

# Prevođenje objektnog modela u relacioni model

---

**Autori:**  
**Goran Savić**  
**Milan Segedinac**

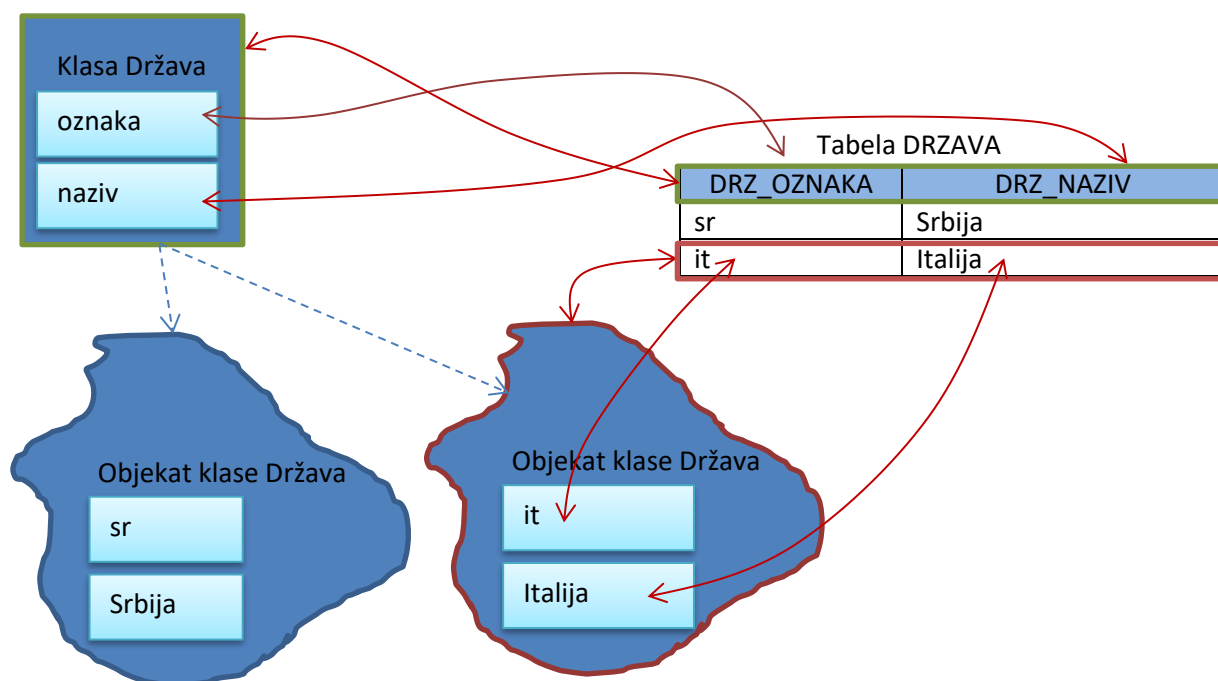
# Prevođenje objektnog modela u relacioni model

## 1. Prevođenje objektnog modela u relacioni model

Ranije smo videli šta predstavlja objektni model podataka i kako se dizajnira. Da podsetimo, objektni model se odnosi na definisanje klasa i relacija između njih. Ovaj model specificira koji podaci će biti evidentirani u radnoj memoriji u toku izvršavanja aplikacije i kako će ovi podaci biti organizovani. Takođe, upoznali smo se i sa relacionim modelom podataka. Ovaj model specificira tabele relacione baze podataka i njihovu organizaciju. Relacioni model podataka definiše koji podaci se skladište u relacionoj bazi podataka i kako su oni organizovani.

Dakle, aplikacija podatke skladišti u radnoj memoriji (privremeno) i u bazi podataka (dugotrajno). Objektni model i relacioni model podataka za jednu aplikaciju su namenjeni skladištenju istih podataka, pa je važno da uporedimo ove modele i uspostavimo analogije između njih. Ovo će nam omogućiti lakše dizajniranje jednog modela na osnovu drugog. Takođe, moguće je automatizovati prebacivanje podataka iz jednog modela u drugi.

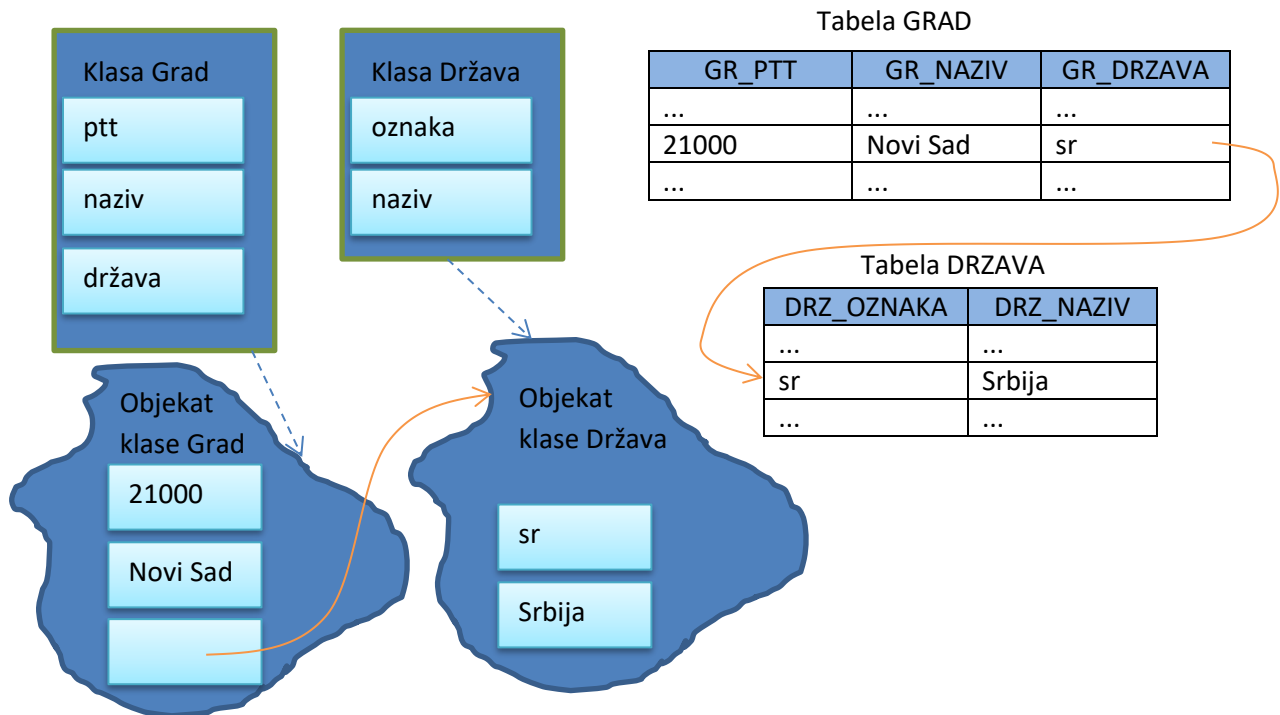
Kada je reč o objektnom modelu, osnovna jedinica evidentiranja podataka za određeni entitet je objekat. U relacionom modelu, podaci o određenom entitetu se skladište kao slog u tabeli. Jedan podatak o entitetu je u objektnom modelu predstavljen atributom objekta, a u relacionom modelu poljem u redu tabele. Šema podataka koji se skladište je u objektnom modelu definisana klasom, a u relacionom modelu tabelom. Ove analogije su ilustrovane na slici kroz primer skladištenja podataka o državama.



## 2. Prevođenje veze asocijacije

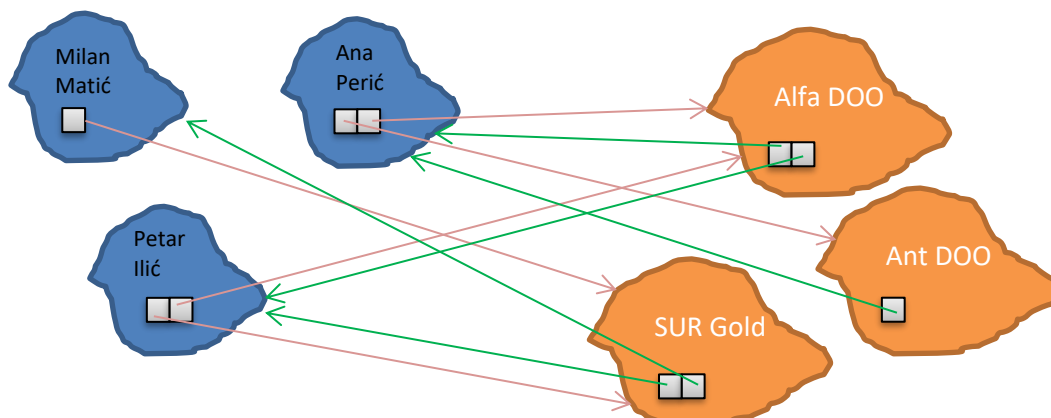
Veza asocijacije između klasa u objektnom modelu se u relacionom modelu predstavlja spoljnim ključem. Ovo je ilustrovano na sledećoj slici kroz primer evidentiranja države u kojoj se grad nalazi.

## Prevođenje objektnog modela u relacioni model



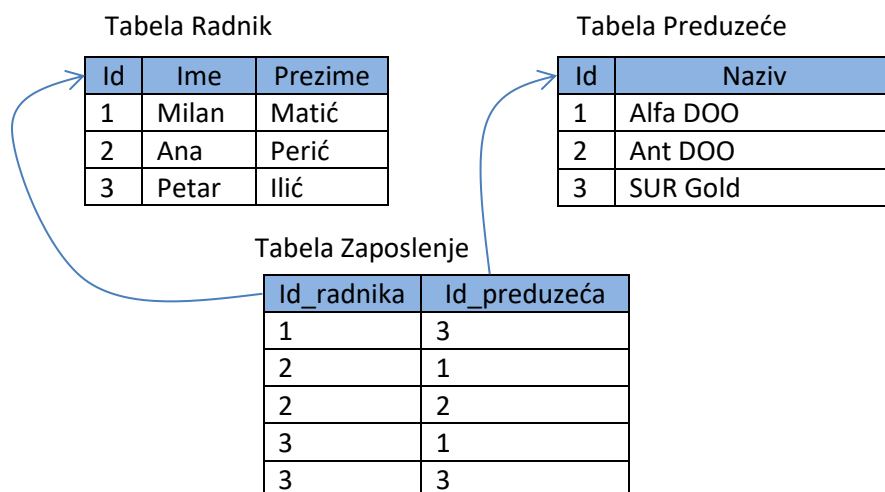
Prikazani primer se odnosi na vezu asocijacije višestrukosti 1:N, jer se u jednoj državi nalazi više gradova. Treba primetiti da je u objektnom modelu za ovakav tip asocijacije moguće definisati dvosmernu vezu, dok je u relacionom modelu veza uvek jednosmerna. Naime, pored informacije o državi u kojoj se grad nalazi, u objektnom modelu možemo specificirati i koji se sve gradovi nalaze u državi. Ovo specificiramo tako što kao atribut klase Država uvedemo kolekciju objekata klase Grad. U relacionom modelu nije moguće niti potrebno u slogu koji predstavlja državu definisati gradove koji se u njoj nalaze. Ovo je indirektno definisano time što svaki grad referencira državu u kojoj se nalazi.

Razmotrimo sada predstavljanje veze asocijacije višestrukosti N:N. Ova veza se u objektnom modelu može predstaviti tako što definišemo kolekcije u oba objekta koji su u vezi. Razmotrimo to na jednom primeru. Posmatrajmo evidenciju preduzeća i radnika, što možemo modelovati klasama **Preduzeće** i **Radnik**. Na koji način u objektnom modelu možemo da evidentiramo preduzeća u kojima je radnik radio? Radnik je radio u različitim preduzećima, a sa druge strane u jednom preduzeću su radili različiti radnici. Ovo je veza tipa N:N, koja se može predstaviti tako što ćemo u klasi **Preduzeće** kao atribut definisati kolekciju objekata klase **Radnik**. Takođe, u klasu **Radnik** ćemo definisati atribut koji je kolekcija objekata klase **Preduzeće**. Ovo je ilustrovano na slici.



## Prevođenje objektnog modela u relacioni model

Iste informacije se ne mogu na ovaj način uskladištiti i u relacionom modelu. Naime, veza u relacionom modelu se ostvaruje uvođenjem spoljnog ključa u jednu tabelu, što uvek specificira vezu N:1 (N slogova tabelle u koju je uveden spoljni ključ referencira jedan slog druge tabelle). U relacionom modelu nije moguće u polje sloga upisati kolekciju, kao što je to moguće sa atributom tipa kolekcija u objektnom modelu. Iz ovog razloga, veza N:N se u relacionom modelu rešava uvođenjem nove tabelle. Često ovu tabelu nazivamo vezna tabela. Nova tabela sadrži spoljni ključ na obe tabelle koje formiraju N:N relaciju. Time smo N:N relaciju modelovali kroz dve veze 1:N. Pogledajmo prethodni primer sa modelom radnika i preduzeća preveden u relacioni model.



Kao što vidimo, dva entiteta iz objektnog modela su predstavljena sa tri tabelle u relacionom modelu. Zaposlenje predstavlja veznu tabelu. Ova tabela ima spoljni ključ i na tabelu Radnik i na tabelu Preduzeće. Praktično slogovi tabelle Zaposlenje označavaju mapiranje radnika na preduzeće. Drugim rečima, ovi slogovi definišu koji radnik je radio u kojem preduzeću. Postavlja se pitanje šta je primarni ključ vezne tabelle. Primarni ključ se može definisati kao kompozitni, kombinacijom dva spoljna ključa koje tabela sadrži. Druga varijanta je uvođenje treće kolone koja bi bila *auto increment* surogat primarni ključ.

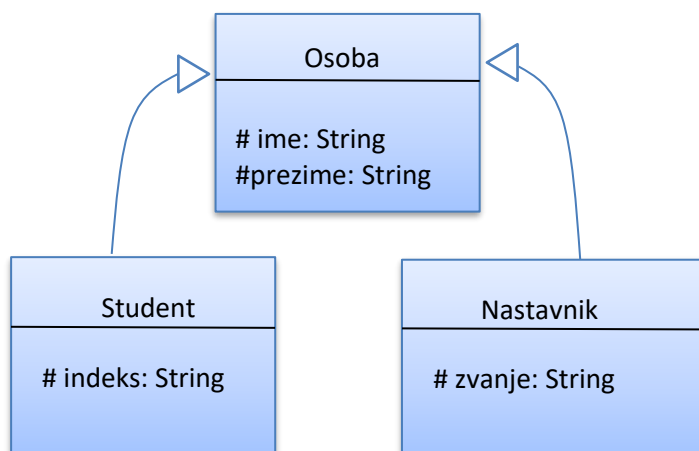
Treba naglasiti da u objektnom modelu nema potrebe za uvođenjem vezne tabelle samo ako za samu vezu nije vezana neka dodatna informacija. Na primer, ako bismo želeli da evidentiramo i datum početka zaposlenja radnika u određenom preduzeću, tada bismo morali da uvedemo klasu Zaposlenje u objektni model. Ta klasa bi sadržala pomenuti datum početka zaposlenja, kao i referencu na objekat klase Radnik i referencu na objekat klase Preduzeće. Time bi objekat klase Zaposlenje predstavljao informaciju o tome koji radnik je bio zaposlen u kom preduzeću i sadržao sve dodatne informacije koje se tiču tog zaposlenja. Informacija o datumu početka zaposlenja bi se u relacionom modelu uskladištila kao treće polje tabelle Zaposlenje.

### 3. Prevođenje veze nasleđivanja

Pored veze asocijacije, u objektnom modelu je moguće između klasa uspostaviti i vezu nasleđivanja. Obzirom da se ovde bavimo modelom podataka, interesuje nas nasleđivanje podataka (atributa), a ne ponašanja (redefinisanjem metoda). Ovaj tip veze ne postoji u relacionom modelu, pa je neophodno pronaći način za očuvanje informacija iz objektnog modela na neki drugi način.

## Prevođenje objektnog modela u relacioni model

Razmotrimo mogućnosti mapiranja nasleđivanja iz objektnog na relacioni model na sledećem primeru. Ako program evidentira podatke o studentima i nastavnicima, određen broj podataka za ova dva tipa entiteta je zajednički pa se može izdvojiti u klasu koja je predak ovim tipovima entiteta. Tu klasu ćemo nazvati *Osoba*. Naredna slika ilustruje ovaj objektni model.



Prikazani objektni model se može mapirati na relacioni koristeći jednu od sledeće tri strategije:

- Jedna tabela za hijerarhiju klasa
- Tabela za svaku konkretnu klasu
- Tabela za svaku klasu

### Jedna tabela za hijerarhiju klasa

Kod ove strategije, sve klase koje čine jednu hijerarhiju tipova u objektnom modelu se prevode u jednu zajedničku tabelu u relacionom modelu. Za svaki atribut svake klase u hijerarhiji, ova tabela sadrži odgovarajuće polje. Pogledajmo objektni model sa prethodne slike preveden u relacioni model po ovoj strategiji.

Id	Tip osobe	Ime	Prezime	Indeks	Zvanje
1	S	Marko	Marković	17382	null
2	N	Petar	Petrović	null	docent
3	S	Ana	Milić	17401	null
4	N	Jovana	Antić	null	vanredni profesor

Vidimo da tabela skladišti i entitete tipa student i entitete tipa nastavnik. Iz tog razloga, u tabelu je uvedeno i polje *Tip osobe* koje sadrži informaciju da li se slog odnosi na nastavnika ili studenta. U prikazanom primeru je izabrano da ova informacija bude označena jednim karakterom (S za studenta, N za nastavnika). Skladištenje različitih entiteta u jednoj tabeli ima za posledicu da neka od polja u slogu uvek imaju vrednost null. To su za slogove koji predstavljaju studente ona polja koja se odnose na podatke o nastavniku. Obrnuto važi za slogove koji predstavljaju nastavnike. Broj slogova u tabeli je jednak zbiru evidentiranih učenika i nastavnika. Ovo treba imati u vidu kod aplikacija sa velikom količinom podataka.

# Prevođenje objektnog modela u relacioni model

## Tabela za svaku konkretnu klasu

Ako je hijerarhija klasa takva da je korenska klasa apstraktna, tada možemo primeniti ovu strategiju. Ovo je slučaj u prikazanom primeru gde je klasa Osoba apstraktna, što znači da u evidenciji uvek postoje ili studenti ili nastavnici, a nikad entiteti koji predstavljaju apstraktnu osobu. Ova strategija podrazumeva definisanje posebne tabele u relacionom modelu za svaku od konkretnih klasa iz objektnog modela. Dakle, za naš primer ćemo imati dve tabele u relacionom modelu. Ovo je ilustrovano na slici.

Tabela Student

Id	Ime	Prezime	Indeks
1	Marko	Marković	17382
2	Ana	Milić	17401

Tabela Nastavnik

Id	Ime	Prezime	Zvanje
1	Petar	Petrović	docent
2	Jovana	Antić	vanredni profesor

Vidimo da se sada studenti i nastavnici evidentiraju u odvojenim tabelama. Obzirom da oba tipa entiteta sadrže ime i prezime, ovi podaci su deo šeme obe tabele. Svako proširenje skupa podataka koje se evidentiraju za osobu, povlači proširenje šeme obe prikazane tabele.

## Tabela za svaku klasu

Treća strategija pri prevođenju objektnog u relacioni model podataka je da se kreira tabela u relacionom modelu za svaku klasu iz objektnog modela. Svaka tabela sadrži polja koja odgovaraju atributima u klasi koju predstavlja. Tabele koje predstavljaju klase naslednice, putem spoljnog ključa referenciraju informacije uskladištene u tabeli koja predstavlja klasu predak. Za primer sa evidencijom nastavnika i studenata to znači da će u relacionom modelu biti tri tabele. Ove tabele su prikazane na slici.

Tabela Osoba

Id	Ime	Prezime
1	Marko	Marković
2	Petar	Petrović
3	Ana	Milić
4	Jovana	Antić

Tabela Student

Id osobe	Indeks
1	17382
3	17401

Tabela Nastavnik

Id osobe	Zvanje
2	docent
4	vanredni profesor

Dakle, sada tabela osoba skladišti sve podatke koji su zajednički za studente i nastavnike. U posebnim tabelama se za studente, odnosno nastavnike skladište samo informacije koje su specifične za taj podtip entiteta osoba. Polje *Id osobe* u ove dve tabele je spoljni ključ na tabelu Osoba. Tako da su sada podaci o studentima, odnosno nastavnicima raspoređeni u dve tabele. Dodatno, polje *Id osobe* u tabelama Student i Nastavnik je istovremeno i primarni ključ za ove tabele, obzirom da su vrednosti u tom polju jedinstvene.