



# DOMAĆA ZADAĆA 1

**Osnove baza podataka**

**Ime i prezime:** Nur Osmanbegović

**Br. Indexa:** 17501

**Odsjek:** Računarstvo i informatika



## Sadržaj

Zadatak 1 – ERD .....	3
Kreiranje tabela .....	5

## Zadatak 1 – ERD

ERD sam crtala u Visual paradigmu iz razloga što je to najmoćniji alat za crtanje UML dijagrama, koji sam ujedno koristila i na drugim predmetima pri crtanju dijagrama.

U nastavku ću obrazložiti i objasniti zašto sam neke stvari odlučila ovako modelirati (samo ključne dijelove oko kojih sam se dvoumila), te ću navesti primjere za svaku vezu:

- **JEDAN NA VIŠE**

U postavci zadaće je bilo rečeno da će kompaniji biti potrebni izvještaji na nivou država i kontinenata te sam iz tog razloga odlučila izdvojiti te entitete. Veza kontinent:država je jedan na više, iz razloga što jedan kontinent može imati više država, ali jedna država može pripadati samo jednom kontinentu. Takva veza, vodeći se istom logikom, postoji i između države i grada.

- Za razliku od pravnih lica koji imaju sve zajedničke atribute (te sam ih odlučila modelirati kao jedan entitet sa dodatnim atributom tip), fizička lica su se razlikovala u većem broju atributa. Dugo sam razmišljala koji bi bio najbolji, optimalni, način za modeliranje fizičkih lica:

- Prvo sam razmišljala da ih modeliram kao što sam i pravna lica. Međutim, uposlenici imaju 5 dodatnih atributa u odnosu na kupce, što u slučaju malog broja kupaca i nije problem. Kako raste broj kupaca (po postavci se da zaključiti da je svjetska kompanija u pitanju), količina rezervisane memorije koja je logički slobodna (ta sva polja koja uposlenici imaju a kupci ne, bi u slučaju kupaca bila null) raste, te je to bespotrebno trošenje resursa. Iz navedenih razlog sam ipak odlučila odustati od ove ideje.
- Druga ideja je bila da skroz razdvojim uposlenike i kupce i da njihove primarne ključeve držim u tabeli FIZIČKA\_LICA sa dodatnim poljem tip, međutim, to rješenje sam odlučila odbaciti zbog dupliciranja kolona koje su zajedničke, te zato što bih na taj način „izgubila“ logičku vezu između kupovine i uposlenika, a oni mogu biti kupci. Također bih na taj način odjele i pozicije u firmi logički povezala sa kupcima, što nema smisla dozvoliti.
- Na kraju sam se odlučila za rješenje koje vidite na ERD-u: U tabeli fizička lica se nalaze i kupci i uposlenici sa svojih 5 zajedničkih atributa, ali za uposlenike postoji „dopunska“ tabela u kojoj se nalaze dodatni atributi. Spajanjem preko ključeva je moguće doći do tih dopunskih informacija – nema dupliciranja entiteta, poslovi unutar firme su povezani isključivo sa uposlenicima, dok su kupovine povezane i sa uposlenicima i sa kupcima, odnosno sa fizičkim licima općenito.

- **JEDAN NA JEDAN**

Vezu 1 na 1 sam iskoristila pri modeliranju fizičkih lica, odnosno za spajanje fizičkih lica i dopunskih informacija za tip uposlenika.

- Obzirom na dio postavke u kojem se spominju nadkategorije, to sam modelirala kao vezu jedan na više na istu tabelu.



- **VIŠE NA VIŠE**

Veze više na više sam riješila tabelom veze, kao primjer takve tabele je tabela KOLICINE\_U\_SKLADISTIMA. Skladišta mogu imati više proizvoda, a jedan proizvod se može nalaziti u više skladišta, pri čemu je količina jednog proizvoda u jednom skladištu neovisna od ostalih. U ovoj tabeli veze imam id\_proizvoda, id\_skladista te količinu tog proizvoda u tom skladištu, na taj način sam riješila vezu više na više između proizvoda i skladišta.

- Još jedna veza više na više je veza između fakture i proizvoda. Jedan proizvod se može nalaziti na više faktura, a jedna faktura može imati više proizvoda. Ovdje sam ovo riješila tabelom veze STAVKE\_FAKTURE, jedan stavka je vezana isključivo za jedan proizvod i isključivo za jednu fakturu, dok jedan proizvod može biti na više stavki i jedna faktura može imati više stavki.
- Vjerovatno me najviše namučio dio sa popustima, razmišljala sam da samo entitetu proizvod dodam polje popust, međutim, na taj način bih izgubila svaki trag o popustima na određeni proizvod, tj. „historiju“ popusta, što je za svaku firmu koja se bavi prodajom sigurno neprihvatljivo. Na kraju sam odlučila imati entitet popust, sa atributom id\_proizvoda, popustom, datum\_od (koji se obavezno unosi) i datum\_do (koji može biti null obzirom da se treba unijeti tek kada popust završi ili kada se promijeni).



## Kreiranje tabela

Pri kreiranju tabela nisam naišla ni na kakve probleme. Karakteristično je bilo kreiranje pojedinih tabela i veza koje ću navesti u nastavku:

Tabela kategorije ima vezu jedan na više na samu sebe, odnosno ima foreign key na samu sebe. Ne može se dodati referenca na atribut entiteta koji još nije kreiran, te sam kreirala tabelu sa svim poljima, primary key kao i foreign key, s tim da foreign key nisam označila kao foreign key nego sam naredbom ALTER TABLE nakon kreiranja tabele, za kolonu nadkategorije dodala referencu na istu tabelu kojoj taj atribut pripada.

Uposlenici i odjel su vezani sa dvije veze jedan na više: jedan uposlenik može raditi u jednom odjelu, jedan odjel može imati više uposlenika. S druge strane, svaki odjel ima jednog šefa, a svaki uposlenik može biti šef u više odjela. Prvo sam kreirala tabelu UPOSLENICI, kod kojih kolona odjel ne predstavlja foreign key te ne referencira ni na šta. Zatim sam kreirala tabelu ODJEL, sa foreign key id\_sefa koji referencira na tabelu UPOSLENICI, a nakon kreiranja tabele ODJEL sam naredbom ALTER TABLE modificirala tabelu UPOSLENICI i dodala referencu za atribut odjel na atribut id\_odjela entiteta odjel.