

Projektovanje baza podataka

Logičko modeliranje

19. maj 2026.

Sadržaj

- 1 Logičko modeliranje
- 2 Dijagrami tabela
- 3 Prevođenje konceptualnog modela u logički

Napomena

Slajdovi su nastali obradom slajdova prof. dr Saše Malkova za predmet Projektovanje baza podataka

Sadržaj

- 1 Logičko modeliranje
- 2 Dijagrami tabela
- 3 Prevođenje konceptualnog modela u logički

Logičko modeliranje

- Konceptualni model je fokusiran na semantiku i na odnose
 - u velikoj meri ne zavisi od vrste baza podataka koja će biti upotrebljavana u implementaciji
- Logički model se spušta bliže izabranom implementacionom modelu
 - uzima u obzir model podataka koji će se upotrebljavati u implementaciji
- Apstraktni koncepti konceptualnog modela se prevode u konkretne logičke modele

Uloga logičkog modela

- Logički model uzima u obzir
 - sve specifičnosti izabranog ciljnog modela podataka
 - dodatne nefunkcionalne zahteve, koji su u fazi konceptualnog projektovanja možda mogli da ostanu zanemareni
- Vodi se računa o
 - fleksibilnosti
 - proširivosti
 - ali ne na uštrb jasnoće i čistoće modela
- Logički model tačno određuje koji se podaci čuvaju i kako će se njima rukovati

Preskakanje logičkog modela

- Projektanti su često u iskušenju da posle konceptualnog modela odmah prave fizički model
 - Ili se preskače logički model, ili se zapravo preskače konceptualni i odmah pravi logički
 - Vrlo često je već pri pravljenju konceptualnog modela poznato koja implementacija SUBP će se koristiti, pa može da izgleda da takvo preskakanje ne predstavlja problem
- Takav pristup nosi rizik da se izgube neke od poželjnih karakteristika baze podataka:
 - kompletnost
 - integritet
 - fleksibilnost
 - efikasnost
 - upotrebljivost

Preskakanje logičkog modela

- Manje baze podataka mogu da se dobro isprojektuju i bez logičkog (i/ili konceptualnog) modela, ali kod većih baza takav pristup obično vodi problemima i pri projektovanju i pri održavanju
- Za odgovore na neka pitanja je vrlo često potrebno imati sva tri modela
- Na primer, pri promenama strukture baze podataka često se pitamo zašto je nešto u fizičkom modelu napravljeno baš tako a ne drugačije? Da li je to prirodan model ili optimizacija?
- Odgovor se često može dobiti samo analizom sva tri modela.

Kriterijumi za unapređivanje iteracija

- Kriterijumi za menjanje su željene karakteristike:
 - Da li model ispunjava funkcionalne zahteve?
 - Da li model ispunjava nefunkcionalne zahteve?
 - Da li model garantuje integritet podataka?
 - Da li model omogućava efikasan rad?
 - Da li je model u skladu sa ciljnim modelom podataka?
- Neki od kriterijuma su “nasleđeni” iz konceptualnog modela, ali neki su novi

Značajne odluke

- Logičko modeliranje može da ima za rezultat i neke značajne odluke, npr:
 - može da se ne pravi jedna nego više baza podataka
 - može da se struktura prilagodi očekivanim promenama (ili očekivanoj vrsti promena)
 - mogu da se donose i neke odluke u skladu sa specifičnostima izabrane implementacije SUBP

Relacioni model kao osnov logičkog modela

- U praksi se najčešće projektuju i implementiraju relacione baze podataka
 - Dalju obradu ove teme nastavljamo uz pretpostavku da je ciljni model upravo relacioni
 - Logički model ćemo da predstavljamo jezikom relacionog modela
 - Entiteti se predstavljaju kao relacije
 - Složeni odnosi se predstavljaju kao relacije
 - Jednostavni odnosi se predstavljaju stranim ključevima

Sadržaj

- 1 Logičko modeliranje
- 2 **Dijagrami tabela**
- 3 Prevođenje konceptualnog modela u logički

Dijagrami tabela

- Često se pogrešno nazivaju „ER-dijagrami“
- To *nisu* ER-dijagrami nego dijagrami tabela (relacija)
 - oznake i termini su obično tabele/kolone/ključevi
 - kardinalnosti se predstavljaju drugačije nego na ER-dijagramima
 - po uzoru na UML
 - obično se koriste samo 0, 1 i *
- Prilagođeni su logičkom i (posebno) fizičkom nivou relacionog modela

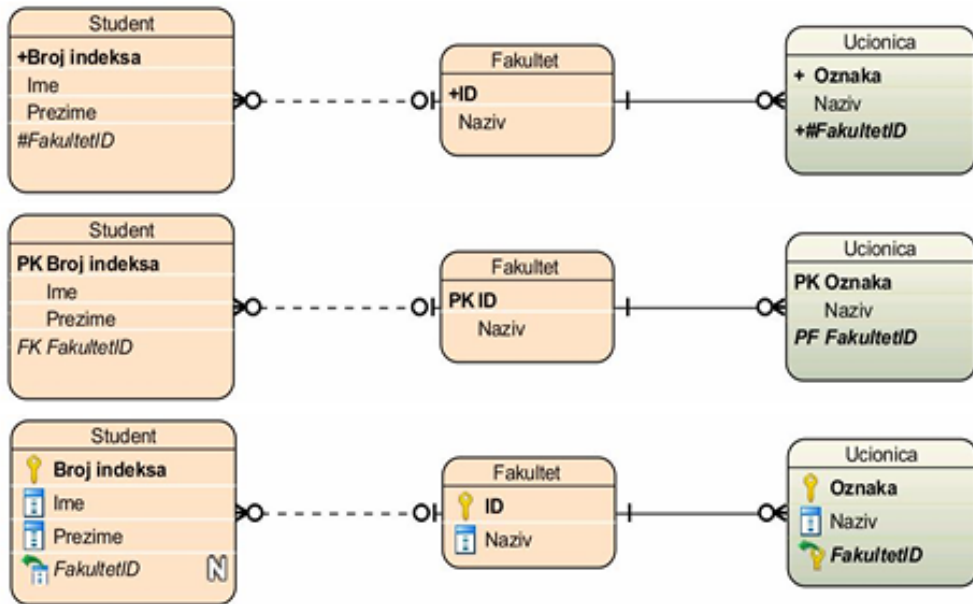
Tabele

- Tabele se predstavljaju pravougaonicima
 - obično zaobljenim
- Tabela je horizontalnom linijom podeljena na dva dela
 - gornji deo sadrži naziv tabele
 - donji deo sadrži opise kolona
 - u nekim varijantama pored opisa kolona mogu da se navedu i druge informacije

Tabele

- Kolone su posebno označene simbolima, slovnim oznakama ili crtežima
 - ako kolona pripada primarnom ključu
 - simbol +
 - oznaka PK
 - crtež ključa
 - ako kolona pripada stranom ključu
 - simbol #
 - oznaka FK (ili PF, ako je i primarni i strani, tzv. slabi entitet)
 - crtež ključa sa strelicom
 - obično i iskošen naziv
 - ako kolona može da sadrži nedefinisanu vrednost
 - simbol, crtež ili oznaka koji asociraju na slovo N

Tabele (primeri)



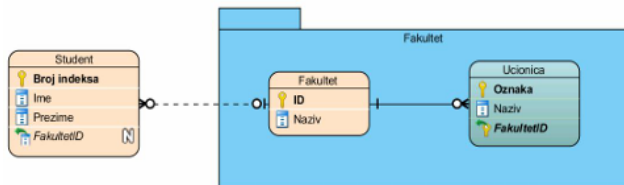
Vrste tabela

- Neki alati i autori razlikuju tabele prema boji
 - jednom bojom jaki entiteti
 - drugom bojom slabi entiteti
 - trećom bojom odnosi
 - može da se uvede i više različitih boja, prema kontekstu
- Neki alati i autori označavaju tabele različitim simbolima ili crtežima u desnom gornjem uglu
 - slično kao stereotipovi u UML-u
 - obično se koristi posebna oznaka za poglede
- Obe prakse su dobre i olakšavaju razumevanje modela

Grupisanje

- Za grupisanje tabela se koriste pravougaonici koji liče na fascikle
 - kao paketi u UML-u
- Mogu da se koriste i boje, ali to može da pravi problem u odnosu na označavanje vrsta tabela

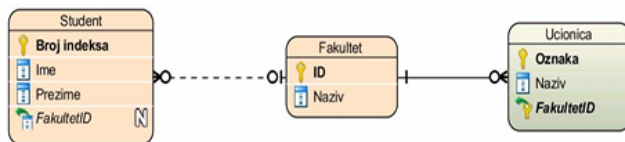
Grupisanje (primer)



Odnosi

- Osnovna vrsta „odnosa” su strani ključevi i odnosi više–više
- Obično se predstavljaju isprekidanim linijama
 - puna linija predstavlja odnos slabog entiteta sa matičnim
- Na kraju linije se predstavlja kardinalnost
 - u duhu UML-a
 - broj entiteta uz koji stoji oznaka, koji mogu da budu u odnosu sa jednim entitetom koji je na drugoj strani
 - grafičke oznake:
 - 0 – kružić
 - 1 – uspravna linija
 - * - linija se grana u tri kratke linije prema tabeli
 - 0-1, 0-*, 1-* – kombinacije prethodnih slučajeva

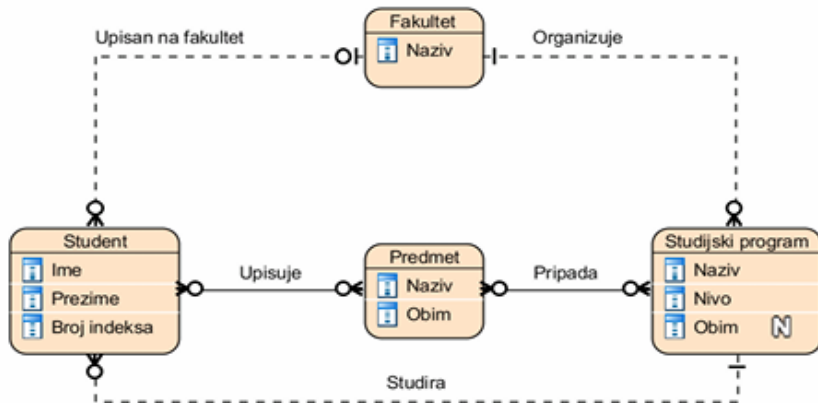
Odnosi (primer)



Nivo modela

- Osnovni oblik dijagrama najviše odgovara logičkom nivou
 - Na konceptualnom nivou se obično prilagođava
 - ne preporučuje se upotreba, osim za jednostavne baze podataka
 - Na logičkom nivou se obično koristi u osnovnom obliku ali može i da se prilagođava
 - Na fizičkom nivou se koristi u osnovnom obliku ali mogu da se dodaju i novi elementi za opisivanje elemenata fizičke implementacije

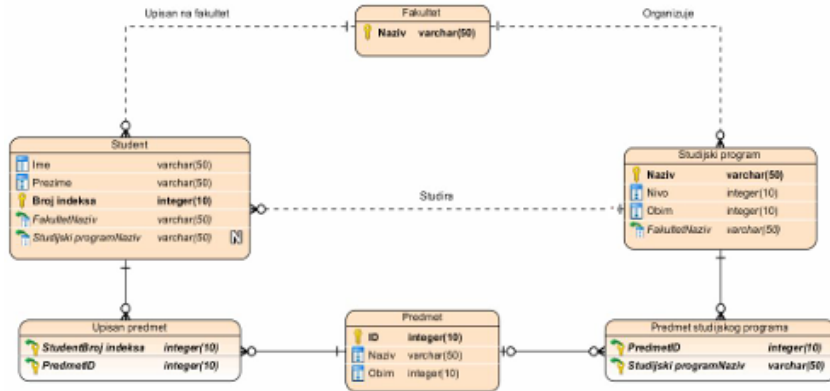
Konceptualni nivo (primer)



Logički nivo

- Dodaju se svi ostali podaci neophodni za preslikavanje u relacioni model:
 - kolone koje su surogat ključevi (ako su potrebne)
 - kolone vezanih tabela koji čine strane ključeve
 - oznake ključeva i vrsta ključeva
 - tipovi kolona
 - odnosi više–više se zamenjuju tabelama koje ih modeliraju
 - pravila integriteta

Logički nivo (primer)



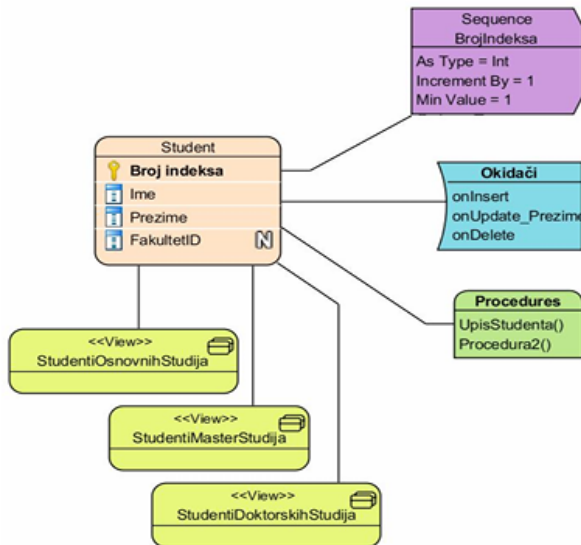
Fizički nivo

- Dodaju se elementi fizičke strukture:
 - pogledi
 - indeksi
 - okidači
 - prostori za tabele
 - ...

Dodatni elementi

- Dodatni elementi omogućavaju preciznije opisivanje
 - Koriste se primarno na fizičkom nivou, ali neki mogu i na logičkom
 - pogledi
 - obično pravougaonici označeni dodatnim simbolom i vezani linijom sa tabelama na osnovu kojih se definišu
 - ako dopuštaju menjanje podataka može i dodatno označavanje
 - okidači
 - uslovi integriteta
 - dodaju se na različite načine, obično u vidu komentara
 - prostori tabela i drugi elementi fizičkog modela
 - indeksi
 - ugrađene procedure
 - i drugo

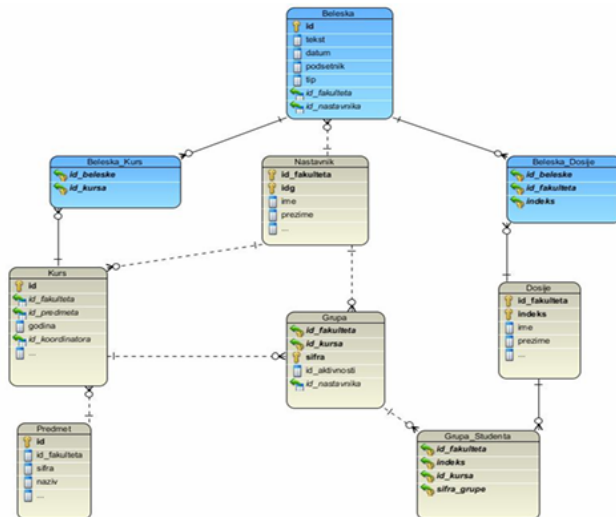
Dodatni elementi



Rezime

- Dijagram tabela je veoma sličan dijagramu klasa domena
 - Ima formalizovane dodatne elemente
 - okidači
 - procedure
 - ...
 - Ima slabije mogućnosti za opisivanje odnosa
 - sve se svodi na strane ključeve
 - Prilagođen logičkom nivou relacionog modela

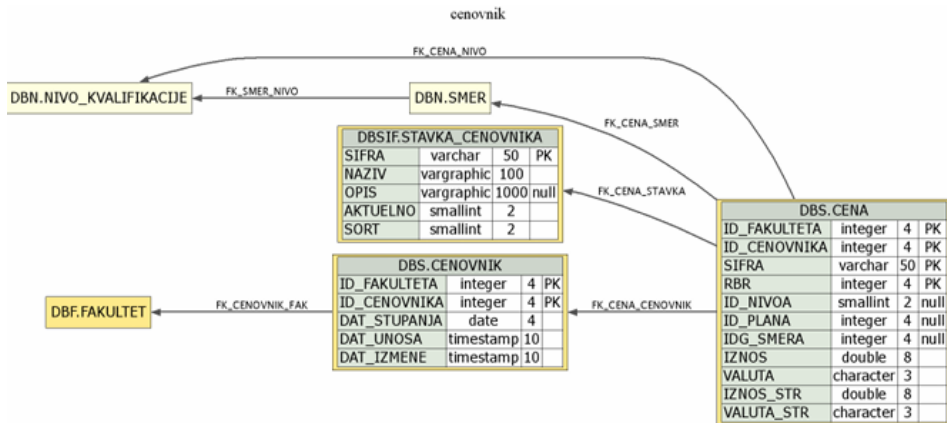
Primer 1 – Beleške u sistemu Studinfo



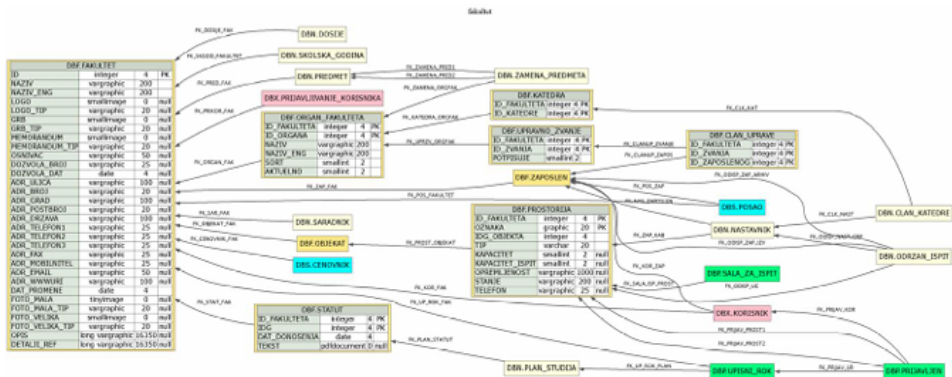
Modifikovani dijagrami

- Neki alati donose modifikacije
 - modifikacije često imaju za cilj približavanje OO modelu
 - na primer, strani ključevi kao strelice u smeru referisanja
 - modifikacije su načelno dopuštene
 - ako su promene intuitivno jasne
 - dok postoji doslednost u okviru projekta

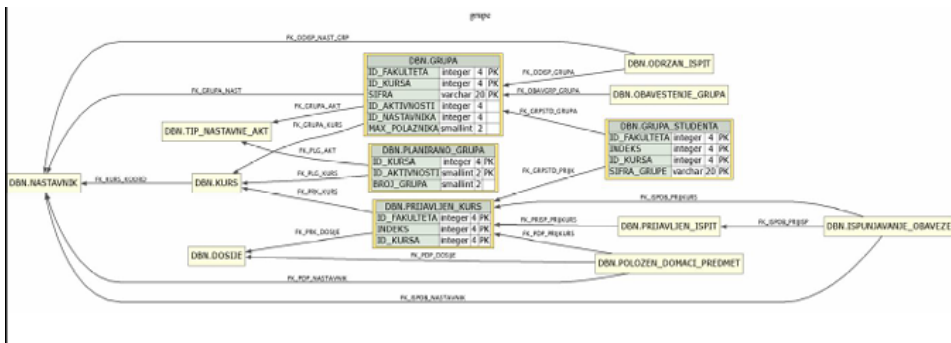
Primer 9a – Automatski generisan dijagram dela modela baze podataka sistema Studinfo



Primer 9b – Automatski generisan dijagram dela modela baze podataka sistema Studinfo



Primer 9c – Automatski generisan dijagram dela modela baze podataka sistema Studinfo



Sadržaj

- 1 Logičko modeliranje
- 2 Dijagrami tabela
- 3 Prevođenje konceptualnog modela u logički

Prevođenje konceptualnog modela u logički

- Prvi korak (prva iteracija) pri pravljenju logičkog modela je prevođenje konceptualnog modela u logički model
 - Ako se oba modela izražavaju na ER-u ili UML-u, onda “*prevođenje*” može da postane “*transformisanje*” ili “*dopunjavanje i nadograđivanje*”
 - U savremenoj praksi je uobičajeno da se logički modeli opisuju dijagramima tabela ili eventualno dijagramima klasa

Prevođenje konceptualnog modela u logički

- Kao što je već naglašeno, u ovoj temi ćemo se posvetiti relacionim bazama podataka
- Pokušaćemo da prevođenje opišemo u formi skupa pravila

Prevođenje ERM na RM

- Prevođenje ER-modela na relacioni model
 - neki aspekti prevođenja su “pravolinijski”
 - neki drugi nisu
 - može da zavisi i od semantike i od konteksta
- Neki slučajevi su jednostavni:
 - Svaki entitet se prevodi u relaciju
 - Atributi koji predstavljaju skupove podataka prevode se u relacije
 - ...
 - ...jednostavnih slučajeva je najviše...
- Ali neki slučajevi su složeniji:
 - ...složenijih slučajeva je manje, ali su od presudne važnosti...

Postupak prevođenja KM u RM

- Postupak prevođenja počinje prevođenjem entiteta
 - usput mogu da se prevedu i neki odnosi, obično 1-*, koji se neposredno prevode u strane ključeve
- Zatim se nastavlja prevođenjem odnosa
 - najpre se prevode odnosi koji proizvode relacije
 - složeni odnosi
 - binarni odnosi *-*
 - zatim se prevode preostali odnosi
 - na kraju se doteruje prevod odnosa
- Najpre ćemo opisati osnovnu ideju prevođenja, a zatim razmotriti različite pojedinačne slučajeve

Prevođenje entiteta u relacije

- Svaki entitet se prevodi u relaciju
 - prepoznaje se primarni ključ
 - prevode se ključni atributi ili se uvode novi surogat ključevi
 - prevode se svi neključni atributi
- Entiteti koji čine hijerarhije se obično najpre prevode u relacije na isti način kao ostali entiteti
- Slab entitet i odgovarajući odnos se obično modeliraju jednom relacijom
 - relaciji se dodaje strani ključ prema vezanom jakom entitetu

Prevođenje atributa

- Skalarni atributi entiteta se prevode u attribute relacija
- Složeni atributi se prevode u više skalarnih atributa
 - Eventualno u relacije, ali se to obično ostavlja za drugu fazu
- Atributi sa više vrednosti se prevode u relacije

Prevođenje odnosa

- Odnosi mogu da se prevedu na dva načina:
 - u strani ključ ili
 - u novu relaciju
- Kod prevođenja odnosa mora da se uzme u obzir priroda relacionog modela
- Odnosi se prevode u zavisnosti od kardinalnosti i vrste odnosa:
 - Binarni odnosi 1-*
 - Binarni odnosi *-*
 - Složeni odnosi
 - Generalizacija

Predstavljanje odnosa stranim ključem

- U relacionom modelu, osnovno sredstvo za modeliranje odnosa su strani ključevi
 - Strani ključ iz “vezane” relacije B prema “baznoj” relaciji A ima vrlo ograničenu semantiku kardinalnosti:
 - kard. A je uobičajeno 0..1, ako ključ može da sadrži NULL
 - kard. A može da bude 1, ako ključ ne sme da sadrži NULL
 - kard. B je uobičajeno 0..*
 - kard. B može da bude 0..1, ako se doda uslov jedinstvenosti ključa uz dopušteno ponavljanje vrednosti NULL
- Sve ostale varijante zahtevaju postavljanje dodatnih uslova i ograničenja
 - a često i dodavanje novih relacija

Prevođenje binarnih odnosa 1-*, 0..1-*

- Ovakve odnose praktično neposredno modelira strani ključ:
 - 0..1-*, strani ključ može da sadrži NULL
 - 1-*, strani ključ ne sme da sadrži NULL
- Takvi odnosi se često prevode već u fazi prevođenja entiteta
 - Ako entitet učestvuje u odnosima 1-* kao slabiji učesnik (u ER-modelu kao učesnik na strani 1, ili u modelu klasa kao učesnik na strani *, ili čak kao eksplicitno označen slab entitet), u prevedenu relaciju se dodaju atributi ključa druge relacije kao strani ključ
 - Ako odnos ima dodatne attribute, oni se dodaju relaciji zajedno sa atributima stranog ključa
 - ako strani ključ može da sadrži NULL, onda to moraju da mogu i ovi atributi
 - štaviše, ako strani ključ sadrži NULL, onda i svi dodatni atributi moraju da budu NULL

Prevođenje binarnih odnosa $*_*$

- Svaki binarni odnos $*_*$ se prevodi u relaciju koja kao strane ključeve sadrži ključeve entiteta koji učestvuju u odnosu
 - Ako odnos ima dodatne atribute, ti atributi se prevode u atribute nove relacije
- Mora da se vodi računa o usklađivanju ograničenja za primarne i strane ključeve sa f. zavisnostima

Prevođenje odnosa sa 3 ili više učesnika

- Složeni odnosi se prevode slično binarnim odnosima *-*
- Svaki odnos se prevodi u relaciju koja kao strane ključeve sadrži ključeve relacija koje su prevodi entiteta koji učestvuju u odnosu
- Dodaju se jedinstveni ključevi u skladu sa f. zavisnostima
- Vodi se računa o usklađivanju ograničenja za primarne i strane ključeve sa f. zavisnostima

Prevođenje atributa odnosa

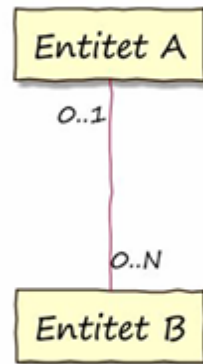
- Ako se odnos prevodi u jedan strani ključ
 - atributi odnosa se dodaju relaciji u kojoj se pravi strani ključ
 - ako atributi stranog ključa mogu da budu NULL onda i ostali atributi preslikani iz odnosa moraju da mogu da budu NULL
- Ako se odnos prevodi u više stranih ključeva
 - (t.j. u strane ključeve u obe relacije, biće više reči o tome)
 - atributi se dodaju u jednu od njih
- Ako se odnos prevodi u novu relaciju
 - atributi odnosa se dodaju toj novoj relaciji
 - opcioni atributi iz KM moraju da mogu da budu NULL

Razmatranje različitih slučajeva

- U nastavku ćemo razmotriti različite vrste odnosa i njihovo prevođenje
- Primere konceptualnog modela ćemo predstavljati (uglavnom) korišćenjem dijagrama klasa domena, ali postupak je praktično isti i u slučaju dijagrama ER

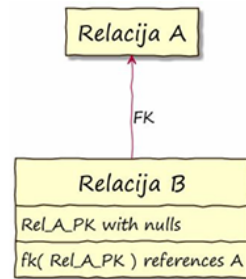
Binarni odnosi 0..1 – 0..N

- Uobičajeno za odnose
 - roditelj–potomak
 - celina–deo
 - ako *celina* može da bude bez *delova*
 - ako *deo* može da bude bez *celine*
- Čest slučaj kod agregacija



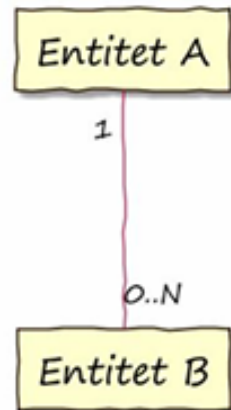
Binarni odnosi 0..1 – 0..N

- Relaciji koja je *deo*
 - dodaje se strani ključ u odnosu na *celinu*
 - strani ključ **može** da bude nedefinisan



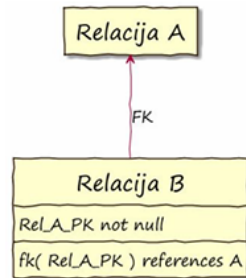
Binarni odnosi 1 – 0..N

- Uobičajeno za odnose
 - roditelj–potomak
 - celina–deo
 - ako *celina* **može** da bude bez *delova*
 - ako *deo* **ne može** da bude bez *celine*
- Čest slučaj kod kompozicija



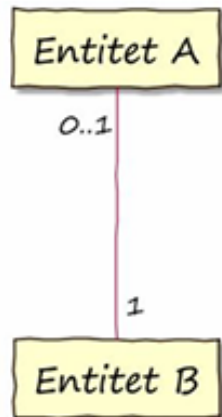
Binarni odnosi 1 – 0..N

- Relaciji koja je *deo*
 - dodaje se strani ključ u odnosu na *celinu*
 - strani ključ **ne sme** da bude nedefinisan



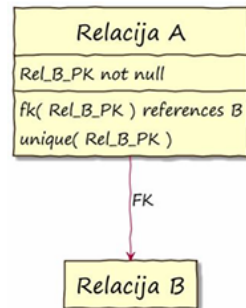
Binarni odnosi 0..1 – 1

- Uobičajeno za odnose
 - nadređeni–podređeni
 - celina–opcionni deo
- Primer:
 - Rukovodilac odeljenja A je službenik B



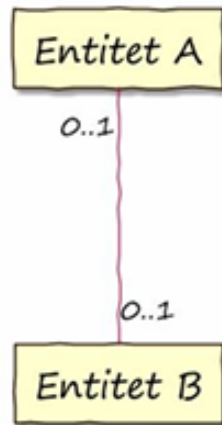
Binarni odnosi 0..1 – 1

- Relaciji koja je *opciona deo*
 - dodaje se strani ključ u odnosu na *celinu*
 - strani ključ **ne sme** da bude nedefinisan
 - vrednost ključa mora da bude jedinstvena



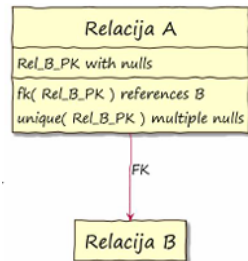
Binarni odnosi 0..1 – 0..1

- Uobičajeno za dvosmerne opcione odnose
- Primer:
 - Projekat A može da ima najviše jednog glavnog programera B
 - Programer B može da bude glavni programer na najviše jednom projektu



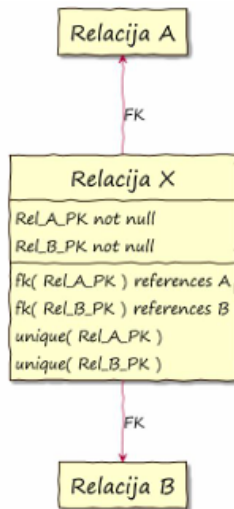
Binarni odnosi 0..1 – 0..1

- Jednoj od relacija
 - dodaje se strani ključ u odnosu na drugu
 - strani ključ **može** da bude nedefinisan
 - vrednost ključa mora da bude jedinstvena
 - nedefinisane vrednosti smeju da se ponavljaju
 - (to je problem kod nekih sistema)



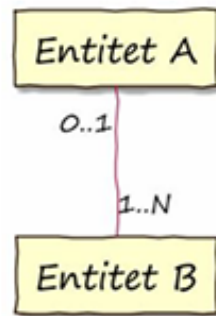
Binarni odnosi 0..1 – 0..1

- ...alternativa...
- Pravi se nova *vezna* relacija
 - sadrži samo strane ključeve u odnosu na obe vezane relacije
 - ključevi **ne smeju** da budu nedefinisani
 - svi atributi čine primarni ključ
 - vrednost svakog od stranih ključeva mora da bude jedinstvena
- Rešenje sa veznom relacijom ima prednost
 - ako je ovaj odnos relativno redak
 - ako je važno poštovanje jedinstvenosti a SUBP ne omogućava jedinstvene ključeve sa višestrukim nedefinisanim vrednostima



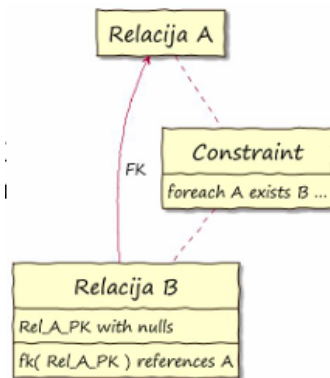
Binarni odnosi 0..1 – 1..N

- Uobičajeno za odnose
 - agregacija sa obaveznim delovima
- Primer:
 - Svako odeljenje A ima najmanje jedno dodeljeno parking-mesto B
 - Svako parking-mesto B može da se dodeli najviše jednom odeljenju A



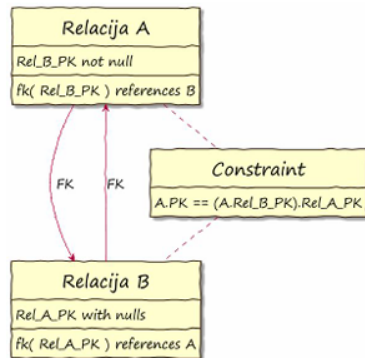
Binarni odnosi 0..1 – 1..N

- Relaciji koja je *deo*
 - dodaje se strani ključ u odnosu na *celinu*
 - strani ključ **može** da bude nedefinisan
 - dodaje se uslov baze podataka: “za svaki entitet A mora da postoji bar jedan odgovarajući entitet B”



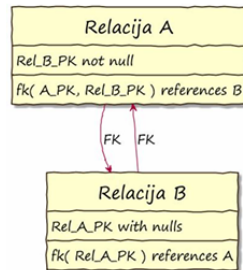
Binarni odnosi 0..1 – 1..N

- ...alternativa 1...
- Među svim delovima jedne celine se prepoznaje (često veštački) jedan izabran
 - u relaciju A se dodaje strani ključ u odnosu na B
 - celina se povezuje sa izabranim delom
 - olakšano je proveravanje postojanja “bar jednog”
 - ali je uvedena redundantnost
 - mora da se doda pravilo koje proverava uzajamnost
 - izabrani deo mora da pripada baš toj celini



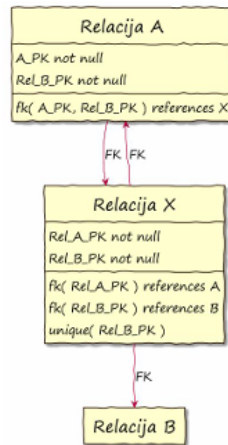
Binarni odnosi 0..1 – 1..N

- ...alternativa 1a...
- Veoma slično kao prethodni slučaj
 - eksplicitan uslov se zamenjuje proširivanjem stranog ključa iz A prema B, tako da uključuje attribute primarnih ključeva i A i B
 - ne omogućavaju svi RSUBP da se referisanje po stranom ključu vrši prema skupu atributa koji nije primarni ključ



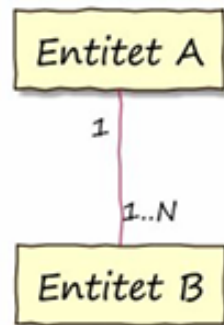
Binarni odnosi 0..1 – 1..N

- ...alternativa 2...
- Uvodi se vezna relacija X
 - ona implementira 0..* – 0..*
 - dodaje se jedinstveni ključ prema B
 - time se rešava odnos 0..1 – 0..*
 - ostaje da se proveri da za svako A postoji bar jedno B
- U relaciju A se dodaje strani ključ u odnosu na X
 - celina se povezuje sa izabranim delom
 - a istovremeno se proverava uzajamnost



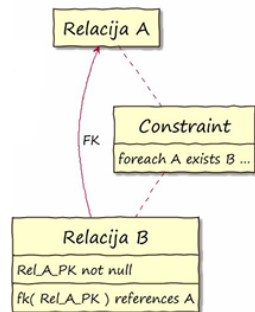
Binarni odnosi 1 – 1..N

- Uobičajeno za odnose
 - kompozicija sa obaveznim delovima
- Primer:
 - Svako odeljenje A ima najmanje jedno dodeljeno parking-mesto B
 - Svako parking-mesto B pripada tačno jednom odeljenju A



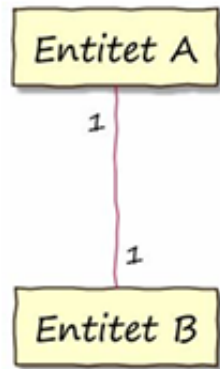
Binarni odnosi 1 – 1..N

- Relaciji koja je *deo*
 - dodaje se strani ključ u odnosu na *celinu*
 - strani ključ **ne sme** da bude nedefinisan
 - dodaje se uslov baze podataka: “za svaki entitet A mora da postoji bar jedan odgovarajući entitet B”
- U svemu slično kao 0..1–1..N
 - pa i u alternativnim rešenjima...



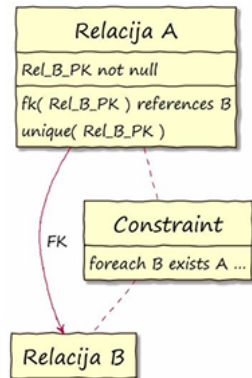
Binarni odnosi 1 – 1

- Uobičajeno za odnose
 - sa obostranim ekskluzivnim pridruživanjem
 - (relativno retko, pitanje je da li su to različiti entiteti)



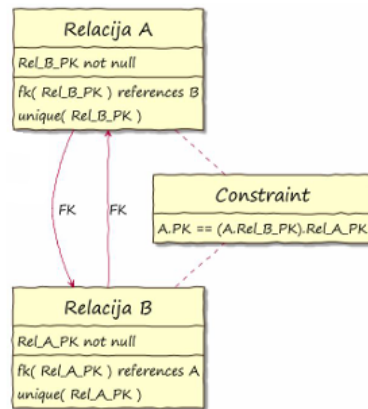
Binarni odnosi 1 – 1

- Jednoj od relacija
 - dodaje se strani ključ u odnosu na drugu
 - strani ključ **ne sme** da bude nedefinisan
 - vrednost ključa mora da bude jedinstvena
 - dodaje se uslov baze podataka: "za svaki entitet B mora da postoji bar jedan odgovarajući entitet A"



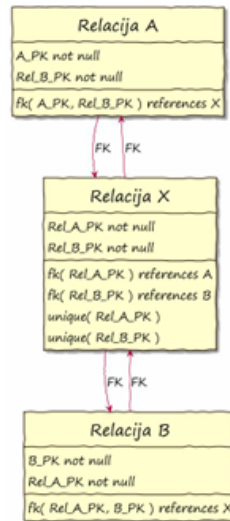
Binarni odnosi 1 – 1

- ...alternativa 1...
- U svakoj od relacija
 - dodaje se strani ključ u odnosu na drugu
 - atributi ključa ne smeju da budu nedefinisani i moraju da budu jedinstveni
 - dodaje se uslov kojim se proverava uzajamnost veze
- ...alternativa 1a...
- U relacijama se definiše isti primarni ključ
 - onda nije potreban dodatni uslov za proveru uzajamnosti veze



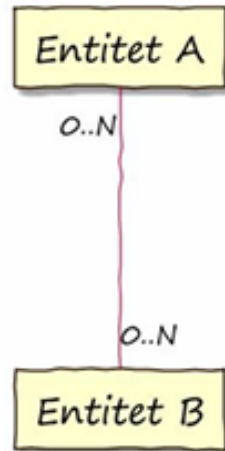
Binarni odnosi 1 – 1

- ...alternativa 2...
- Dodaje se vezna relacija X
 - jedinstven ključ prema A
 - jedinstven ključ prema B
- U svakoj od relacija A i B
 - dodaje se strani ključ u odnosu na veznu relaciju X
 - time se proverava uzajamnost veze



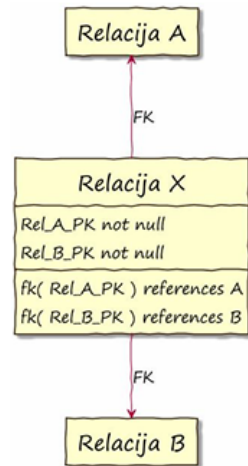
Binarni odnosi $0..N - 0..N$

- Uobičajeno za odnose
 - asocijacije bez posebnih ograničenja



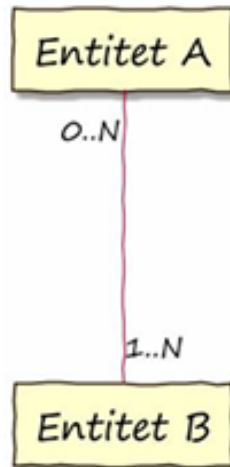
Binarni odnosi $0..N - 0..N$

- Pravi se nova *vezna* relacija
 - sadrži samo strane ključeve u odnosu na obe vezane relacije
 - ključevi **ne smeju** da budu nedefinisani
 - svi atributi čine primarni ključ



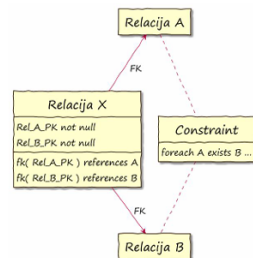
Binarni odnosi $0..N - 1..N$

- Uobičajeno za odnose
 - asocijacije slabih i jakih entiteta
 - agregacije kod kojih *deo* može da čini više celina, ali ne može da postoji samostalno



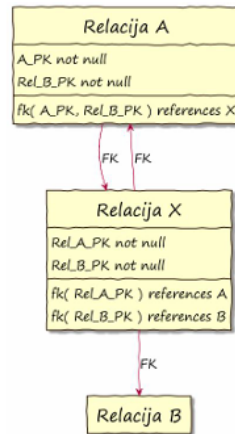
Binarni odnosi $0..N - 1..N$

- Pravi se nova *vezna* relacija
 - sadrži samo strane ključeve u odnosu na obe vezane relacije
 - ključevi **ne smeju** da budu nedefinisani
 - svi atributi čine primarni ključ
 - dodaje se uslov baze podataka: “za svaki entitet A mora da postoji bar jedan odgovarajući entitet B”
- ...alternativa...
 - u A se dodaje strani ključ prema izabranom B odnosno X...



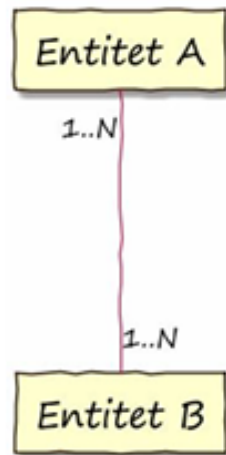
Binarni odnosi $0..N - 1..N$

- ...alternativa...
- U relaciju A se dodaje strani ključ prema X
 - tako se povezuje A sa izabranim B (t.j. parom A, B)
 - nisu potrebni dodatni uslovi
- Nešto je složenije ažuriranje u slučaju promene izabranog para...



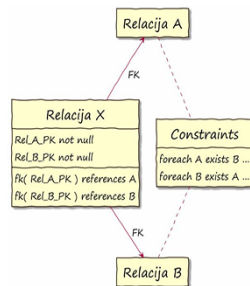
Binarni odnosi $1..N - 1..N$

- Uobičajeno za odnose
 - agregacije kod kojih
 - *deo* može da čini više celina, ali ne može da postoji samostalno
 - *celina* ne može da postoji *bez delova*
 - (relativno retko se koristi)



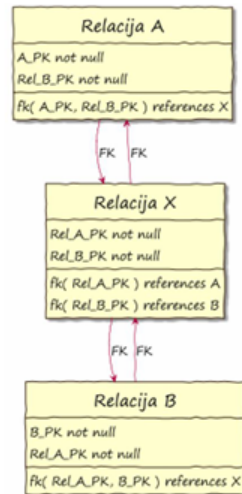
Binarni odnosi 1..N – 1..N

- Pravi se nova *vezna* relacija
 - sadrži samo strane ključeve u odnosu na obe vezane relacije
 - ključevi **ne smeju** da budu nedefinisani
 - svi atributi čine primarni ključ
 - dodaje se uslov baze podataka: “za svaki entitet A mora da postoji bar jedan odgovarajući entitet B”
 - dodaje se uslov baze podataka: “za svaki entitet B mora da postoji bar jedan odgovarajući entitet A”
- ...alternativa...
 - i u A i u B se dodaju strani ključevi prema izabranim delovima/celinama...



Binarni odnosi 1..N – 1..N

- ...alternativa...
 - slično kao i za 0..N – 1..N
 - i u A i u B se dodaju strani ključevi prema relaciji X, kojim se referišu izabrani parovi
 - nisu potrebni dodatni uslovi



...nastavak na sledećem času...

- Primeri...
- Složeni odnosi...
- Hijerarhije entiteta...