact_u)\})\}$
 - » nebitna uloga, koristi se za tačno jednu šemu relacije, nad skupom više obeležja
 - » $sact_i, sact_u = \{NoAction, SetNull, SetDefault, <<UserDef>>\}$

Tip logičke strukture obeležja opet kaže da i dalje radimo sa jednom jedinom ulogom šeme relacije čiji je naziv nebitan, set * znači da radimo nad skupom od više atributa, i ovo ograničenje ima svoje kritične operacije (ins, update) kao i skup skup akcija koje se okidaju kada neko pokuša da naruši to ograničenje.

- Tip definisanosti TOd - *jednorelaciono ograničenje*
 - Ovo je i dalje jednorelaciono ograničenje jer se definiše nad jednom jedinom šemom relacije sa nebitnom ulogom.
- Tip interpretacije TOi - *ograničenje torke*
 - Ograničenje torke zato što koristimo sada više obeležja (skup obeležja jedne šeme relacije), što znači da njegova interpretacija ne može da ide nad jednom jedinom vrednošću, nego moramo za njegovu interpretaciju da podmetnemo celu torku da bi ograničenje bilo validirano

- Šablon za tip zapisa TFz

- $TFz = \tau(N) = (\{\tau(N, A) \mid A \in R\}, Con(N))$
 - » N – oznaka šeme relacije za ograničenje
 - » R – oznaka skupa svih obeležja šeme relacije N
 - » $Con(N)$ – logički uslov ograničenja torke

Ograničenje torke definišemo sa nekim tau od en, i onda kažemo da je prva komponenta ovog ograničenja skup svih ograničenja vrednosti obeležja za sva obeležja iz šeme relacije koja je navedena, iz njenog skupa obeležja R , a druga komponenta $Con(N)$ je logički uslov ograničenja torke (logički constraint).

- Pravilo za interpretaciju TPi

- prema definiciji u RMP
- interpretira se za svaku pojedinačnu torku t iz relacije $r(N)$
 - » $t \in r(N)$
 - » $\tau(N)(t) = (\{\tau(N, A) \mid A \in R\}, Con(N))(t) =$
 $(\forall A \in R)(\tau(N, A)(t[A])) \wedge Con(N)(t)$

Ograničenje se interpretira za bilo koju torku iz relacije nad datom šemom relacije. Da bi smo ga interpretirali potreban nam je sadržaj cele moje torke (t element $r(N)$). Primenjujemo specifikaciju ograničenja nad jednom instanciranom torkom iz relacije ($\tau(N)(t)$), i primenjujemo tu torku nad celom specifikacijom

$(\{\tau(N, A) \mid A \in R\}, Con(N))(t)$, tako što za svaki atribut iz skupa atributa moje šeme relacije primenim ograničenje vrednosti obeležja nad onom vrednošću torke koja odgovara tom mom obeležju, onda opet imamo conditional and koji kaže ukoliko su ti ograničenja vrednosti obeležja zadovoljena za sva obeležja tvoje šeme relacije onda proveriti i da li je logički uslov ograničenja torke definisan u ovom ograničenju takođe validan za ovu torku t .

Sve ovo od gore prikazano u tabelarnom obliku:

TipO	TupleCon		ograničenje torke					
T(t)	Role ₁	Δ	Mult ₁	1	AtStr ₁	set	AtMult ₁	*
	ins	NoAction, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, SetNull, SetDefault, <<UserDef>>						
TOd	1	jednorelaciono ograničenje						
TOi	t	ograničenje torke						
TFz	τ(N) = ({ τ(N, A) A ∈ R }, Con(N))							
TPi	τ(N)(t) = ({ τ(N, A) A ∈ R }, Con(N))(t) = (∀ A ∈ R)(τ(N, A)(t[A])) ∧ Con(N)(t)							

Pitanje: Kako intepetiramo ograničenje torke i kada ga vršimo ? Potrebna su nam ograničenja vrednosti svakog obeležja u toj torci(potrebna su nam ograničenja vrednosti dva ili više obeležja unutar iste torke). Da bi smo evaluirali, potrebna nam je vrednost cele torke (da bi na njoj to mogli i izvršiti).

Pitanje: Kritične operacije su insert I update, zašto delete nije ? Zato što je samo unutar jedne torke ograničenje. Ako obrišemo, nećemo narušiti ništa unutar neke druge torke, tj zato što pravilo interpretacije ne dotiče ni jednu drugu torku osim te torke koje se tiče to ograničenje.

Pitanje: Zašto nema Cascade u skupu kritičnih operacija? Zato što je jednorelaciono ograničenje I zato što pravilo interpretacije ne dotiče ni jednu drugu torku osim te torke koje se tiče to ograničenje.

Prošireno ograničenje torke

Ima dosta dodirnih tačaka sa običnim ograničenjem torke, samo sada možemo taj logički uslov da protegnemo na dve ili više šema relacija.

– $ExTupleCon(T(t), *, t, TFz, TPI)$

- Tip logičke strukture
 - $T(t) = \{(\Delta, *, set, *, \{(ins, sact_i), (upd, sact_u)\})\}$
 - » nebitna uloga, koristi se za više šema relacija, nad skupom više obeležja, svake od njih
 - » $sact_i, sact_u = \{NoAction, SetNull, SetDefault, <<UserDef>>\}$

Primer o kom priča profesor na predavanju vezano za banku, ekpoziture i blagajne, poenta priče i **pitanje** je koje ograničenje je potrebno uvesti da bi zabranili blagajnici da isplati veću sumu novca nego što je stanje blagajne ? Kako ćemo iskazati to ograničenje ?

To ograničenje je između dva obeležja, između različitih šema relacije, jedno je njeno stanje blagajne na dan i smena koju je otvorila na početku, a drugo je nalog za transakciju šeme relacije u kojoj postoji iznos transakcije. Iznos transakcije iz šeme relacije transakcija povezujemo sa stanje blagajne šeme relacije stanje blagajne na dan. I kažemo da stanje blagajne mora da bude veće ili jednako od bilo koje izlazne transakcije isplate novce. Kojim ograničenjem ovo može da se iskaže? Da li može običnim ograničenjem torke? Pa ne može zato što obuhvata dve šeme relacija. Potreban nam je neki vid spajanja sa torkom stanja blagajne.

Tip logičke strukture obeležja takođe obuhvata jednu ulogu (Δ u $T(t)$) ali se u toj jednoj ulozi može pojaviti više šema relacija ($*$ u $T(t)$), i nad tih više šema relacija na dalje se može pojaviti skup obeležja iz više šema relacija ($set, *$ u $T(t)$), kritične operacije su update i delete. Skupovi mogućih akcija su definisani iznad.

- Tip definisanosti TOd - *višerelaciono ograničenje*
- Tip interpretacije TOi - *ograničenje torke*

Ovo je sada višerelaciono ograničenje ali je i dalje ograničenje torke jer se validira na jednom jedinom torkom.

- Šablon za tip zapisa TFz

- $TFz = \tau_{ex}(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m) = Con(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)$
 - » logički uslov proširenog ograničenja torke
 - » definiše nad skupom obeležja koja pripadaju uniji $\cup_{i=1}^m (R_i)$

Tau ex ograničenje torke nad šemama relacija koje dovodimo u neku vezu zbog prirodnog spajanja

$$\tau_{ex}(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)$$

I definišemo logički uslov koji definiše kako ćemo proveravati vrednosti naših obeležja nad nekom torkom

$$Con(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)$$

- Pravilo za interpretaciju TPI

- prema definiciji u RMP
- interpretira se za svaku pojedinačnu torku t koja pripada spoju relacija $r(N_1) \triangleright \triangleleft \dots \triangleright \triangleleft r(N_m)$
 - » $t \in r(N_1) \triangleright \triangleleft \dots \triangleright \triangleleft r(N_m)$
 - » $\tau_{ex}(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)(t) = Con(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)(t)$

Pravilo interpretacije kaže sledeće, uzmi nad svim prethodno iskorišćenim šemama relacija (spoj tih relacija), bilo koju torku iz te relacije koja je nastala prirodnim spajanjem tih relacija i intepetiraj to ograničenje nad tom torkom tako što ćeš taj logički uslov intrepetirati tačno nad njom. Šta to znači ? To znači da mi imamo sadržaje šema relacija, napravimo spoj, i svaka torka iz tog prirodnog spoja mora da zadovoljava ovaj definisan uslov.

$$Con(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)(t)$$

Pošto ovo ograničenje kontrolišemo u procesu ažuriranja podataka kritične operacije su insert i update ali u bilo kojoj od ovih relacija.

Sve ovo od gore prikazano u tabelarnom obliku:

TipO	ExTupleCon	prošireno ograničenje torke
	<div> <div>Role₁</div> <div>Δ</div> <div>Mult₁</div> <div>*</div> <div>AtStr₁</div> <div>set</div> <div>AtMult₁</div> <div>*</div> </div>	
T(t)	ins	NoAction, SetNull, SetDefault, <<UserDef>>
	upd	NoAction, SetNull, SetDefault, <<UserDef>>
TOd	*	višerelaciono ograničenje
TOi	t	ograničenje torke
TFz		$\tau_{ex}(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m) = Con(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)$
TPI		$\tau_{ex}(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)(t) = Con(N_1 \triangleright \triangleleft \dots \triangleright \triangleleft N_m)(t)$

Predavanje 2.

Ograničenje ključa

Ovo ograničenje je veoma poznato iz baza 1 i 2, ovo bi zapravo trebalo da bude rekapitulacija.

– KeyCon(T(t), 1, r, TFz, TPI)

- Tip logičke strukture
 - $T(t) = \{(\Delta, 1, set, *, \{(ins, fact_i), (upd, fact_u)\})\}$
 - » nebitna uloga, koristi se za tačno jednu šemu relacije, nad skupom više obeležja
 - » $fact_i, fact_u = \{NoAction, <<UserDef>>\}$

Tip logičke strukture obeležja uključuje jednu jedinu ulogu (delta u zagradi kod T(t)), ta uloga je nebitna, dozvoljava jednu šemu relacija sa navedenom ulogom i koristimo skup obeležja te šeme relacija da bi smo iskazali šta je to ograničenje ključa. Kritične operacije su insert i update. Akcije koje se preduzimaju prilikom kršenja su NoAction i neke korisničke definisane akcije.

- Tip definisanosti TOd - *jednorelaciono ograničenje*
- Tip interpretacije TOi - *relaciono ograničenje*
- Šablon za tip zapisa TFz
 - $TFz = Key(N, X), X \subseteq R$
 - » X je ključ šeme relacije N, sa skupom obeležja R
 - » $(\forall A \in X)(Null(A) = \perp)$ – zabrana nula vrednosti za X

Jednorelaciono je ograničenje zato što se definiše nad jednom šemom relacije, takođe je i relaciono ograničenje zato što se proverava nad celo šemom relacije.

U formuli za zapisivanje zaključujemo samo da je X ključ šeme relacije N. Takva formulacija sa sobom obavezno donosi da su nula vrednosti za X zabranjene.

Pitanje: Zašto delete nije kritična operacija za ograničenje ključa ? Zato što je ključ unique vrednost, kad brišemo znamo da ne možemo ništa da narušimo, kad radimo insert i update možemo narušiti ovo ograničenje.

Zanimljivost ovde je što se minimalisot uopšte ne kontroliše, prepuštamo to na verovanje da je projektant odradio posao kako treba. Tu zaključujemo da se deo problema rešava na projektantskom nivou a deo se prepušta sistemu da rešava zadavanjem ograničenja.

- Pravilo za interpretaciju TPI
 - prema definiciji u RMP
 - interpretira se za celu relaciju r(N)
 - 1^o: $(\forall u, v \in r(N))(u[X] = v[X] \Rightarrow u = v)$
 - 2^o: $(\forall X' \subset X)(\neg 1^o(X'))$

- Pravilo za interpretaciju kaže, za svake dve torke ako su one jednake na X-u onda su one jednake generalno. Odnosno ako idemo obrat po kontrapoziciji ove implikacije rećemo za svake dve torke iz relacije mora da važi ako su svake dve torke različite makar moraju biti različite na X-u, na ovaj način i SBP proverava.

2. Drugo svojstvo postoji kao projektantsko svojstvo, ono kaže da svakom pravom podskupu od X ne sme ostati zadovoljeno svojstvo 1 (u bilo kojoj relaciji). To znači da će uvek postojati neka relacija koja će narušiti svojstvo 1, ne mora značiti da će svaka narušiti.

Sve ovo od gore prikazano u tabelarnom obliku:

TipO	KeyCon	ograničenje ključa
T(t)	Role _i	Δ Mult _i 1 AtStr _i set AtMult _i *
	ins	NoAction, <<UserDef>>
	upd	NoAction, <<UserDef>>
TOd	1	jednorelaciono ograničenje
TOi	r	relaciono ograničenje
TFz	Key(N, X), $X \subseteq R$	
TPi	$1^0: (\forall u, v \in r(N))(u[X] = v[X] \Rightarrow u = v) \wedge$ $2^0: (\forall X' \subset X)(\neg 1^0(X'))$	

Pitanje: Zašto su SetDefault i SetNull ovde otpali? Null logično zbog ključa otpada, a Default zbog Uniquea, zbog prvog uslova jedinstvenosti torke, Default je ponovljena vrednost i zato to ovde nema smisla.

Ograničenje jedinstvenosti

– UniqueCon(T(t), 1, r, TFz, TPi)

- Tip logičke strukture
 - $T(t) = \{(\Delta, 1, set, *, \{(ins, sact_i), (upd, sact_{uj})\})\}$
 - » nebitna uloga, koristi se za tačno jednu šemu relacije, nad skupom više obeležja
 - » $sact_i, sact_{uj} = \{NoAction, SetNull, <<UserDef>>\}$
- Tip definisanosti TOd - jednorelaciono ograničenje
- Tip interpretacije TOi - relaciono ograničenje
- Šablon za tip zapisa TFz
 - $TFz = Unique(N, X), X \subseteq R$
 - » X predstavlja ograničenje jedinstvenosti vrednosti obeležja šeme relacije N, sa skupom obeležja R

Tip logičke strukture se definiše sada isto kao kod ograničenja ključa. Medjutim, sada kod akcija imamo i SetNull. Ovo ograničenje je jednorelaciono i relaciono kao i ograničenje ključa, zapisujemo ga kao Unique(N,X).

- Pravilo za interpretaciju TPi
 - prema definiciji u RMP
 - interpretira se za celu relaciju r(N)
 - » $1^0: (\forall u, v \in r(N))((u[X] \neq \omega \wedge v[X] \neq \omega) \Rightarrow (u[X] = v[X] \Rightarrow u = v))$
 - » $2^0: (\forall X' \subset X)(\neg 1^0(X'))$

Kako ga proveravamo? Znači za svake dve torke ukoliko važi da na X-u obe te torke imaju ne nula vrednosti, onda insistiramo na tome da ako su dve torke jednake na X-u onda su generalno jednake, ili ako su različite onda moraju biti različite makar na tom X-u. I takođe se insistira na minimalnosti na projektantskom kriterijumu.

TipO	UniqueCon	ograničenje jedinstvenosti
T(t)	Role _i	Δ Mult _i 1 AtStr _i set AtMult _i *
	ins	NoAction, SetNull, <<UserDef>>
	upd	NoAction, SetNull, <<UserDef>>
TOd	1	jednorelaciono ograničenje
TOi	r	relaciono ograničenje
TFz	Unique(N, X), $X \subseteq R$	
TPi	$1^0: (\forall u, v \in r(N))((u[X] \neq \omega \wedge v[X] \neq \omega) \Rightarrow (u[X] = v[X] \Rightarrow u = v)) \wedge$ $2^0: (\forall X' \subset X)(\neg 1^0(X'))$	

Pitanje: Šta je razlika u odnosu na ograničenje ključa? Imamo dozvoljenu Null vrednost. I ograničenje jedinstvenosti je slabije od ograničenja ključa.

Zavisnost sadržavanja

$$- \ln \text{Con}(T(t), 2, m, TFz, TPi)$$

- Tip logičke strukture
 - $T(t) = \{(referencing, 1, array, *, \{(ins, sact_i^1), (upd, sact_u^1)\}), (referenced, 1, array, *, \{(del, sact_i^2), (upd, sact_u^2)\})\}$
 - » svaka od uloga se koristi za tačno jednu šemu relacije i niz od više njenih obeležja
 - » $sact_i^1, sact_u^1 = \{NoAction, SetNull, SetDefault, <<UserDef>>\}$
 - » $sact_i^2, sact_u^2 = \{NoAction, Cascade, SetNull, SetDefault, <<UserDef>>\}$
- Tip definisanosti TOd - *višerelaciono ograničenje*
- Tip interpretacije TOi - *međurelaciono ograničenje*
- Sablon za tip zapisa TFz
 - $TFz = N_i[X] \subseteq N_j[Y]$
 - » X predstavlja niz obeležja referencirajuće šeme relacije N_i
 - » Y predstavlja domenski kompatibilni niz obeležja referencirane šeme relacije N_j
- Pravilo za interpretaciju TPi
 - prema definiciji u RMP
 - interpretira se nad celokupnim relacijama $r(N_i)$ i $r(N_j)$
 - » $\pi_X(r(N_i)) \subseteq \pi_Y(r(N_j))$

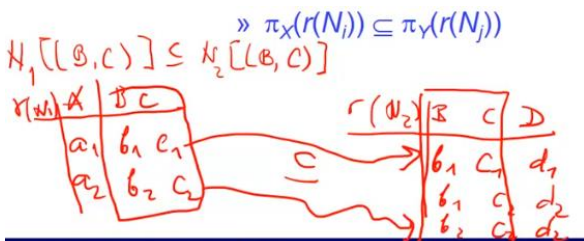
Krenucemo od šablona za zapis, kakvi su to X i Y , pa oni su nizovi jednake dužine, da bi bili domenski kompatibilni mora još da bude redosled obeležja u nizu X i redosled obeležja u nizu Y isti. Domenski kompatibilna obeležja moraju biti na istim pozicijama u nizu.

Sada tip logičke strukture obuhvata dve uloge, jedna uloga ce biti uloga referencirajuće seme relacije, a druga uloga je uloga referencirane seme relacije. I u jednoj i u drugoj može se u zavisnosti sadržavanja javiti samo jedna relacija. Na dalje koriste se nizovi više obeležja i za referenciranu i za referencirajuću (ono sto gore pise array, *). Nigde nismo definisali da moraju da budu iste dužine i da budu domenski kopatibilna obeležja na istim pozicijama u nizu, da bi ovo definisali morali bi da budemo mnogo pedantniji.

Ovo je višerelaciono i međurelaciono ograničenje, međurelaciono je zato što je za njegovu kontrolu neophodno da imamo dve relacije, jednu koja odgovara referencirajućoj a druga koja odgovara referenciranoj semi relacije.

Intencija formule kod interpretacije jeste da projekcija na skup obeležja X referencirajuće seme relacije mora biti podskup od projekcije na skup obeležja Y od referencirane seme relacije.

TipO	InCon	zavisnost sadržavanja						
T(t)	Role ₁	referencing	Mult ₁	1	AtStr ₁	array	AtMult ₁	*
	ins	NoAction, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, SetNull, SetDefault, <<UserDef>>						
	Role ₂	referenced	Mult _m	1	AtStr _m	array	AtMult _m	*
	del	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
TOd	2	višerelaciono ograničenje ("dvorelaciono")						
TOi	m	međurelaciono ograničenje						
TFz	$N_i[X] \subseteq N_j[Y]$							
TPi	$\pi_X(r(N_i)) \subseteq \pi_X(r(N_j))$							



Proširena zavisnost sadržavanja

– $ExInCon(T(t), *, m, TFz, TPi)$

- Tip logičke strukture
 - $T(t) = \{(referencing, *, array, *, \{(ins, sact_i^1), (upd, sact_u^1)\}), (referenced, *, array, *, \{(del, sact_i^2), (upd, sact_u^2)\})\}$
 - » svaka od uloga se koristi za više šema relacija i niz od više njihovih obeležja
 - » $sact_i^1, sact_u^1 = \{NoAction, SetNull, SetDefault, <<UserDef>>\}$
 - » $sact_i^2, sact_u^2 = \{NoAction, Cascade, SetNull, SetDefault, <<UserDef>>\}$
- Tip definisanosti TOd - višerelaciono ograničenje
- Tip interpretacije TOi - međurelaciono ograničenje

Ono što odmah primećujemo kao razliku između obične i proširene zavisnosti sadržavanja je to što u proširenoj imamo više referenciranih i referencirajućih šema. I dalje je isto višerelaciono i međurelaciono ograničenje.

- Šablon za tip zapisa TFz
 - $TFz = (\triangleright \triangleleft N_{k=i_1}^{im})[X] \subseteq (\triangleright \triangleleft N_{l=j_1}^{jm})[Y]$
 - » X predstavlja niz obeležja referencirajućih šema relacija
 - » Y predstavlja domenski kompatibilni niz obeležja referenciranih šema relacija
- Pravilo za interpretaciju TPi
 - prema definiciji u RMP
 - interpretira se nad spojevima relacija
 - $\triangleright \triangleleft_{k=i_1}^{im}(r(N_k))$ i $\triangleright \triangleleft_{l=j_1}^{jm}(r(N_l))$
 - » $\pi_X(\triangleright \triangleleft_{k=i_1}^{im}(r(N_k))) \subseteq \pi_Y(\triangleright \triangleleft_{l=j_1}^{jm}(r(N_l)))$

Šta kaže pravilo za interpretaciju? Prvo uradimo spoj svih relacija nad šemama sa leve strane, pa uradimo spoj svih relacija nad šemama sa desne strane, pa napravimo projekciju te relacije koju smo dobili kao rezultat spajanja nad X , pa napravimo projekciju te relacije koju smo dobili kao rezultat spajanja na Y i proverimo da li je Pi od X podskup Pi od Y .

TipO	ExInCon		proširena zavisnost sadržavanja					
T(t)	Role ₁	referencing	Mult ₁	*	AtStr ₁	array	AtMult ₁	*
	ins	NoAction, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, SetNull, SetDefault, <<UserDef>>						
	Role ₂	referenced	Mult _m	*	AtStr _m	array	AtMult _m	*
	del	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
TOd	*	višerelaciono ograničenje						
TOi	m	međurelaciono ograničenje						
TFz	$(\triangleright \triangleleft N_{k=i_1}^{im})[X] \subseteq (\triangleright \triangleleft N_{l=j_1}^{jm})[Y]$							
TPi	$\pi_X(\triangleright \triangleleft_{k=i_1}^{im}(r(N_k))) \subseteq \pi_Y(\triangleright \triangleleft_{l=j_1}^{jm}(r(N_l)))$							

Pitanje: Gde smo ovako nešto sreli ? Kod dekompozicije kod algoritama kad smo dekomponovali na šeme morali smo da zadržimo redosled.

Selektivna zavisnost sadržavanja

Njih smo imali kada smo radili prevođenje iz ER šeme u relacioni model podataka.

– SellInCon($T(t)$, 2, m , TFz , TPi)

- Tip logičke strukture
 - $T(t) = \{(referencing, 1, array, *, \{(ins, sact_i^1), (upd, sact_u^1)\}), (referenced, 1, array, *, \{(del, sact_i^2), (upd, sact_u^2)\})\}$
 - » svaka od uloga se koristi za tačno jednu šemu relacije i niz od više njenih obeležja
 - » $sact_i^1, sact_u^1 = \{NoAction, SetNull, SetDefault, <<UserDef>>\}$
 - » $sact_i^2, sact_u^2 = \{NoAction, Cascade, SetNull, SetDefault, <<UserDef>>\}$
- Tip definisanosti TOd - višerelaciono ograničenje
- Tip interpretacije TOi - međurelaciono ograničenje

Ista je situacija i definicija u logičkoj strukturi podataka kao kod obične zavisnosti sadržavanja. Takođe, imamo referenciranu i referencirajuću šemu relacije, iste kritične operacije i iste akcije prilikom narušavanja ograničenja

- Šablon za tip zapisa TFz
 - $TFz = \sigma_{F_i}(N_i[X]) \subseteq \sigma_{F_j}(N_j[Y])$
 - » X predstavlja niz obeležja referencirajuće šeme relacije N_i
 - » Y predstavlja domenski kompatibilni niz obeležja referencirane šeme relacije N_j
- Pravilo za interpretaciju TPi
 - prema definiciji u RMP
 - interpretira se nad selekcijama relacija $\sigma_{F_i}(r(N_i))$ i $\sigma_{F_j}(r(N_j))$, redom, po formulama F_i i F_j
 - » $\pi_X(\sigma_{F_i}(r(N_i))) \subseteq \pi_Y(\sigma_{F_j}(r(N_j)))$

Kakvu intenciju imamo sa ovakvim pravilom za interpretaciju? Ne treba sve torke iz relacije sa leve strane da zadovolje referenciranje već samo neke koje zadovoljavaju ovu formulu F_i . Takođe, kada referenciraju neke torke iz ove relacije N_j to ne mogu da budu bilo koje torke iz N_j nego samo selektovane torke iz F_j (Kada je objasnjavao objasnjavao je za šablon tipa zapisa TFz).

Pitanje: Gde smo imali pojavu selektivnih zavisnosti sadržavanja odnosno selektivnih ograničenja referencijalnog integriteta? Npr možemo imati spisak studenata koji treba da diplomiraju, tu može da bude selektivna zavisnost sadržavanja prema spisku svih studenata, ako želimo da napravimo prijavu diplomskog rada onda ona može referencirati ne bilo kog studenta nego studenta koji je ispunio sva prava za odbranu diplomskog. (To je ovo npr σ_{F_j})

Tabelarni prikaz svega ovoga, razlika je što sad insistiramo na upotrebi selekcionih formule, putem koje mi kažemo “samo neke torke sa leve i samo neke torke sa desne strane”

TipO	SellInCon	selektivna zavisnost sadržavanja
$T(t)$	$Role_1$ referencing $Mult_1$ 1 $AtStr_1$ array $AtMult_1$ *	
	ins	NoAction, SetNull, SetDefault, <<UserDef>>
	upd	NoAction, SetNull, SetDefault, <<UserDef>>
	$Role_2$ referenced $Mult_m$ 1 $AtStr_m$ array $AtMult_m$ *	
	del	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>
	upd	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>
TOd	2	višerelaciono ograničenje ("dvorelaciono")
TOi	m	međurelaciono ograničenje
TFz	$\sigma_{F_i}(N_i[X]) \subseteq \sigma_{F_j}(N_j[Y])$	
TPi	$\pi_X(\sigma_{F_i}(r(N_i))) \subseteq \pi_Y(\sigma_{F_j}(r(N_j)))$	

Selektivna proširena zavisnost sadržavanja

– $SelExInCon(T(t), *, m, TFz, TPi)$

- Tip logičke strukture
 - $T(t) = \{(\text{referencing}, *, \text{array}, *, \{(ins, \text{sact}_1^1), (upd, \text{sact}_u^1)\}), (\text{referenced}, *, \text{array}, *, \{(del, \text{sact}_1^2), (upd, \text{sact}_u^2)\})\}$
 - svaka od uloga se koristi za više šema relacija i niz od više njihovih obeležja
 - $\text{sact}_1^1, \text{sact}_u^1 = \{\text{NoAction}, \text{SetNull}, \text{SetDefault}, \ll UserDef \gg\}$
 - $\text{sact}_1^2, \text{sact}_u^2 = \{\text{NoAction}, \text{Cascade}, \text{SetNull}, \text{SetDefault}, \ll UserDef \gg\}$
- Tip definisanosti TOd - višerelaciono ograničenje
- Tip interpretacije TOi - međurelaciono ograničenje

Sada imamo sve isto kao i kod selektivne zavisnosti sadržavanja ali imamo i proširene zavisnosti sadržavanja pa ćemo dozvoliti više šema relacija sa ulogom referencirajućom šemom relacije i više uloga sa referenciranom šemom relacija, sve ostalo je isti ambijent kao što smo malopre imali.

- Šablon za tip zapisa TFz
 - $TFz = \sigma_{F_i}(\triangleright \triangleleft (N_{k=i_1}^{im})[X] \subseteq \sigma_{F_j}(\triangleright \triangleleft (N_{l=j_1}^{jm})[Y])$
 - X predstavlja niz obeležja referencirajućih šema relacija
 - Y predstavlja domenski kompatibilni niz obeležja referenciranih šema relacija
- Pravilo za interpretaciju TPi
 - prema definiciji u RMP
 - interpretira se nad selekcijama spojeva relacija, redom, po formulama F_i i F_j

$$\sigma_{F_i}(\triangleright \triangleleft_{k=i_1}^{im}(r(N_k))) \text{ i } \sigma_{F_j}(\triangleright \triangleleft_{l=j_1}^{jm}(r(N_l)))$$

$$\pi_X(\sigma_{F_i}(\triangleright \triangleleft_{k=i_1}^{im}(r(N_k)))) \subseteq \pi_Y(\sigma_{F_j}(\triangleright \triangleleft_{l=j_1}^{jm}(r(N_l))))$$

Formula za zapisivanje je sada najkompleksnija moguća, idemo na spojeve relacija nad šemama sa desne strane pa pravimo selekciju nad tim torkama iz tog spoja, pa kažemo u načelu nad svim onim manama nad kojima smo diskutovali kada je ova formula u pitanju, kažemo da je potrebno da napravimo spoj relacija sa leve strane na osnovu zadatih šema relacija, da napravimo iz tog spoja selekciju sa onih torki koje zadovoljavaju određenu formulu (F_j) pa onda uzmemo projekciju na skup obeležja X od ovog niza obeležja X

$$\pi_X(\sigma_{F_i}(\triangleright \triangleleft_{k=i_1}^{im}(r(N_k))))$$

Pa onda isto tako spojimo sve relacije nad onim relacijama koje su nabrojane sa desne strane. Napravimo selekciju po nekoj formuli (F_j), napravimo projekciju na skup obeležja Y , i ovo što je sa leve strane mora da bude podskup onoga sa desne strane.

TipO	SelExInCon		selektivna proširena zavisnost sadrž.					
T(t)	Role ₁	referencing	Mult ₁	*	AtStr ₁	array	AtMult ₁	*
	ins	NoAction, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, SetNull, SetDefault, <<UserDef>>						
	Role ₂	referenced	Mult _m	*	AtStr _m	array	AtMult _m	*
	del	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
TOd	*	višerelaciono ograničenje						
TOi	m	međurelaciono ograničenje						
TFz	$\sigma_{F_i}(\triangleright \triangleleft (N_{k=i_1}^{im})) [X] \subseteq \sigma_{F_j}(\triangleright \triangleleft (N_{l=j_1}^{jm})) [Y]$							
TPi	$\pi_X(\sigma_{F_i}(\triangleright \triangleleft_{k=i_1}^{im}(r(N_k)))) \subseteq \pi_Y(\sigma_{F_j}(\triangleright \triangleleft_{l=j_1}^{jm}(r(N_l))))$							

Zavisnosti sadržavanja mogu biti i proširene i selektivne i proširene i selektivne u isto vreme.

Pitanje: Šta označava termin „bilo koja relacija“ kada kažemo kod tipova ograničenja kada kažemo da se njihova interpretacija odnosno validacija odnosi na bilo koju relaciju. To je relacija iz baze podataka ili iz pogleda.

Specifikacija ograničenja

Sada prelazimo na specifikacije konkretnih ograničenja na nivou sistema za upravljanje bazom podataka.

Idemo na to da što formalnije moguće iskažemo specifikacije konkretnih ograničenja. Obezbeđuje se specifikacija jednog konkretnog ograničenja datog tipa i mora se poštovati opšta specifikacija tog tipa ograničenja $TipO(T(t), TOd, Toi, TFz, TPi)$ kao i specifikacija tog konkretnog izabranog tipa ograničenja.

Specifikacija ograničenja šeme BP

OgrNaz(OgrTip, OgrF, T(o))

- *OgrNaz* - naziv ograničenja šeme BP
- *OgrTip* - oznaka tipa ograničenja
- *OgrF* - formula – zapis ograničenja
- *T(o)* - logička struktura obeležja

To je struktura koja ima naziv ograničenja šeme BP. Kada pravimo konkretno ograničenje mi se moramo referencirati na tip ograničenja za koji pravimo konkretno ograničenje (*OgrTip*). Takođe, moramo da definišemo konkretnu formulu u skladu sa parametrizovanim šablonom formule, za tip ograničenja koje je prethodno uvedeno. Pored toga, moramo zadati i konkretnu logičku strukturu obeležja.

- *OgrNaz* - naziv ograničenja šeme BP
 - niz znakova
 - proizvoljno odabrani naziv ograničenja
 - ukazuje na semantiku konkretnog ograničenja
 - ne postoje posebni uslovi za zadavanje naziva ograničenja
 - » u praksi, moraju se poštovati konvencije imenovanja objekata na nivou izabranog SUBP

- *OgrTip* - oznaka tipa ograničenja
 - obavezna komponenta specifikacije
 - preuzeta oznaka tipa ograničenja iz skupa svih oznaka tipova
 - $OgrTip \in SetTipO$
 - $SetTipO = \{$
DomCon, AttValCon, TupleCon, ExTupleCon, KeyCon,
UniqueCon, InCon, ExInCon, SelInCon, SelExInCon,
RefInCon, ExRefInCon, SelRefInCon, SelExRefInCon,
InvRfCon, ExInvRfCon, SelInvRfCon, SelExInvRfCon,
<<UserDefCon>>
 $\}$
 - ograničenje nasleđuje sve osobine specificiranog tipa

- OgrF

- formula – zapis ograničenja

- obavezna komponenta specifikacije
- niz znakova
- predstavlja formulu za zapis konkretnog ograničenja
 - napisanu pomoću sintakse, zadate putem pravila TFz
 - izabranog tipa ograničenja $OgrTip = TipO(T(t), TOd, TOi, TFz, TPi)$
- praktično, to je konkretizovana formula izvedena iz parametrizovanog šablona TFz

- T(o)

- logička struktura obeležja

- obavezna komponenta specifikacije
- u kontekstu je date specifikacije tipa logičke strukture $T(t)$
 - izabranog tipa ograničenja
 - $OgrTip = TipO(T(t), TOd, TOi, TFz, TPi)$
- za $T(t) = \emptyset$, biće $T(o) = \emptyset$
- za $T(t) \neq \emptyset$, strukturirana putem formule
 - $T(o) = \{(N_i, \rho_i, \{(op_i^{ij}, At_i, act_i^{ij}) \mid i_j \geq 1\}), \dots, (N_m, \rho_m, \{(op_m^{im}, At_m, act_m^{im}) \mid i_m \geq 1\})\}$
 - Definicija strukture elementa
 - $(N_j, \rho_j, \{(op_j^{ij}, At_j, act_j^{ij}) \mid i_j \geq 1\})$



- Definicija strukture elementa

$$(N_j, \rho_j, \{(op_j^{ij}, At_j, act_j^{ij}) \mid i_j \geq 1\})$$

- N_j
 - » oznaka šeme relacije, obuhvaćene ograničenjem $OgrF$
- $\rho_j = Role_i$
 - » uloga šeme relacije u ograničenju
 - » $Role_i$ mora postojati u kontekstnoj strukturi $T(t)$
- $(op_j^{ij}, At_j, act_j^{ij})$
 - » specifikacija kritične operacije ograničenja s akcijom
 - » op_j^{ij} – kritična operacija, mora biti navedena u strukturi za dati $Role_i$
 - » At_j – skup obeležja šeme relacije na koji se odnosi akcija, ili *, kao oznaka za sve attribute
 - » act_j^{ij} – odabrana akcija iz odgovarajućeg skupa mogućih akcija $sact_j^{ij}$ za odabrani op_j^{ij}

OgrNaz	naziv ograničenja				
OgrTip	oznaka tipa ogranič.	duži naziv tipa ograničenja			
OgrF	formula - zapis ograničenja				
T(o)	N_1	naziv šeme relacije		ρ_1	uloga
	op_1^1	At_1^1 (skup obeležja, ili *)		act_1^1 (izabrana akcija)	
	op_1^2	At_1^2 (skup obeležja, ili *)		act_1^2 (izabrana akcija)	
	...				
	N_m	naziv šeme relacije		ρ_m	uloga
	op_m^1	At_m^1 (skup obeležja, ili *)		act_m^1 (izabrana akcija)	
	op_m^2	At_m^2 (skup obeležja, ili *)		act_m^2 (izabrana akcija)	

• Primer – tip ograničenja



TipO	RefInCon		ograničenje referencijalnog integriteta					
T(t)	Role ₁	referencing	Mult ₁	1	AtStr ₁	array	AtMult ₁	*
	ins	NoAction, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, SetNull, SetDefault, <<UserDef>>						
	Role ₂	referenced	Mult _m	1	AtStr _m	array	AtMult _m	*
	del	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
	upd	NoAction, Cascade, SetNull, SetDefault, <<UserDef>>						
TOd	2	višerelaciono ograničenje ("dvorelaciono")						
TOi	m	međurelaciono ograničenje						
TFz	$N_i[X] \subseteq N_i[Y, Key(N_j, Y)]$							
TPi	$\pi_X(r(N_i)) \subseteq \pi_Y(r(N_j))$							

U ovom primeru imamo klasično ograničenje referencijalnog integriteta. U tipu ograničenja referencijalnog integriteta mi smo izdvojili dve uloge, jedna je uloga referencirajuće druga je uloga referencirane. Dozvolili smo jednu ulogu i kod jedne i kod druge (Mult u tabeli) i rekli smo da će nadalje biti korišćen niz od više atributa i sa leve i sa desne strane. Kritične operacije su za referencirajucu insert i update a za referenciranu delete i update, sa akcijama koje su nabrojane u tabeli. Višerelaciono i međurelaciono ograničenje i zapisuje se po formuli koja je data u tabeli, uz to još kada je u pitanju ref integritet kažemo da taj Y kao skup obeležja mora biti ključ šeme relacije N_j . I ono

što još kažemo da za svaku torku sa referencirajuće strane mora da postoji tačno jedna torka iz relacije sa referencirane strane takva da je ona tamo na X jednaka onoj tamo na Y (pslednja rečenica je za TPi).

• Primer – konkretno ograničenje datog tipa



<i>OgrNaz</i>	<i>Fakt_PoslPart_RI</i>			
<i>OgrTip</i>	<i>RefInCon</i>	ograničenje referencijalnog integr.		
<i>OgrF</i>	<i>Faktura[IdPP] ⊆ PoslPart[IdPP]</i>			
<i>T(o)</i>	<i>N₁</i>	<i>Faktura</i>	<i>ρ₁</i>	<i>referencing</i>
	<i>ins</i>	*	<i>NoAction</i>	
	<i>upd</i>	{ <i>IdPP</i> }	<i>NoAction</i>	
	<i>N₂</i>	<i>PoslPart</i>	<i>ρ_m</i>	<i>referenced</i>
	<i>del</i>	*	<i>Cascade</i>	
	<i>upd</i>	{ <i>IdPP</i> }	<i>NoAction</i>	

Po našem internom kompanijskom standardu zadajemo naziv (Fakt_PoslPart_RI) što u prevodu znaci prvo navodimo alias od referencirajuće pa onda od referencirane šeme relacije i navodimo RI kao sufiks koji označava referencijal integrity. Onda kažemo da je naše ograničenje tipa ograničenja referencijalnog integriteta (RefInCon). Zapisujemo ga u formi Faktura od IdPP podskup od Poslovni partner od IdPP, znamo da IdPP predstavlja ključ šeme relacije PoslPart. U ovom ograničenju participiraju dve šeme relacija, jedna referencirajuća N1, kritična operacija insert novih torki u relaciju Faktura, i kaže ukoliko prilikom

upisa novih torki u relaciju Faktura narušiš referenciranje prema Poslovnom partneru biće primenjen NoAction, znači pokušaj upisa fakture za koji ne postoji IdPP u poslovnom partneru biće sprečen. Druga kritična operacija je operacija Update stranog ključa koji je ovde IdPP, dakle na pokušaj izmene vrednosti IdPP u Fakturi biće izvršen NoAction što znači prekid takve operacije. U slučaju poslovnih partnera, mi ćemo reći da je PoslPart referencirana šema relacije, kritične operacije su delete i update, znači na pokušaj brisanja torke iz PoslPart kaskadno primeni brisanje i na sve torke u relaciji Faktura, a ukoliko neko u PoslPart pokuša izmenu šifre PoslPart odnosno IdPP-a i ukoliko postoji barem jedna faktura koja postoji povezana sa tip IdPP u tabeli Faktura, tada će se obustaviti operacija.