

PROJEKAT

Predmet: Baze podataka

Tema: Limundo online aukcijska kuća

Student: Ružica Milojević 2017004008

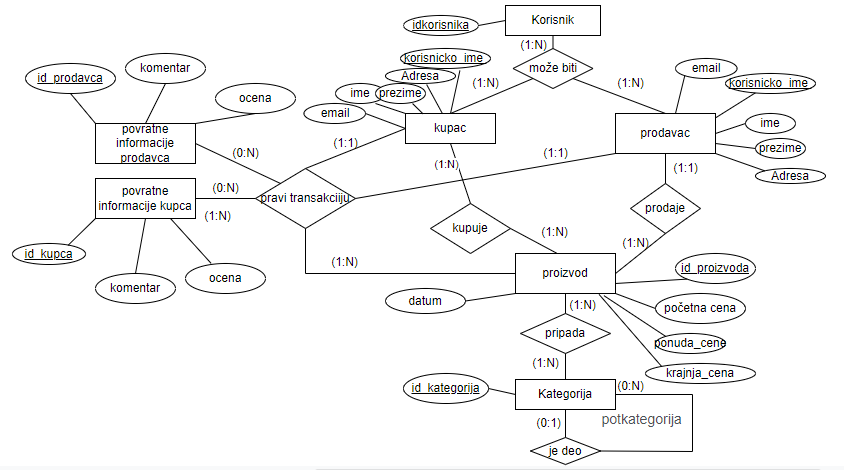
Profesor: dr Danijela Tešendić

Novi Sad, 2022

Sadržaj

1. Opis informacionog sistema
2. Projektovanje baze podataka
   1. ER model
   2. Relacioni model
3. Naredbe za kreiranje šeme baze podataka
4. Unos podataka
5. Upiti nad podacima
6. Opis informacionog sistema

Informacioni sistem koji prikazuje kakva je struktura izrade digitalne baze podataka za online sajt koji radi na aukcijskom sistemu, gde korisnici mogu da koriste sistem kupovine proizvoda ,kao i postavljanje proizvoda za prodaju na web sajtu, da budu prodavci ili kupci.Prikazuje korisnika i koje bitne informacije se uzimaju za transakciju kupovine,Postavlja se sistem ponude cene za neki proizvod. Sistemom se korisnik služi da bi pobedio u aukciji I sproveo transakciju kupovine nekog proizvoda koji želi.

1. Projektovanje baze podataka
   1. ER model

**Poveznik korisnik-kupac**

Korisnik može biti kupac 1 ili više puta.Kupac može biti jedan ili više korisnika.

**Poveznik korisnik-prodavac**

Korisnik može biti prodavac 1 Ili više puta,a prodavac može biti jedan ili više korisnika.

**Poveznik kupac-proizvod**

Kupac može da kupuje proizvod 1 Ili više puta,a proizvod može da bude kupljen od 1 ili više korisnika.

**Poveznik prodavac-proizvod**

Prodavac prodaje bar 1 proizvod ili više proizvoda.Proizvod može da bude postavljen od strane jednog prodavca.

**Poveznik proizvod-kategorija**

Proizvod može da pripada jednoj ili više kategoriji.Kategorija može da ima 1 ili više proizvoda u njoj.

**Poveznik kategorija**

Jedna kategorija može biti potkategorija maksimalno 1 kategorije,minimalno ni jedne.

Glavna kategorija može da ima više potkategorija,a može i da ih nema.

**Poveznik prodavac-povratne informacije prodavca.**

Prodavac može da ostavi neke povratne informacije više puta a ne mora da ih ostavi.

Ostavljene informacije daje jedan I samo jedan prodavac.

**Poveznik kupac-povratne informacije kupac**

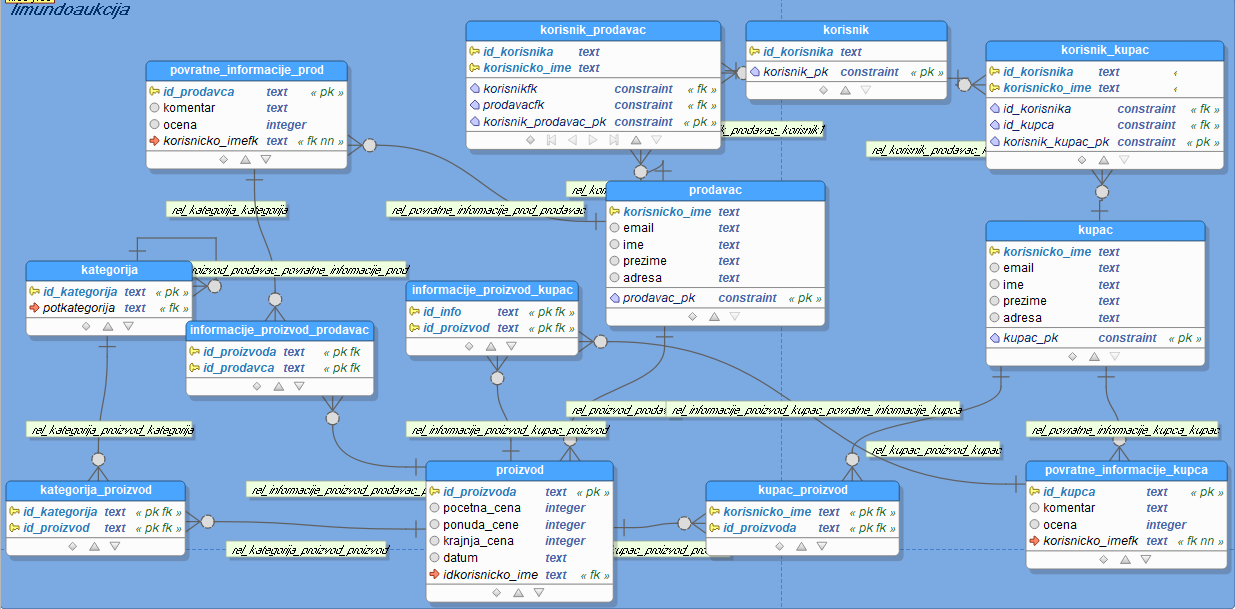
Kupac može da ostavi neke povratne informacije više puta a ne mora da ih ostavi.

Ostavljene informacije daje jedan I samo jedan kupac.

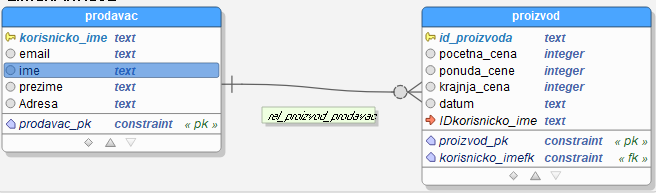
**Poveznik povratne informacije kupca,prodavca I proizvoda.**

Povratne informacije dobijaju 1 ili više proizvoda.Jedan proizvod može da ne dobije nikakve povratne informacije a može da dobije I više.

* 1. Relacioni model

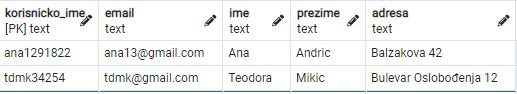


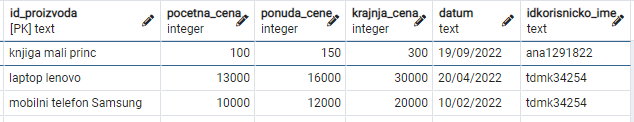
Prevođenje er dijagrama u relacioni.

Prodavac-Proizvod 

Primarni ključ iz prodavca(iz strane više prelazi u stranu manje) tj “korisničko\_ime” prećiće u proizvod kao strani ključ.Minimalni kardinaliteti su stavljeni na 1 što znači da jedan prodavac mora da prodaje bar jedan proizvod I da može da prodaje više ,i da jedan proizvod prodaje tačno jedan prodavac(minimalno jedan I maksimalno jedan).

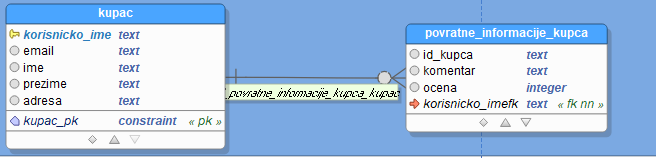
Vrednosti stranog ključa će moći da se ponavljaju što znači da ćemo u tabeli imati izložene različite proizvode koje može da prodaje isti prodavac.





U ovom primeru tabele imamo 2 proizvoda :laptop,mobilni telefon koji prodaje prodavac sa korisničkim imenom tdmk34254 odnosno što piše u tabeli prodavca da je to Teodora Mikic, a da knjigu prodaje drugi prodavac sa korisničkim imenom ana1291822 tj Ana Andrić.

**Kupac-povratne informacije**

****

Veza jedan prema više.

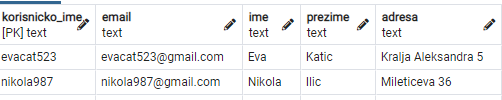
Ako gledamo 1 kupca on može da ostavi više povratnih informacija(komentara ocena I utisaka) a za 1 povratnu informaciju moramo da imamo tačno jednog kupca koji je ostavio svoje utiske.

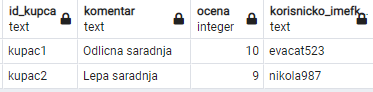
Primarni ključ ”korisničko ime” iz kupca prelazi u povratne informacije i tamo postaje strani ključ.

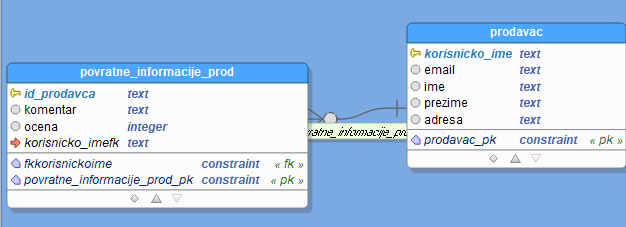
Stavićemo da je strani ključ “korisničko ime” u povratnim informacija not null što znači da će uvek biti vrednosti u toj koloni I da imamo informacije.

Pošto je veza jedan prema više neće biti unique ograničenja.

Isto tako će da se prevedu entiteti prodavac I povratne informacije prodavca

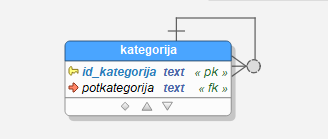




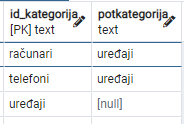


Rekurzivni tip poveznika Kategorije

Nastaće strani ključ na osnovu primarnog ključa kategorije.Znači da ćemo u tabeli dodati kolonu I nazvati je potkategorija koji je strani ključ od glavne kategorije.Glavna kategorija je oznaka za onu kategoriju koja može da ima više potkategorija.



UPDATE limundokuca4.kategorija SET potkategorija = 'uređaji' WHERE id\_kategorija = 'računari' popunjavamo tabelu kategorije I njene kolone,jer kada koristimo insert into on popunjava samo primarne ključeve,jer ako kao u ovom slučaju želim da kažem da je računar potkategorija uređaja, on neće hteti da se popuni jer prethodno nije definisan računar sam,ili ako jeste, izbaciće da ga registruje samo kao primarni ključ,tj na početku ne mogu svi podaci da se popune zajedno,zato moram da koristim update da bi popunila tabelu.



Računari je potkategorija kategorije koji ima primarni ključ “uređaji”

Kao I telefoni mogu biti potkategorija kategoriji telefoni.

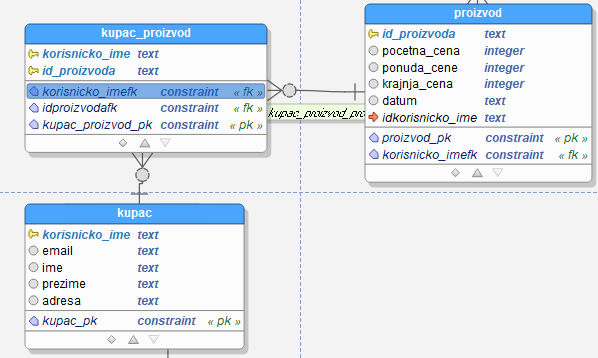
Uređaji ne moraju da pripadaju ni jednoj potkategoriji,zato može biti null.

**Veze više prema više**

prevodi se kao posebna presečna tabela.2 tipa entiteta postaju tabela,

.Njihovi primarni ključevi id\_proizvoda I korisničko\_ime prelaze u presečnu tabelu,tamo postaju strani ključevi i postaju primarni ključevi te tabele.

Nastala je tabela kupac\_proizvod od tabela proizvod I kupac.



Na osnovu veza više prema više nastale su presečne tabele za povratne informacije kupca,povratne informacije prodavca I proizvoda.

Korisnik I kupac,prodavac dobijaju svoje presečne tabele jer je njima veza isto 1:N.

1. Naredbe za kreiranje šeme baze podataka

-- object: limundoaukcija | type: SCHEMA --

-- DROP SCHEMA IF EXISTS limundoaukcija CASCADE;

CREATE SCHEMA limundoaukcija;

-- ddl-end --

- Kaskada koristi se da se podređeni podaci ili brišu ili ažuraraju.Oni su postavljeni na svoje podrazumevane vrednosti.

Pravi se objekat limundoaukcija ,stavi šemu, ako postoji ona će napraviti šemu pod nazivom limundoaukcija.

CREATE TABLE limundoaukcija.prodavac (

korisnicko\_ime text NOT NULL,

email text,

ime text,

prezime text,

adresa text,

CONSTRAINT prodavac\_pk PRIMARY KEY (korisnicko\_ime)

);

Pozivom imena šeme(ako je definisano u ovom slučaju jeste) kreira tabelu prodavac,njene kolone(korisnicko\_ime) tipa text NOT NULL-“želi pripadnost neke vrednost” u ovom slučaju strani ključ koji je definisan kao ograničenje prodavac\_pk

ALTER TABLE limundoaukcija.proizvod ADD CONSTRAINT korisnicko\_imefk FOREIGN KEY (idkorisnicko\_ime)

REFERENCES limundoaukcija.prodavac (korisnicko\_ime) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE limundoaukcija.povratne\_informacije\_kupca ADD CONSTRAINT fkkorisnicko\_ime FOREIGN KEY (korisnicko\_imefk)

REFERENCES limundoaukcija.kupac (korisnicko\_ime) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

Alternativa za 2 tabele koje imaju vezu više prema više. Stavlja je kao strani ključ jer je presečna tabela 2 tabele i primarne ključeve svoje tabele nove tabele.

1. Unos podataka

INSERT INTO:

Izabrana je tabela sa kupcima koja treba da prikaze informacije istih koji su napravili nalog.

Primer komande napisane u query toolu:

INSERT INTO limundoaukcija.kupac(korisnicko\_ime,email,ime,prezime,adresa)

VALUES ('doramikic13','doramik@gmail.com','Teodora ','Mikic','Bulevar Kralja Petra 13');

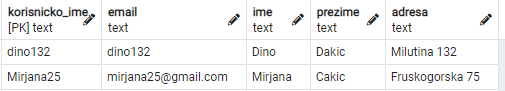
Čime je naredjeno da se preko kreirane šeme pozove tabela kupac u čijim kolonama će se dodati vrednosti.

Rezultat nakon unosa više podataka:



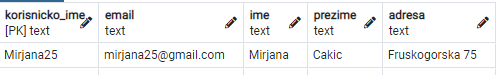
UPDATE I DELETE:

Da bi izmenili neke specifčne podatke, na primer napravljena je greška kod Dina I da bi to ispravili slediće:



Treba da se ispod kolone email napise licni email Dina.

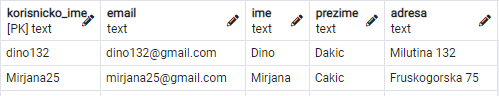
Ali pre toga mora da se obriše iz prodavca njegovo ime, jer su te tabele povezane preko stranog ključa email.

delete from limundoaukcija.prodavac where korisnicko\_ime='dino132' 

Jer da to nismo obrisali ne bi nam radio sledeći kod:

update limundoaukcija.prodavac set korisnicko\_ime='dino132'

where email='dino132@gmail.com'

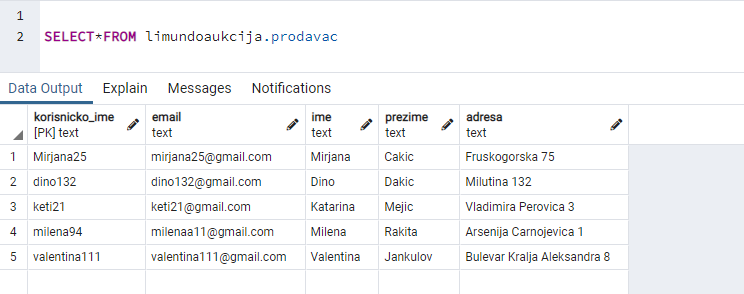


1. Upiti nad podacima

Prikazati primere sledećih upita:

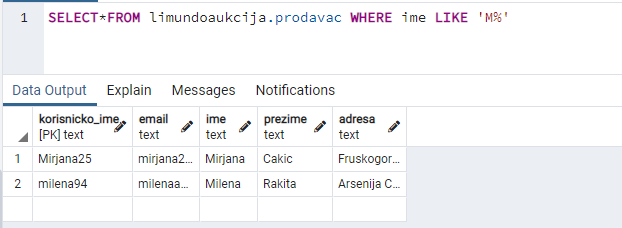
1. Prikaz svih podataka iz tabele

Da bi prikazala sve prodavce sledi:

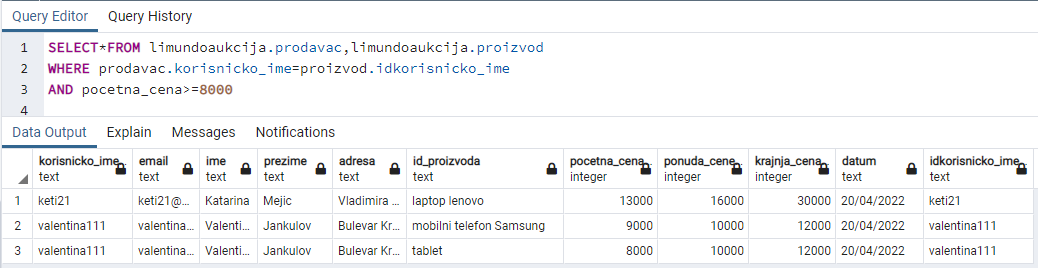


1. Filtriranje podataka iz jedne tabele na osnovu nekog kriterijuma

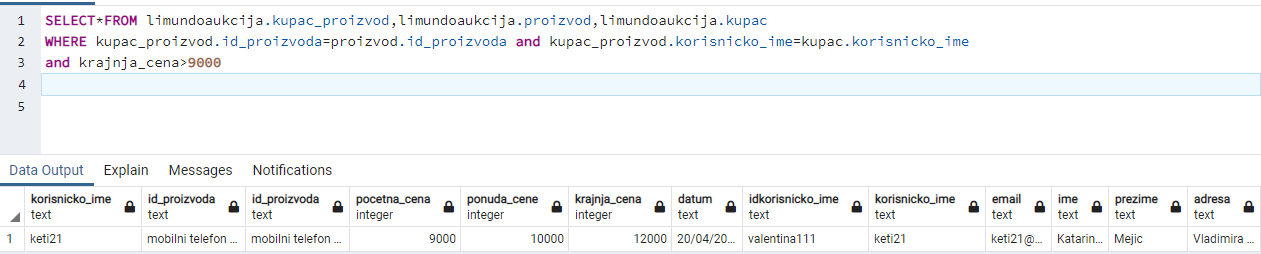
\*Filtriranje korisnickih imena svih prodavaca kojima ime počinje na veliko slovo M.



1. Upit sa spajanjem dve tabele i uslovom

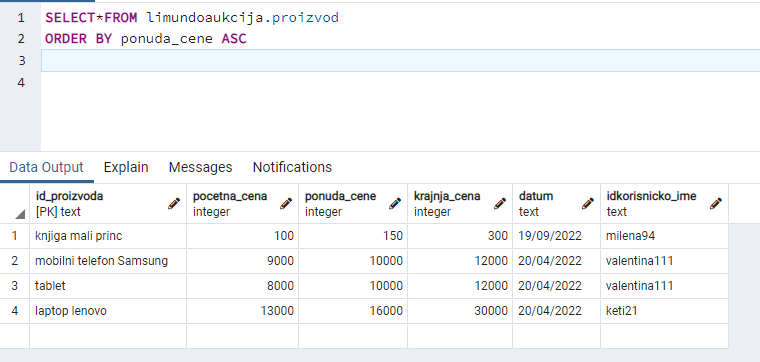


1. Upit sa spajanjem tri tabele i uslovom



1. Group by upit sa spajanjem dve tabele i primenom skupovne funkcije

Filtrirao je pocetnu cenu tako da prvo idu manje vrednosti ponude cene



1. Ugnježdeni upit

Za svaku ponudu cene proverava da li je veći od svih vrednosti kada se izvrši upit. 