

Elektrotehnički fakultet

Univerzitet u Sarajevu

Predmet: Osnove baza podataka

## ZADAĆA 1

Sudent: Nedžad Hadžiosmanović

Broj indeksa: 18033

Asistent: Šeila Bećirović

Sarajevo , 4 .11.2018.

## **Odabit Visual Paradim-a:**

Svoj ERD (Entity Relationship Diagram) sam odabrao da radim u programu „Visual paradigm“, iz prostog razloga što mi se učinio kao lagan alat za savladati, koji ima sve potrebne elemente da bih završio svoj zadatak. Također, ovaj program, s obzirom da je koristen od velike strane ljudi, posjeduje mnogo online tutorijala i vodiča koji samo savladavanje njegovih osnova čine još lakšim, te me to dodatno nagnalo da izaberem baš njega.

## **O mom ERD-u :**

Nakon čitanja postavke zadaće i pitanja koja su ostali studenti postavljali na grupi nadležnima, u nekoliko navrata sam pokušavao nacrtati shemu za pomenuti zadatak. Krajnja verzije koju sam odlučio realizovati u odabranom programu za modeliranje entiteta, njihovih atributa i veza između odabranih entiteta sadrži 22 tabele od kojih je nekoliko vezanih. U nastavku teksta ću u kratkim crtama pokušati objasniti svoj dijagram.

Prva tabela koju sam modelirao, „SKLADISTE“, prema tekstu zadaće, trebala je imati attribute koji daju ime skladišta, adresu skladišta, grad u kojem se skladište nalazi, državu i kontinent. S obzirom da se podaci o tome koji u kojem se gradu kontinentu i državi nalazi skladište logički ne moraju biti definisani direktno u tabeli skladište. Iako jedno skladište može biti na jednom mjestu, u jednoj državi i jednom kontinentu, u jednom gradu može biti više skladišta, kao i u državi i na kontinentu, te sam se stoga odlučio napraviti zasebne entitete „GRAD“, „DRŽAVA“, „KONTINENT“ kojima će se moći pristupiti iz tabele skladište u koje sam stavio strane ključeve koji su jednaki stvarnim ključevima pomenute tri tabele. Zbog opisane relacije u kojoj u jednom gradu, državi ili kontinentu može biti više skladišta, sve tri veze su oblike n:1 koje su napravljene na osnovu primarnog ključa tabele „GRAD“, „DRŽAVA“ i „KONTINENT“. Također, ta tri entiteta su međusobno povezana vezama n:1, grad se može nalaziti u jednoj državi, dok u jednoj državi ima više gradova. Isto vrijedi i za vezu između država i kontinenta.

Razlog zbog kojeg je veza „SKLADISTE“-„GRAD“ n:1, što sam modelirao grad tako da ima tri atributa, od kojih niti jedan ne specificira taj entitet kao što bi bila „adresa“. U tom slučaju tada bi veza morala biti 1:1, jer na jednoj adresi se ne može nalaziti više skladišta, dok ovakvim odabirom atributa dobijamo mogućnost da veu modeliramo kao n:1, jer u jedno skladište može biti samo u jednom gradu, dok u jednom gradu može biti više skladišta.

Vezu između entiteta „UPOSLENIK“ i entiteta „SKLADISTE“ ostavrujem vezom 1:n, jer više skladišta mogu ići na jednog odgovornog (urađeno na osnovu jednog od odgovora na sheet-u)

Prelazimo na tabelu/entitet „UPOSLENIK“ o kojem pored osnovnih informacija kao što su njegovo ime, prezime, adresa, grad, država, odjel u kojem radi, ugovor o zaposlenju, te ko je šef uposleniku.

S obzirom da svaki uposlenik ima šefa, međutim i sam šef je uposlenik, kojem nema niko nadređen, entitet "UPOSLENIK" se povezuje sa samim sobom vezom 1:n, jer jedan šef ima više uposlenika, pri čemu sam stavio da atribut "sef\_id" može biti null (nullable), iz razloga što šef, nema šefa. Pri ostavri vanju te veze, vežemo ključ tabele, sa atributom "sef\_id".

Druga stvar, koju u prvim pokušajima pisanja tabela I vzeo između njih, nisam uradio je da sam informacije kao što su ime zaposlenika, prezime, adresa stanovanja i grad u kojem radi čuvao u zasebnoj novoj tabeli, tabeli "FIZICKO\_LICE". Radi izbjegavanja dupliranja atributa u vise entiteta, s obzirom da se naknadno iz teksta zadaće saznanje da je I "KUPAC" jedan vrsta fizičkog lica, i novu tabelu "KUPAC" vežemo za "FIZICKO\_LICE", u kojoj također izbjegavamo dupliranje atributa koje smo naveli u tabeli "FIZICKO\_LICE". I tabela "KUPAC" I tabela "UPOSLENIK" su vezani za tabelu "FIZICKO\_LICE" vezama 1:1, jer jedan uposlenik može biti jedno fizičko lice, dok također, jedan kupac može biti jedno fizičko lice. Također, jedno fizičko lice može biti ili uposlenik, ili kupac. Stoga u entitete "KUPAC" i "UPOSLENIK" dodajemo strani ključ koji odgovara primarnom ključu tabele "FIZICKO\_LICE" na osnovu čega se ostavruju veze. U tabelu "FIZICKO\_LICE" dodajemo atribut "vrsta" na osnovu kojeg ćemo znati da li je fizičko lice zaposlenik ili kupac, recimo ako definišemo da vrsta=1 odnosi na uposlenika, a vrsta=2 odnosi na kupca (uz uslov da ovaj atribut nikada ne smije biti null, da bi u svakom trenutku znali na koga se odnosi fizičko lice, što će nam biti potrebno u nekim budućim tabelama).

Također sam se odlučio napraviti zasebnu tabelu pod nazivom "UGOVOR\_UPOSLENIKA". Prvenstveno, ova tabela se zvala samo "UGOVOR", međutim, u tabeli koju ćemo kasnije spomenuti "PRVANA\_LICA" javljala se također potreba za tabelom "UGOVOR", pa sam prvenstveno mislio realizovati tu tabelu tako da odgovara i za uposlenike I pravna lica, što bi ostavrio preko istog principa kao I u tabeli "FIZICKO\_LICE", tj. dodao bih u tabelu "UGOVOR" atribut pod nazivom vrsta koji bi specificirao o kojem tipu ugovora je riječ. Dok sam radio zadatku učinilo mi se kao praktičnije I logičnije rješenje da zamišljenu tabelu "UGOVOR" razbijem na dvije zasebne tabele, "UGOVOR\_UPOSLENIKA" i "UGOVOR\_PRAVNIH", s obzirom da radi u potpunosti dva različita tipa ugovora. "UGOVOR\_UPOSLENIK" posjeduje atribute "datum\_potpisivanja" i "datum\_raskidanja", od kojih logički gledano datum raskidanja može biti null, jer uposlenik može biti zaposlen ugovorom koji nema ograničenja, već se radi o stalnom radnom odnosu. Uz te atribute, tabela "UGOVOR\_UPOSLENIKA" ima ključ tabele, "ugovor\_usposlenika\_id" koji se također nalazi u entitetu "UPOSLENIK", kao strani ključ na osnovu kojeg je ostavrena veza 1:1, jer jedan usposlenik može imati jedan ugovor, isto kao što jedan ugovor u zaposlenju može biti vezan za samo jednog uposlenika.

Za razliku od "UGOVOR\_UPOSLENIK", entitet "UGOVOR\_PRAVNOG", koji posjeduje iste atribute, atribut "datum\_raskidanja" ne može biti null, jer vrlo rijetko u stvarnom životu se mogu susresti situacije u kojima kompanija potpisuje ugovor s nekim pravnim licem (kuriskom službom ili

proizvođačem) u kojoj se radi o stalnom radnom odnosu, već se takvi tipovi ugovora potpisuju na određene vremenske periode, te konstantno obnavljaju.

Vratimo se na entitet uposlenik da bi objasnili zadnju tabelu sa kojom je vezan. Radi se o entitetu "ODJEL". Za zasebnu tabelu za odjel sam se odlučio iz razloga što ova tablea logički ne pripada samom zaposleniku, već samo predstavlja jednu dodatnu informaciju o njemu. S obzirom da jedan uposlenik može raditi u jednom odjelu, dok jedan odjel može zapošljavati jednog ili više radnika, ostavrio sam vezu n:1 pomoću ključa tabele "ODJEL" i stranog ključa tabele "UPSOLEJNIK" koji odgovara ključu tabele "ODJEL".

Također, u tabeli odjel postoji i atribut "sef\_id" zato što svaki odjel ima svog jednog šefa, te to predstavlja bitnu karakteristiku samog odjela. "sef\_id" će, ukoliko se zaista pristupa ovom atributu preko tabele "USPOSLENIK" čiji atribut "sef\_id" nije null (znaci radi se o radniku, a ne sefu), imati dvije kolone koje daju iste informacije, dok će u drugu ruku, ako se pristupa ovom atributu preko entiteta "UPOSLENIK" u kojoj je sef\_id null (radi se o sefu), tada će "zaposlenik\_id" (ključ tabele uposlenik) iz tabele "UPOSLENIK" i "sef\_id" iz tabele "ODJEL" biti isti.

Tabela "PRAVNA\_LICA" je povezana vezama 1:1 sa tabelama "PROIZVODJAC" i "KURIRSKA SLUŽBA" koje u sebi imaju sporedni ključ jednak pravome ključu tabele "PRAVNO\_LICE", tj. "pravno\_lice\_id". Jedno pravno lice može biti ili proizvođač, ili kurirska služba, dok jedan proizvođač može biti jedno pravno lice, i također jedna kurirska služba može biti jedno pravno lice.

Pređimo na pojašnjenje tabele "PROIZVOD". Ova tabela treba biti vezna vezom n:m sa entitetom "SKLADIŠTE", jer se u jedno skladištu može nalaziti više istih proizvoda, te se jedan te isti proizvod može nalaziti u više skladišta. Za tu svrhu sam kreirao poveznju tabelu "POVEZNA\_SKLADISTE-PROIZVOD" u koju sam stavio primarne ključeve tabele "PROIZVOD" i "SKLADIŠTE", čime sam veznu n:m, razbio na dvije veze 1:n, a povezivanje izvršio pomoću primarnih ključeva.

Entitet "PROIZVOD" ima svoju kategoriju, koja logički gledano predstavlja zaseban entitet. Veza između ova dva entiteta bi mogla n:m, s obzirom da jedan proizvod može spadati u više kategorija, a jedna kategorija može imati više proizvoda, kao npr. neki kabl, mogao bi se naći i u kategoriji oprema za računa, i u kategoriji kablovi. Stoga, da bi pretragu nekom ko posjećuje stranicu za kupovinu, olakšali posao pretraživanja, ovu vezu realiziramo više na više. Ona se realizira poveznjom tabelom "POVEZNA\_PROIZVOD-KATEGORIJA" koja sadrži primarne ključeve entiteta "PROIZVOD" i "KATEGORIJA" na osnovu kojih se ostavruju dvije 1:n veze.

Tabela "PROIZVOD" sadrži i atribut "proizvodjaci\_id", jer svaki proizvod ima svog unikatnog proizvođača, tj. taj atribut je strani ključ na osnovu kojeg je ostavrena veza 1:n između "PROIZVODA" i "PROIZVODJAC", jer jedan proizvođač može napraviti više proizvoda.

"GARANCIJA" je entitet koji vezemo za entitet "PROIZVOD" vezom 1:n, jer jedna garancija može biti vezana za više proizvoda, dok jedan proizvod mora imati jednu garanciju.

Entitet "POPUST" je vezan i za entitet "PROIZVOD" i za pomocni entitet "POVEZNA\_PROIZVOD-RACUN", jer popust može biti različit na računu koji je izdat u određenom trenutku, a da poslije popust na proizvodu naraste, ili čak i nismo. Zbog te činjenice, strani ključ u tabeli "POPUST" može biti null.

Iako se možda mogla ostavriti direktna veza između entiteta "KUPAC" i entiteta "RACUN", u tablici "RACUN" sam se odlucio da iskoristim strani ključ "fizicko\_lice\_id", preko kojeg bih uvijek znao da li je osoba na čije ime se izdaje račun radnik ili kupac. Ostavrenje veze na ovaj način mi olakšava da pri izradi računa znamo da li se radi o kupcu ili o uposleniku, jer uposlenici mogu dobiti dodatni popust na kupljene stavri.

"RACUN" je povean povznom tabelom sa proizvodom, jer se na jednom računu može nalaziti više proizvoda, i jedan te isti proizvod se može naći na više računa, pa je n:m vezu bilo potrebno razbiti upotrebom povezne tabele. U tu poveznu tabelu sam dodao i atribut "kolicina" s obzirom da da logicki gledano taj atribut ne pripada ni "PROIZVOD" tabeli, ni "RACUN" tabeli, jer svaki proizvod ima zasebnu količinu.

"RACUN" također posjeduje vezu sa dosad ne spomenutim entitetom "ISPORUKA", za koji je vezan preko stranog ključa "isporuka\_id". U jednoj isporuci može biti više računa, dok jedan račun se može isporučiti u samo jednoj isporuci, pa je pomenuta veza 1:n. "ISPORUKA" je vezana za "KURIRSKA\_SLUZBA" vezom n:1, jednu isporuku vrši jedna kurirska služba, dok jednu kurirska služba vrši više isporuka.

## **Pisanje koda u SQLTools:**

Prilikom pisanja koda u SQLToolsu naisao sam na probleme povezivanja 1:1, koje sam riješio tako što sam vezu opisanu kao 1:n sa predavanja 5, „nadogradio“ koristeći ključnu riječ 'unique'.

Još jedan problem na koji sam naišao bila je činjenica da nisam znao da iza svakog atributa napravljenje tabele se mora napisati da li sta vrijednost može, ili ne biti null, sa odgovarajućim porukama 'null' i 'not null'.

Naučio sam da nakon što je kreirana tabela, koji naknadno izmijenimo ostaje spašena na našim akauntovima na SQLTools-u. To sam provjeravao koristeći naredbu 'Object Wierer' koja daje prikaz svih kreiranih tabela. Da bi se nova, modifikovana verzija mogla pokrenuti, staru verziju moramo izbrisati naredbom 'drop', ili iskoristiti naredbu koja je napisana u skolu pitanja za zadaću da bi izbrisali sve kreirane tabele.