

MATEMATIČKI FAKULTET

PROJEKAT IZ PREDMETA INFORMACIONI SISTEMI

ŠKOLSKA 2018/2019

Informacioni sistem poslovanja veleprodaje apoteke

Autori:

Nikola Stojević 1111/2018

Profesor:

Dr. Saša Malkov

Asistent:

Ivana Tanasijević

December 29, 2018



Sadržaj

1	Opis	2
1.1	Relacioni model	2
1.2	Uslovi	3
2	Upiti	5
2.1	PROGRAM 1.c	5
2.2	PROGRAM 2.c	5

1 Opis

Projekat se bavi sektorom prodaje u okviru poslovanje neke veleprodaje. Zaposleni u prodaje može biti komercijalista i menadžer. Komercijalista evidentira kupce i u kontaktu sa njima prima porudžbinu, recimo putem imejla. Tabela Kupac predstavlja pravno lice, dok tabela Kontakt osobe, predstavlja fizička lica sa kojima poslujemo u okviru određene firme.

Glavni deo baza predstavlja odnos specijalizavije tabele Dokument na Porudžbina, Ponuda, Otpremnica i Faktura. Dokument svaki ima posebne attribute, kao i svoje stavke, koje daju informacije o proizvodu i potrebnoj količini istog.

Porudžbinu evidentira komercijalista u kontaktu sa kupcem. Zatim komercijalista sastavlja ponudu kupcu, uz eventualno prisustvo menadžera, koji može dati popust određeni, radi uspešnijeg nastavka saradnje. Tabela Izrađuje čuva podatke o zaposlenima koji učestvuju u sastavljanju određenog dokumenta. Ponuda sadrži za razliku od porudžbine sadrži cene, i detalje posla, kao što su način isporuke, rok isporuke, rok plaćanja, opciono kontinuitet isporuke (u sličaku da može roba iz više puta da se dostavi).

Ponuda može imati više otpremnica, u sličaju da je postignut dogovor oko kontinuiteta isporuke ili trenutno samo toliko možemo isporučiti. Otpremnica ne sadrži cene i ona se šalje magacinu za odvajanje robe.

Na osnovu otpremnice se izrađuje faktura, koja naplaćuje kupcu robu koja će mu biti isporučena na osnovu otpremnice.

Rekurzivni odnos se ostvaruje preko tabele Proizvod, jer postoji proizvodi koji se prodaju, a sastoje se od postojećih proizvoda. Recimo pakovanja od vise komada, promocije, ili paketi lekova koji se često zajedno naručuju.

1.1 Relacioni model

- Zaposleni (idZaposleni*, ImePrezime, DatumZaposlenja, StručnaSprema, Fukcija, UgovorenaPlata)
- Kupac (idKupac*, PIB, NazivFirme, TekuciRačun)
- Kontakt osoba (idKontOsobe*, idKupac*, ImePrezime, Telefon)
- Dokument (idDokument*, VremeKreiranja, VremeIzmene, Status)
- Izrađuje (Dokument_idDokument*, Zaposleni_idZaposleni*)

- Porudžbina (idPorudzbina*, Kupac_idKupac, DatumPorudzbine)
- Proizvod (idProizvod*, Naziv, Cena)
- Proizvod_has_Proizvod (idProizvod*, idProizvod1*)
- Stavka (idDokumenta*, idProizvod*, Kolicina)
- Ponuda (idPonuda*, idPorudžbina, DatumPonude, Cena, Popust, ProdajnaCena, Kontinuitet i broj isporuka, Način isporuke, Rok plaćanja)
- Otpremnica (idOtpremnica*, idMagacina,Ponuda_idPonuda)
- Faktura (idFaktura*, idOtpremnica, Rok isporuke)

1.2 Uslovi

- Nezavisni entiteti: Zaposleni, Kupac, Porudžbina, Ponuda, Otpremnica, Faktura
- Agregirani entiteti: Izradjuje
- Rekurzivni odnos: Proizvod_has_Proizvod
- Slabi entitet: Kontakt_osoba
- Generalizacija/Specijalizacija: Dokument -> Porudzbina,Ponuda,Otpremnica,Faktura

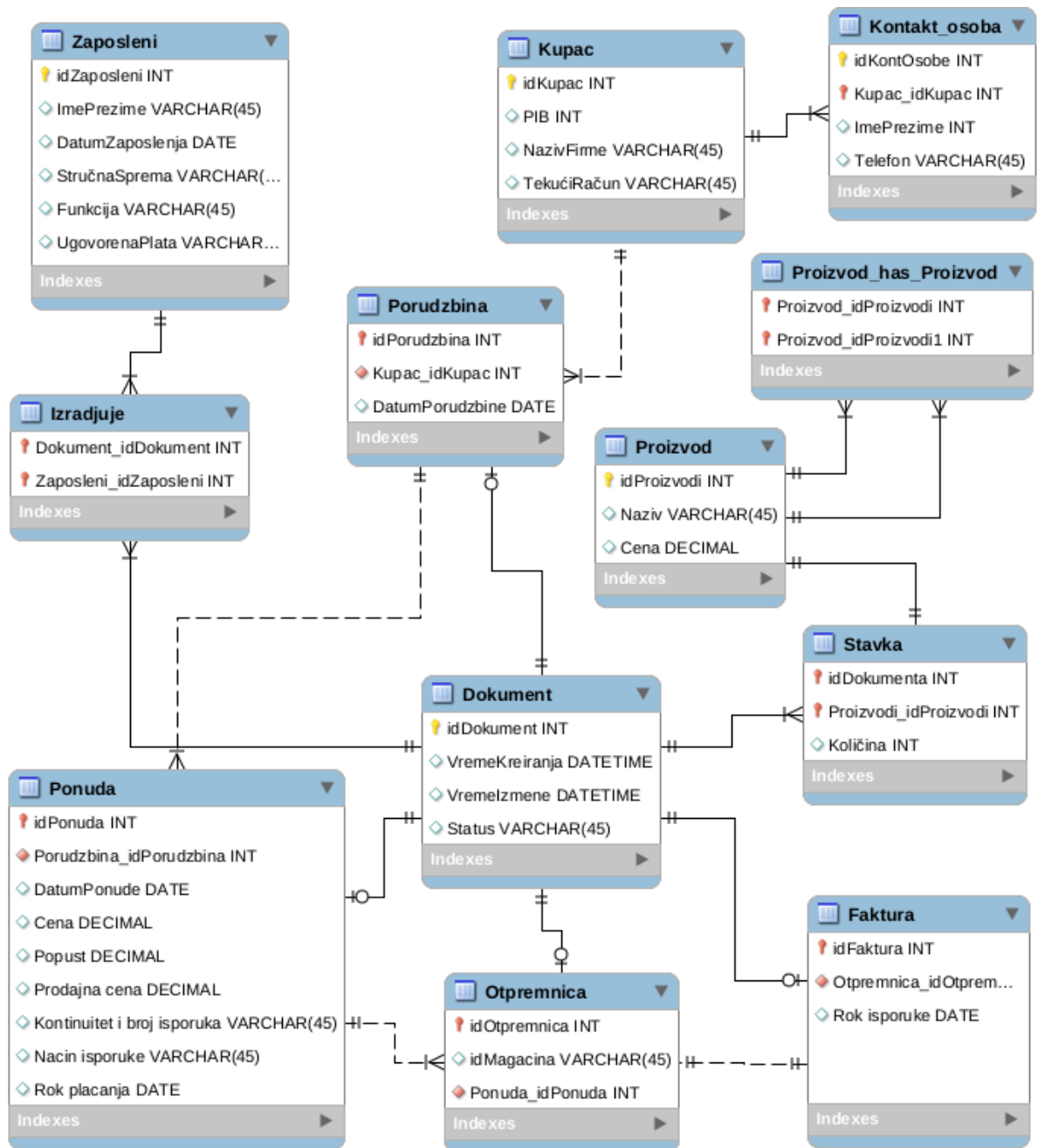


Figure 1: EER dijagram

2 Upiti

2.1 PROGRAM 1.c

Sadrži sledeće upite za prikaz stanja:

```
select * from Porudzbina;  
select * from Dokument;  
select * from Izradjuje;
```

Dok za menjanje stanja:

```
insert into Porudzbina(Kupac_idKupac, DatumPorudzbine) values  
(idKupca,Datum)
```

Ovo povlači trigger koji pronalazi odgovarajući id za unetu Porudzbinu, popunjava roditeljsku tabelu Dokument i tabelu Izradjuje, koja nam govori koji korisnik korisnik je učestvovao u izradi određenog dokumenta.

Nakon toga se preko nove konekcije sa drugim korisnikom (pera, ranije nikola korisnik) vrši naredba ažuriranja datuma pristizanja porudzbine.

```
update Porudzbina set DatumPorudzbine = datum where idPorudzbina  
= idPorudzbine);
```

Povlači se trigger koji u tabeli Dokument menja korisnika koji je izvršio poslednju izmenu, dodaje u tabelu Izradjuje novog korisnika.

2.2 PROGRAM 2.c

Sadrži sledeće upite za prikaz stanja:

```
select * from Proizvod;  
select * from Stavka;  
select idPonuda, Porudzbina_idPorudzbina, Cena, Popust, Prodajna_cena from  
Ponuda;
```

Dok za menjanje stanja:

```
insert into Stavka(idDokumenta, Proizvodi_idProizvodi, Kolicina)  
values (idDokumenta, idProizvoda, kolicina);
```

Ovo povlači trigger koji računa cenu te stavke na osnovu količine i cene proizvoda. Dalje trigger ažurira odgovarajući red u tabeli Ponuda, formira cenu na osnovu zbira svih stavki, i formira prodajnu cenu sa uračunatim popustom.

Pored ove tu je i naredba:

```
UPDATE Ponuda SET Popust = NoviPopust WHERE idPonuda =
```

idPonuda;

Vrši ažuriranje popusta i poziva se trigger koji ažurira prodajnu cenu na osnovu novog popusta.