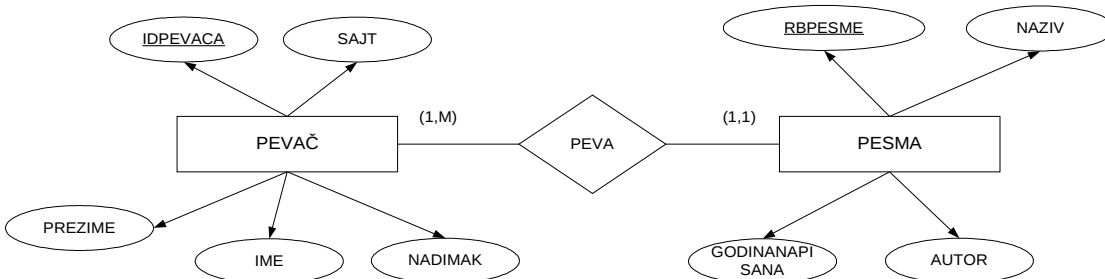


1. Napisati MySQL skript za kreiranje baze podataka pod nazivom PESME, sa strukturom tabela koja odgovara relacionom modelu podataka koji se dobija prevođenjem datog EER dijagrama u relacioni model. Prilikom definisanja ograničenja spoljnih ključeva treba definisati akcije referencijalnog integriteta u slučaju izmene i brisanja podataka u povezanim tabelama restriktivno.



Domeni atributa:

Naziv atributa	Tip podatka	Ograničenje	Obavezn a vrednost
idpevaca	Ceo broj	Većeg opsega	da
ime	String	Max. 30 karaktera	da
prezime	String	Max. 30 karaktera	da
nadimak	String	Max. 20 karaktera	ne
rbpesme	Automatski brojač	1, 2, 3, ..., n	da
naziv	String	Max. 50 karaktera	da
autor	String	Max. 50 karaktera	ne
sajt	String	Max. 200 karaktera	ne
godinanapisana	Ceo broj	Vrednost mora biti veća od 1900 Inicijalna vrednost je 2011	da

Napisati SQL upite za ažuriranje baze podataka PESME:

a) Upisati nove podatke u bazu: Pesmu „The Road To Hell“ je napisao 1989. godine Chris Rea, koji je istovremeno i pevač.

b) Promeniti podatke o sajtu pevača Chris Rea. Web adresa sajta je: „<http://www.chrisrea.org>“

c) Obrisati podatke iz baze za pesmu „The Road The Hell“ i pevača Chris Rea.

Prilikom pisanja SQL upita smatrati da baza podataka još ne sadrži podatke o pesmama i obratiti pažnju na redosled kojim bi se izvršavali upiti!

REŠENJE:

```
CREATE DATABASE IF NOT EXISTS pesme;
```

```
USE pesme;
```

```
CREATE TABLE IF NOT EXISTS pevac
```

```
(
  idpevaca bigint NOT NULL primary key,
  prezime varchar(30) NOT NULL,
  ime varchar(30) NOT NULL,
  nadimak varchar(20) NULL,
  sajt varchar(200) NULL
);
```

```
CREATE TABLE pesma
```

```
(
  rbpesme bigint NOT NULL primary key AUTO_INCREMENT,
  autor varchar(50) NULL,
  naziv varchar(50) NOT NULL,
  godinanapisana int NOT NULL DEFAULT 2011 ,
  idpevaca bigint NOT NULL,
  CONSTRAINT fk_peva FOREIGN KEY (idpevaca) REFERENCES pevac (idpevaca)
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
);
```

```
ALTER TABLE pesma
```

```
ADD CONSTRAINT ck_pesma CHECK (godinanapisana>1900);
```

Kako MySQL ne podrzava CHECK opciju, neophodno je isprogramirati trigger:

```
#POCETAK TRIGERA
```

```
DELIMITER $$
```

```
CREATE TRIGGER provera_godine BEFORE INSERT ON pesma
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE msg VARCHAR(255);
```

```
IF (NEW.godinanapisana<1900)
```

```
THEN
```

```
SET msg='Greska: godina kada je pesma napisana mora biti veca od 1900!';
```

```
SIGNAL sqlstate '45000' SET message_text= msg;
```

```
END IF ;
```

```
END $$
```

```
DELIMITER ;
```

```
#KRAJ TRIGERA
```

USE pesme;

a) INSERT INTO pevac
VALUES (1,'Rea', 'Chris', null, null);

INSERT INTO pesma
VALUES (null,'Chris Rea', 'The Road To Hell', 1989, 1);
#Napomena: Probati sa godinom manjom od 1900!

b) UPDATE pevac
SET sajt='http://www.chrisrea.org'
WHERE prezime='Rea' and ime='Chris';

c) DELETE FROM pesma
WHERE naziv='The Road To Hell' and idpevaca=1;

DELETE FROM pevac
WHERE prezime='Rea' and ime='Chris' and idpevaca=1;

2. Kreirati, uz pomoć MySQL skripta, bazu podataka pod nazivom RACUNARI,

sa strukturom tabela koja odgovara relacionom modelu podataka koji se dobija prevođenjem datog EER dijagrama u relacioni model podataka. Kolone IDKUPCA i IDKONFIGURACIJE su auto inkrement (vrednosti 1, 2, 3, ..., n) koje imaju početnu vrednost 1 i isti takav inkrement. Prilikom uspostavljanja ograničenja spoljnih ključeva definisati akcije referencijalnog integriteta u slučaju izmene i brisanja podataka u povezanim tabelama kaskadno. Inicijalna vrednost za obeležje BOJA je 'siva'. Za kolonu Cena postaviti ograničenje tako da vrednost mora biti veća od 0. Za kolonu Postanski_broj obezbediti da se unose samo vrednosti u obliku petocifrenog broja (pri čemu prva cifra mora biti veća od nule). U bazi podataka ne mogu postojati dve različite konfiguracije sa istim nazivom. Sve kolone su obavezna polja i moraju imati neku vrednost, osim popusta konfiguracije i telefona kupca. Domeni ostalih obeležja: IDKUPCA, IDKONFIGURACIJE - ceo broj većeg opsega; PREZIME, IME, ADRESA, POSTANSKI_BROJ, BOJA, NAZIV, TELEFON - string/tekst maksimalne dužine 30 karaktera; DATUMKUPOVINE - datum; CENA,POPUST - ceo broj manjeg opsega.

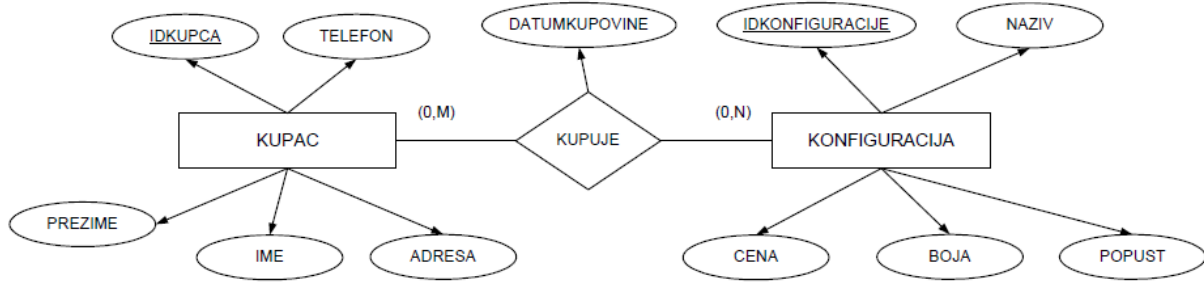
Ažurirati, uz pomoć SQL upita, bazu podataka RACUNARI. Smatrati da je baza podataka prazna, tj. da ne sadrži podatke.

a) Dodati sledeće podatke: kupac - osoba MARKO MARKOVIC, sa adresom Đure Đakovića bb u Zrenjaninu, čiji je poštanski broj 23000 je kupila Laptop računar T-200 firme Toshiba, crne boje, 20.01.2008. godine, po ceni od 100.000 dinara.

b) Prilikom prodaje konfiguracija primeniti popust od 10% za sve konfiguracije čija je cena veća od 90.000 dinara. Povećati cenu za 10% svim konfiguracijama čija je boja 'bela'.

c) Obrisati podatke o kupcu MARKO MARKOVIC, konfiguraciji i kupovini Laptop računara T-200.

Prilikom pisanja DML upita obratiti pažnju na redosled izvršavanja upita!



```
CREATE DATABASE IF NOT EXISTS racunari;
```

```
USE racunari;
```

```
CREATE TABLE kupac
```

```
(
  idkupca bigint NOT NULL PRIMARY KEY AUTO_INCREMENT,
  ime varchar(30) NOT NULL,
  prezime varchar(30) NOT NULL,
  adresa varchar(30) NOT NULL,
  postanski_broj varchar(30) NOT NULL,
  telefon varchar(30) NULL
);
```

```
CREATE TABLE konfiguracija
```

```
(
  idkonfiguracije bigint NOT NULL PRIMARY KEY AUTO_INCREMENT,
  naziv varchar(30) NOT NULL,
  boja varchar(30) NOT NULL DEFAULT 'siva',
  cena integer NOT NULL,
  popust tinyint NULL,
  constraint ak_konf unique (naziv)
);
```

```
CREATE TABLE kupuje
```

```
(
  idkonfiguracije bigint NOT NULL,
  idkupca bigint NOT NULL,
  datumkupovine date NOT NULL,
  CONSTRAINT PK_kupuje PRIMARY KEY (idkonfiguracije, idkupca),
  CONSTRAINT
  FK_kupuje_konfiguracija FOREIGN KEY (idkonfiguracije) REFERENCES konfiguracija
  (idkonfiguracije)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  CONSTRAINT FK_kupuje_kupac FOREIGN KEY (idkupca) REFERENCES kupac (idkupca)
  ON UPDATE CASCADE
  ON DELETE CASCADE);
```

```
#ALTER TABLE konfiguracija
```

```
#add constraint ck_konf check (cena>0);
```

```
#ALTER TABLE kupac
```

```
#add constraint ck_kupac check (postanski_broj rlike '[1-9][0-9][0-9][0-9][0-9]');
```

Umesto check ogranicenja pisemo trigere:

```
#POCETAK TRIGERA
DELIMITER $$
CREATE TRIGGER provera_konf_cena BEFORE INSERT ON konfiguracija
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
  IF (NEW.cena<=0)
THEN
SET msg='Greska: Cena mora biti veca od nule!';
  SIGNAL sqlstate '45000' SET message_text= msg;
END IF ;
END $$
DELIMITER ;
#KRAJ TRIGERA
```

```
#POCETAK TRIGERA
DELIMITER $$
CREATE TRIGGER kupac_postanski_broj BEFORE INSERT ON kupac
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
  IF (NEW.postanski_broj not rlike '[1-9][0-9][0-9][0-9][0-9]')
THEN
SET msg='Greska: Postanski broj nije u odgovarajucem formatu!';
  SIGNAL sqlstate '45000' SET message_text= msg;
END IF ;
END $$
DELIMITER ;
#KRAJ TRIGERA
```

```
use racunari;
```

a) insert into kupac values (null,'Markovic', 'Marko', 'Đure Đakovića bb - Zrenjanin', '23000', null) ;

```
insert into konfiguracija
```

```
values (null,'Laptop računar Toshiba T-200', 'crna', 100000, null) ;
```

```
insert into kupuje values (1, 1, '2008-01-20') ;
```

b)

```
update konfiguracija
```

```
set popust=10
```

```
where cena>90000;
```

```
update konfiguracija
```

```
set cena=cena*0.1
```

```
where boja='bela' ;
```

c)

```
delete from kupuje
```

```
where (idkupca=1 and idkonfiguracije=1 )
```

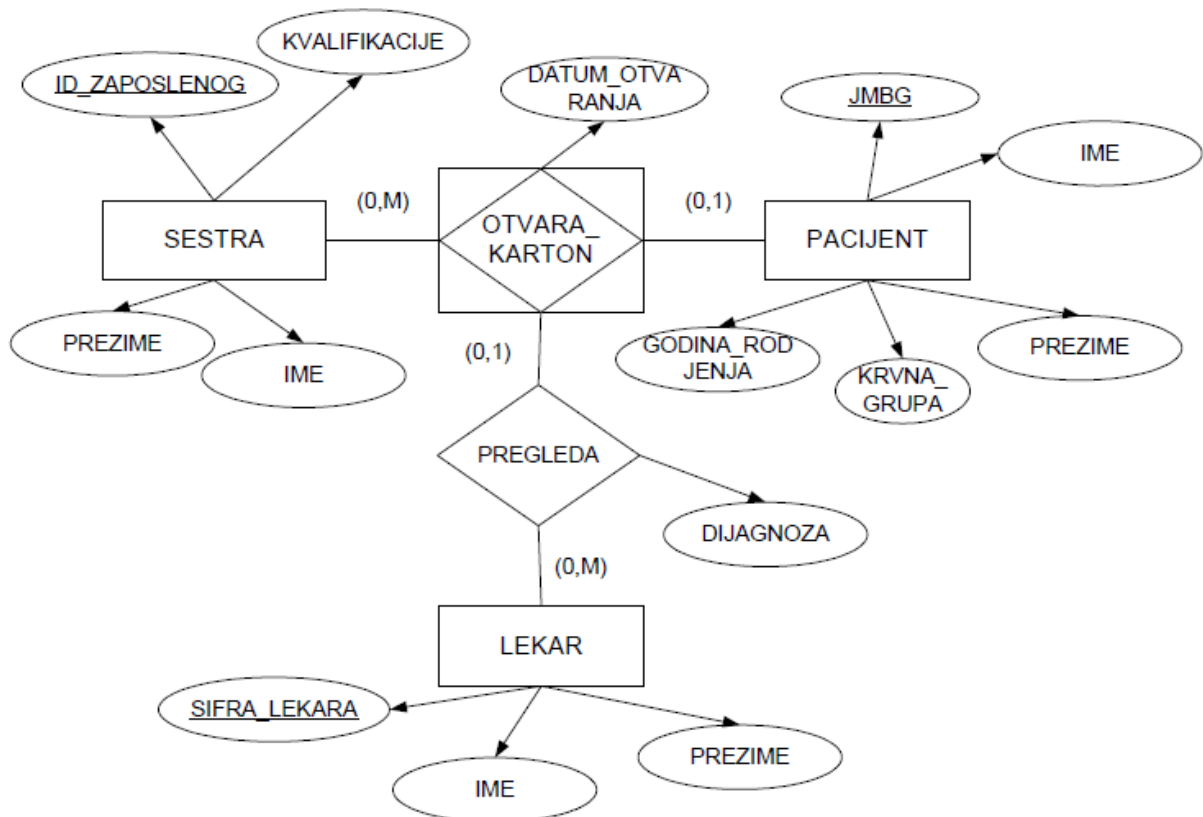
delete from kupac
where prezime='Markovic' and ime='Marko'

delete from konfiguracija
where naziv='Laptop računar Toshiba T-200'

3. Napisati MySQL skript za kreiranje baze podataka pod nazivom ORDINACIJA,

sa strukturom tabela koja odgovara relacionom modelu podataka koji se dobija prevođenjem datog EER dijagrama u relacioni model. Ograničenja spoljnih ključeva formirati uz ograničenja na nivou polja i tom prilikom treba definisati akcije referencijalnog integriteta u slučaju izmene i brisanja podataka u povezanim tabelama restriktivno.

Obeležje ID_ZAPOSLENOG ima početnu vrednost 1 i isti takav inkrement (1,2,3,...,n). Inicijalna vrednost za obeležje KRVNA_GRUPA je 'O+'. Za kolonu GODINA_RODJENJA postaviti ograničenje tako da vrednost mora biti veća od 1900. Sve kolone su obavezna polja i moraju imati neku vrednost, osim kvalifikacije sestree i datuma otvaranja kartona. Domeni ostalih obeležja: PREZIME, IME, KRVNA_GRUPA, KVALIFIKACIJE - string/tekst maksimalne dužine 30 karaktera; DIJAGNOZA - string/tekst maksimalne dužine 250 karaktera; DATUM_OTVARANJA - datum; GODINA_RODJENJA - ceo broj; SIFRA_LEKARA - ceo broj većeg opsega, JMBG - fiksno 13 karaktera.



Napisati SQL upite za ažuriranje baze podataka ako ORDINACIJA. Smatrati da je baza podataka prazna, tj. da tabele ne sadrže podatke.

- a)** Dodati sledeće podatke: osobi M arko M arković , JM BG: 1203965850033, rođen 1965., krvna grupa AB-, je sestra Mirković Mina, viša medicinska sestra 25.1.2009. godine otvorila karton. Na pregledu je lekar 1 – Đorđe Šarac uspostavio dijagnozu b.o.
- b)** Promeniti dijagnozu Marku Markoviću na tahikardija.
- c)** Obrisati sve podatke o pac ijentu Marku Marković u i pregledu. Podatke o lekaru i medicinskoj sestri ne treba brisati iz baze podataka.

```
CREATE DATABASE IF NOT EXISTS ordinacija ;
USE ordinacija;
```

```
CREATE TABLE SESTRA
(
id_zaposlenog int NOT NULL PRIMARY KEY AUTO_INCREMENT,
ime varchar(30) NOT NULL,
prezime varchar(30) NOT NULL,
kvalifikacije varchar(30) NULL
);
```

```
CREATE TABLE IF NOT EXISTS PACIJENT
(
jmbg varchar(13) NOT NULL PRIMARY KEY,
ime varchar(30) NOT NULL,
prezime varchar(30) NOT NULL,
krvna_grupa VARCHAR(32) NOT NULL DEFAULT '0+',
godina_rodjenja int NOT NULL
);
```

```
CREATE TABLE OTVARA_KARTON
(
jmbg varchar(13) NOT NULL PRIMARY KEY,
id_zaposlenog int NOT NULL,
datum_otvaranja date NULL,
CONSTRAINT
FK_otvara_sestra FOREIGN KEY (id_zaposlenog) REFERENCES sestra (id_zaposlenog)
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT
FK_otvara_pacijent FOREIGN KEY (jmbg) REFERENCES pacijent (jmbg)
ON UPDATE NO ACTION
ON DELETE NO ACTION
);
```

```
CREATE TABLE LEKAR
(
sifra_lekara bigint NOT NULL PRIMARY KEY,
ime varchar(30) NOT NULL,
prezime varchar(30) NOT NULL
);
```

```
CREATE TABLE PREGLEDA
(
jmbg varchar(13) NOT NULL PRIMARY KEY,
sifra_lekara bigint NOT NULL,
dijagnoza varchar(30) NOT NULL ,
```

```
CONSTRAINT FK_pregleda_karton FOREIGN KEY (jmbg) REFERENCES otvara_karton (jmbg)
ON UPDATE NO ACTION
ON DELETE NO ACTION ,
CONSTRAINT
FK_pregleda_lekar FOREIGN KEY (sifra_lekara) REFERENCES lekar (sifra_lekara)
ON UPDATE NO ACTION
ON DELETE NO ACTION
);
```

```
#ALTER TABLE PACIJENT
#add constraint ck_pacijent check(godina_rodjenja>1900);
```

```
#POCETAK TRIGERA
DELIMITER $$
CREATE TRIGGER provera_pacijent_godina BEFORE INSERT ON PACIJENT
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
IF (NEW.godina_rodjenja<=1900)
THEN
SET msg='Greska: Godina rodjenja pacijenta mora biti veca od 1900!';
SIGNAL sqlstate '45000' SET message_text= msg;
END IF ;
END $$
DELIMITER ;
#KRAJ TRIGERA
```

```
use ordinacija;
a) insert into pacijent
values ('1203965850033', 'Marko', 'Markovic', 'AB-',1965) ;
```

```
# Pokusati sa unosom pacijenta cija je godina rodjenja manja od 1900!
```

```
insert into sestra
values (null, 'Mina', 'Mirkovic', 'viša medicinska sestra');
```

```
insert into otvara_karton
values ('1203965850033', 1, '2009-01-25');
```

```
insert into lekar
values (1, 'Đorđe', 'Šarac');
```

```
insert into pregleda
values ('1203965850033', 1, 'b.o. ');
```

```
b)
update pregleda
set dijagnoza='tahikardija'
where jmbg='1203965850033' and sifra_lekara=1 ;
```

```
c)
delete from pregleda
where jmbg='1203965850033' and sifra_lekara=1 ;
delete from otvara_karton
where jmbg='1203965850033' ;
delete from pacijent
```


where jmbg='1203965850033';