



# **ZADACA BR 2**

**Osnove baza podataka**

**Ime i prezime:** Nur Osmanbegović

**Br. Indexa:** 17501

**Odsjek:** Računarstvo i informatika

**Godina:** 2018/19



## Sadržaj

Zadatak 1 – Ispravak ERD .....	3
Zadatak 2 – Azuriranje vec kreirane baze podataka.....	4
Zadatak 3 – Punjenje tabela smislenim podacima.....	5
Zadatak 4 – Kreiranje i izvorsavanje upita navedenih u postavci.....	6



## Zadatak 1 – Ispravak ERD

U svom ERD-u sam izmijenila četiri stvari, nijedna izmjena nije bila neophodna po pitanju upita koje sam uradila, te moram drukcije formulirati napravljene izmjene od onoga što je napisano u postavci zadace. U nastavku ću objasniti zašto sam se na te korake odlucila, iako nije bilo neophodno:

1. Izbrisala sam tabelu PRAVNALICA\_UGOVORI i dodala atribut id\_lica (koji predstavlja strani ključ) u entitetu UGOVOR. Razlog je što sam, zbog brzopletosti, pri dizajniranju ERD-a, vezu između entiteta PRAVNA\_LICA i UGOVOR gledala kao vezu N:M i riješila to tabelom veze (tabela PRAVNALICA\_UGOVORI je bila upravo ta tabela veze). Ovo rješenje bi bilo tačno i adekvatno da je veza između pravnih lica i ugovora veza N:M, ali nije. Veza je 1:N, odnosno jedno pravno lice može imati više ugovora, ali jedan ugovor je vezan isključivo za jedno pravno lice, samim tim, ovo nije veza N:M i nema potrebe za tabelom veze. Sve upite bih mogla bez problema napisati (konkretno vezano za ovaj dio) ali bih svaki put morala uključiti jednu tabelu više, kao i jedan dodatni uslov (tabela više povlači i uslov spajanja više). I naravno, tu je bespotrebno zauzimanje memorije tabelom koja je zapravo beskorisna, obzirom da je jedan dodatni atribut u entitetu UGOVOR optimalno rješenje.
2. Druga stvar koju sam uradila je dodavanje primary key atributa u entitet KOLICINE\_U\_SKLADISTIMA. Sasvim je moguće da ovaj atribut neće nikada iskoristiti, i činjenica je da on sam po sebi nije neophodan (moguće da griješim, ali u ovom trenutku stojim iza ovoga), posebno ne za upite koje sam pisala u sklopu zadace. Međutim, ovo sam uradila iz razloga što sam shvatila u međuvremenu da je to konvencija i da bi svaka tabela trebala imati atribut koji omogućava jedinstveno indentificiranje njenih redova, odnosno PRIMARY KEY. Tako da ova izmjena nije bila neophodna, posebno ne za ovu zadacu, ali obzirom na važnost ovog koraka kao i konvencija koja se inače prate pri dizajniranju baza, smatram da je ova izmjena ipak opravdana.
3. Treća izmjena koju sam napravila je dodavanje entiteta GARANCIJA. Mislila sam da je, obzirom da je garancija fiksna, držanje te informacije kao atributa u entitetu Proizvod najispravnije rješenje, međutim, svaki put kada bi kupac želio servisirati neki aparat morala bih računati koliko je vremena prošlo od kupovine do dana reklamacije i vidjeti da li je prošlo vrijeme vazenja garancije. Uvodjenjem ovog entiteta je dosta olakšan rad sa bazom obzirom da su to informacije kojima se često pristupa i koje bi inače trebalo previše puta računati.
4. Četvrta i ujedno posljednja izmjena koju sam napravila je dodavanje atributa DATUM\_ISPORUKE u entitetu ISPORUKA. Razlog je čisto pragmaticne prirode. Pri prvobitnom crtanju ERD-a sam podrazumijevala da je datum isporuke ujedno i datum kupovine odnosno datum fakture. Međutim, to ne mora biti, i vrlo često nije slučaj a podatak o datumu kada je izvršena isporuka je svakako bitan, te smatram da bi bila velika greška ne čuvati tu informaciju.



## Zadatak 2 – Azuriranje vec kreirane baze podataka

Obzirom na mali broj izmjena koje sam napravila u svom dijagramu, nisam imala nikakvih poteskoca za azuriranje baze u skladu s napravljenim izmjenama.

Koristila sam:

DROP TABLE naredbu za brisanje tabele PRAVNALICA\_UGOVORI.

ALTER TABLE za dodavanje stranog kljuca na primarni kljuc (id\_lica) tabele PRAVNA\_LICA u tabeli ugovori.

ALTER TABLE za dodavanje kolone id (koja je ujedno i primarni kljuc te tabele) u tabeli kolicina\_u\_skladistima.

ALTER TABLE za dodavanje kolone datum\_isporuke u tabeli isporuka.



### Zadatak 3 – Punjenje tabela smislenim podacima

Punjenje tabela sam zapocela rucno, u zelji da znam sta se tacno nalazi i da mi bude smisleno preklapanje stranih i primarnih kljuceva. Medjutim, u jednom trenutku mi je ponestalo „inspiracije“ za imena, adrese i rodjendane pa sam pocela koristiti generator: Mockaroo.

Punjenje baze mi je vjerovatno bio najzahtjevniji dio ove zadace. Sva povezivanja se vrse preko id odnosno brojeva, jako je tesko upratiti sta je sta i kako smisleno popuniti.

Sto se tice generatora naprimjer, on generise nasumicne datume, a treba provjeriti da je datum pocetka vazenja garancije prije datuma isteka, da je broj mjeseci izmedju ta dva tuma tacno onoliko kolika je garancija određenog proizvoda.

Trebalo je voditi racuna i da je cijena na fakturi jednaka zbiru cijena pojedinačnih stavki koje cine tu fakturu. A da su cijene na pojedinačnim stavkama jednake umnosku kolicine proizvoda i cijene proizvoda.

Trebalo je voditi racuna pri dodavanju proizvoda da im kategorije zaista odgovaraju. Da su vrijednosti id-eva koje dodjeljujem zauzete.

Takodjer su kategorije trebale odgovarati proizvodu, npr. Da kategorija proizvoda tastatura ne bude audio oprema.

Koliko god da mi je generator pomogao za adrese, rodjendane, imena, bilo je dosta stvari koje sam morala ispraviti da bi zaista smisleno popunjeno bilo.

Bilo bi mnogo lakse da postoje funkcije/procedure koje se mogu pozivati za popunjavanje pojedinih polja koji se mogu dobiti „racunanjem“ iz dostupnih informacija.

Vecina tabela ima 10 redova, osim npr kontinenata, tipova pravnih i fizickih lica, garancija (obzirom da od 10 stavki fakture, nisu svi proizvodi imali garancije), odjela posto je u prvoj zadaci receno da je 5 odjela u firmi.



## Zadatak 4 – Kreiranje i izvršavanje upita navedenih u postavci.

U prilogu se nalaze dvije skripte, jedna koja se može citava izvršiti dok je druga s brojem upita i njegovom postavkom pa tek onda rješanjem. Drugu skriptu je potrebno izvršavati upit po upit.

Upite sam birala prvenstveno da ispunim uslov jednog upita po kategoriji, a zatim sam radila one koji su meni bili intuitivniji, za koje sam bila prilično sigurna da shvatam šta trebaju vratiti (da ne bih pogriješila). Nisam imala nekih većih problema, zanimljivi su mi bili upiti u kojima sam trebala koristiti LEFT/RIGHT OUTER JOIN i SET operator.

Posebno bih izdvojila naredni upit:

7. Prikazati najjeftiniji i najskuplji proizvod po kategorijama, sortirano po nazivu kategorije te po zbiru cijene najskupljeg i najjeftnijeg proizvoda

```
SELECT t1.kategorija NazivKategorije, t1.proizvod "proizvod sa max cijenom", t1.maxcijena  
"cijena max", t2.proizvod "proizvod sa min cijenom", t2.mincijena "cijena min",  
t1.maxcijena+t2.mincijena Zbir
```

FROM

```
(SELECT k.naziv_kategorije kategorija, p.naziv proizvod, p.cijena maxcijena
```

```
FROM proizvodi p, kategorije k
```

```
WHERE p.id_kategorije=k.id_kategorije AND p.cijena=(SELECT max(cijena) FROM  
kategorije k1, proizvodi p1 WHERE p1.id_kategorije=k1.id_kategorije AND  
k.id_kategorije=k1.id_kategorije)) t1,
```

```
(SELECT k2.naziv_kategorije kate, p.naziv proi, p.cijena mincijena
```

```
FROM proizvodi p, kategorije k2
```

```
WHERE p.id_kategorije=k2.id_kategorije AND p.cijena=(SELECT min(cijena) FROM  
kategorije k1, proizvodi p1 WHERE p1.id_kategorije=k1.id_kategorije AND  
k2.id_kategorije=k1.id_kategorije)) t2
```

```
WHERE t1.kategorija=t2.kate
```

```
ORDER BY NazivKategorije, Zbir DESC;
```

Prvo sam uradila ovaj upit na način da sam vraćala za svaku kategoriju maksimalnu i minimalnu cijenu proizvoda, jer na prvu mi nije naumpalo kako bih mogla vratiti u jednom redu i naziv proizvoda i proizvod sa maksimalnom i minimalnom cijenom. Dugo sam radila ovaj upit ali sam sigurna najviše naučila radeci ga. Prvi upit koji sam napisala je vraćao kategoriju, najveću i najmanju cijenu proizvoda u kategoriji i zbir tih cijena. Drugi upit koji sam napisala je vraćao



po kategoriji proizvode s njihovom maksimalnom i minimalnom cijenom. Na kraju sam dosla na ideju da napravim dvije tabele, jednu koja vraća proizvod sa najvećom cijenom, drugu koja vraća proizvod sa najmanjom cijenom po kategoriji. Da spojim te tabele izbjegavajući dekartov proizvod njihovim spajanjem po id\_kategorije.

Posebno mi je zanimljiv upit bio

10. Izlistati top 10 proizvoda koji imaju najveći popust.

```
SELECT *
```

```
FROM ( SELECT p.naziv Proizvod, p.cijena Cijena, p.cijena*o.popust*0.01 "Cijena sa  
popustom", s.naziv_skladista Naziv
```

```
FROM proizvodi p, popust o, skladista s, kolicina_u_skladistima k
```

```
WHERE p.id_proizvoda=k.id_proizvoda AND k.id_skladista=s.id_skladista AND  
p.id_proizvoda=p.id_proizvoda
```

```
ORDER BY o.popust) t
```

```
WHERE rownum<11;
```

U početku nisam bila sigurna kako napisati ovaj upit – jedino sporno je bilo ograničenje na top 10 proizvoda, našla sam više rješenja na internetu ali jedino koje je izbacivalo ono što se u upitu traži je ono priloženo u skripti.