Zadaci za vježbu

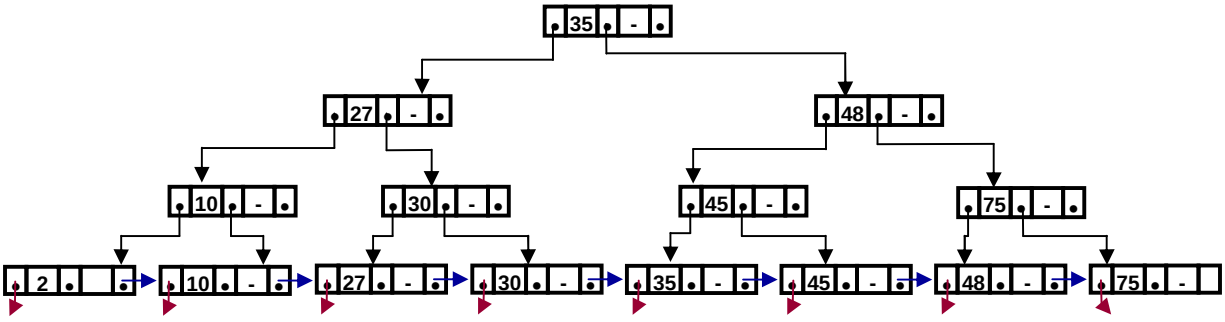
(uz predavanja 9 - Fizička organizacija podataka, 10 - Integritet baze podataka)

1. Fizička organizacija podataka

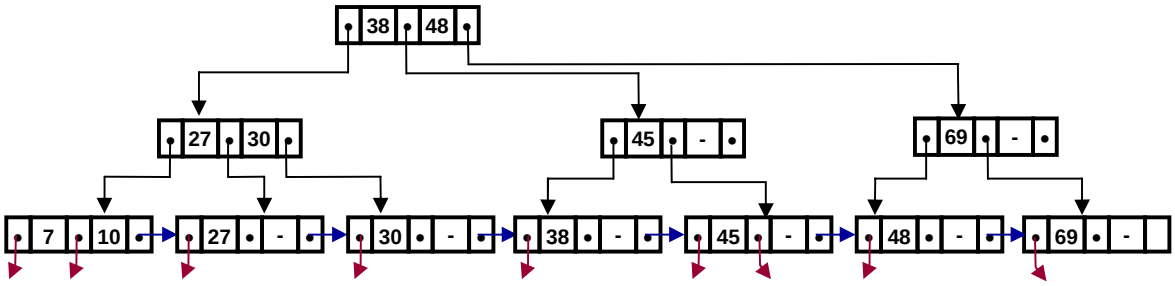
1. U B^+ stablu reda 15, koliko najmanje kazaljki može sadržavati interni čvor koji nije korijen?
2. U B^+ stablu reda 15, koliko najmanje kazaljki može sadržavati korijen?
3. U B^+ stablu reda 20, koliko najviše, a koliko najmanje kazaljki može sadržavati list stabla?
4. Koji je najveći mogući broj razina koji može imati B^+ stablo reda **n** = 20 za broj n-torki **m** = 500?
5. Koliko je najviše UI operacija potrebno da bi se dohvatio zapis prema ključu za B^+ stablo iz prethodnog zadatka.
6. Koliko je maksimalno UI operacija potrebno za dohvat zapisa prema vrijednosti ključa ako je broj n-torki 1 000 000, a ključevi su organizirani u B^+-stablo reda 16?
7. Relacija stud(mbr, prez, ime) sadrži n-torke sa sljedećim vrijednostima atributa mbr: 2, 5, 6, 10, 14, 16, 19, 20, 22, 27, 30, 35. Nacrtati B^+ stablo reda **6** za atribut mbr tako da popunjenost stabla bude minimalna.
8. Koliko n-torki sadrži relacija ako je nad njom izgrađeno B^+ stablo reda 100, s ukupno 6 razina i s **minimalnom** popunjenošću **svih** čvorova?
9. Koliko n-torki sadrži relacija ako je nad njom izgrađeno B^+ stablo reda 100, s ukupno 6 razina i s **maksimalnom** popunjenošću **svih** čvorova?

(Zadaci 10–12 su samo za one koji žele znati više i odnose se na Predavanja 9. – Dopunski materijali)

10. Iz B^+ stabla sa slike obrisati zapis 2.



11. U B^+ stablo na slici (zadatak 10) ubaciti zapis 42.
12. U B^+ stablo sa slike ubaciti zapis 9.



U zadacima koji slijede koriste se relacije iz baze podataka **studAdmin.** Detaljnije objašnjenje **studAdmin** baze podataka možete pronaći na web stranicama predmeta.

13. Kreirana je relacija mjestoNovo sljedećom naredbom:

```
CREATE TABLE mjestoNovo
     (rbrMjesto     SERIAL      NOT NULL
    , pbr           INTEGER     NOT NULL
    , nazMjesto     NCHAR(40)   NOT NULL
    , sifZupanija   SMALLINT);
```

Nad tom relacijom kreirajte **najmanji mogući** broj indeksa koji će omogućiti efikasno obavljanje svih sljedećih upita:

```
SELECT * FROM mjestoNovo ORDER BY rbrMjesto, nazMjesto, sifZupanija
SELECT * FROM mjestoNovo ORDER BY rbrMjesto, nazMjesto
SELECT * FROM mjestoNovo WHERE nazMjesto MATCHES 'N*' AND rbrMjesto > 10
SELECT * FROM mjestoNovo WHERE rbrMjesto = 30 AND sifZupanija = 51000
SELECT * FROM mjestoNovo ORDER BY rbrMjesto, nazMjesto, sifZupanija DESC
SELECT * FROM mjestoNovo WHERE pbr BETWEEN 10000 AND 20000
SELECT * FROM mjestoNovo ORDER BY pbr DESC, rbrMjesto DESC
SELECT * FROM mjestoNovo
        ORDER BY rbrMjesto DESC, nazMjesto DESC, sifZupanija ASC
```

14. Uz pomoć indeksa osigurajte da dva studenta unutar istog mjesta rođenja ne mogu imati jednako ime i prezime. Isprobajte što će se dogoditi ukoliko pomoću INSERT ili UPDATE naredbe pokušate narušiti to pravilo
15. Uništite indeks koji ste kreirali u prethodnom zadatku.
16. Nad relacijom student često se obavljaju upiti oblika:

```
SELECT * FROM student WHERE ime = 'Ante' AND prezime = 'Horvat';
SELECT * FROM student ORDER BY prezime, ime;
SELECT * FROM student WHERE prezime = 'Horvat';
SELECT * FROM student WHERE ime = 'Ante';
```

Koje ćete indekse kreirati da bi se upiti obavljali što efikasnije?

2. Integritetska ograničenja

1. Napisati naredbe kojima će se stvoriti nove relacije:

TIPVOZIL0 = {sifTipVozilo, nazTipVozilo}
$PK_{TIPVOZILO}$ = {sifTipVozilo}

VOZILO = { sifVozilo, sifTipVozilo , nazVozilo, serijskiBroj, nosivost }
PK_{VOZILO} = { sifVozilo }
$K2_{VOZILO}$ = { serijskiBroj }

Pri tom osigurati:

- entitetski integritet i integritet ključa u obje relacije
- integritet ključa K2 u relaciji VOZILO.
- nemogućnost pojave NULL vrijednosti za bilo koji atribut obje relacije
- ograničenje nosivosti na interval **[100, 50000]** (radi se o broju kilograma koje vozilo može nositi)
- referencijski integritet pozivajuće relacije VOZILO s obzirom na pozivanu relaciju TIPVOZILO

Unesite nekoliko zapisa u obje tablice i pokušajte narušiti sva integritetska ograničenja koja ste definirali kako biste se upoznali s porukama koje SUBP javlja u slučaju pokušaja narušavanja integriteta.

2. Napisati naredbe kojima će se stvoriti nove relacije:

TIPVOZIL0 = {sifTipVozilo, nazTipVozilo}
$PK_{TIPVOZILO}$ = {sifTipVozilo}

VOZILO = { sifVozilo, sifTipVozilo , nazVozilo, serijskiBroj, nosivost }
PK_{VOZILO} = { sifVozilo }
$K2_{VOZILO}$ = { serijskiBroj }

Niti jedan atribut u obje relacije ne smije poprimiti NULL vrijednost. Osim NOT NULL ograničenja nemojte definirati niti jedno drugo ograničenje.

Kada su relacije napravljene:

a) Osigurati entitetski integritet i integritet ključa u obje relacije relacije.
b) Osigurati integiritet ključa (K2) u relaciji VOZILO.
c) Atribut nosivost u relaciji VOZILO se odnosi na broj kilograma koje vozilo može nositi – osigurati da se može unijeti samo broj u intervalu [100, 50000].

d) Osigurati referencijski integritet pozivajuće relacije VOZILO s obzirom na pozivanu relaciju TIPVOZILO.
e) Uklonite integritetska ograničenja definirana u zadacima a) – d)

3. Napisati naredbe kojima će se stvoriti nove relacije:

TECAJ = {sifTecaj, nazTecaj}
PK_{TECAJ} = {sifTecaj}

OSOBA= {sifOsoba, imeOsoba, prezOsoba}
PK_{OSOBA} = {sifOsoba}

POLAZNIK_TECAJ = { sifTecaj, sifOsoba, datumUpis, datumIspis}
$PK_{POLAZNIK\_TECAJ}$ = { sifTecaj, sifOsoba }

POLAZNIK_TECAJ_RATA = { sifTecaj, sifOsoba, datumRata, iznosRata}
$PK_{POLAZNIK\_TECAJ\_RATA}$ = { sifTecaj, sifOsoba, datumRata }

Pri tom osigurati:

- samo atribut ***datumIspis*** relacije POLAZNIK_TECAJ smije poprimiti NULL vrijednost
- entitetski integritet i integritet ključa u obje relacije
- referencijski integritet pozivajuće relacije POLAZNIK_TECAJ s obzirom na pozivane relacije TECAJ i OSOBA, kao i referencijski integritet pozivajuće relacije POLAZNIK_TECAJ_RATA s obzirom na poziavnu relaciju POLAZNIK_TECAJ
- datum ispisa mora biti veći od datum upisa polaznika tečaja
- iznos rate mora biti veći od 0
- pri brisanju osobe ili tečaja automatski se brišu i svi zapisi iz relacije POLAZNIK_TECAJ vezani za tu osobu ili tečaj, kao i zapisi iz relacije POLAZNIK_TECAJ_RATA

Unesite nekoliko zapisa u sve tablice, pokušajte brisati/mijenjati zapise relacija tecaj i osoba, te analizirajte sadržaj relacije polaznik_tecaj nakon obavljenih naredbi.

Rješenja

1. Fizička organizacija podataka

1. 8
2. 2
3. 19, 10
4. 3 razine. (Najveći broj razina je u slučaju kada je popunjenost čvorova minimalna)
5. 4 UI operacije. (3 UI operacije za dohvat lista u kojem se nalazi kazaljaka i 1 UI operacija za dohvat bloka s podacima.)
6. Maksimalan broj UI operacija će biti kada se radi o najlošijem slučaju, odnosno o B^+-stablu sa minimalnom popunjenosti čvorova. Prvo se treba odrediti broj razina B^+-stabla.

   0 razina = 1 čvor, najmanje 2 kazaljke

   1 razina = 2 čvora, najmanje 16 kazaljki

   2 razina = 16 čvorova, najmanje 128 kazaljki

   3 razina = 128 čvorova, najmanje 1 024 kazaljki

   4 razina = 1 024 čvorova, najmanje 8 192 kazaljki

   5 razina = 8 192 čvorova, najmanje 65 536 kazaljli

   6 razina = 65 536 čvorova, najmanje 524 288 kazaljki

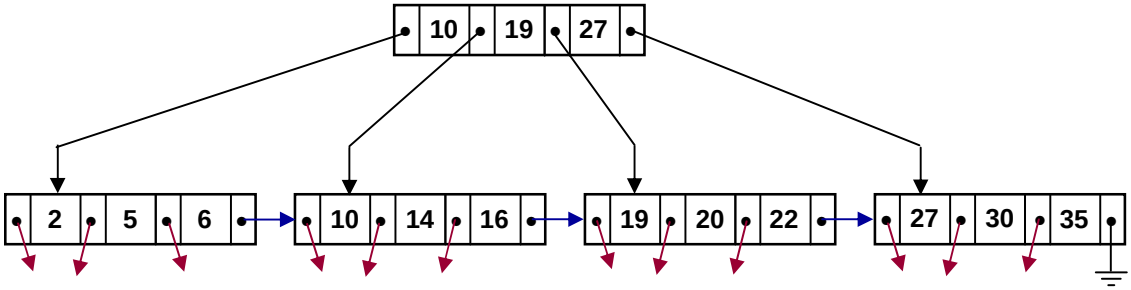
   7 razina = 524 288 čvorova, 4 194 304 kazaljki

   Da bi stablo imalo 8 razina (uključujući i razinu korijena), moralo bi imati preko 4000000 kazaljki na n-torke. To znači da stablo koje ima 1000000 n-torki može imati maksimalno 7 razina (uključujući i razinu korijena).

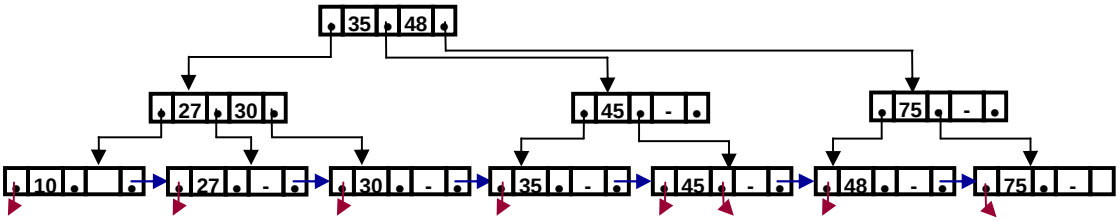
   Razine se mogu izračunati i ovako: $d \leq \log_{\lceil n/2 \rceil}(m/2) + 1$, $d \leq \log_6(500000) = 7$. (Oni koje žele znati više objašnjenje mogu naći u Predavanja 9. - Dopunski materijali.)

   Broj UI operacija je 8. (7 UI operacije za dohvat lista u kojem se nalazi kazaljaka i 1 UI operacija za dohvat bloka s podacima.)

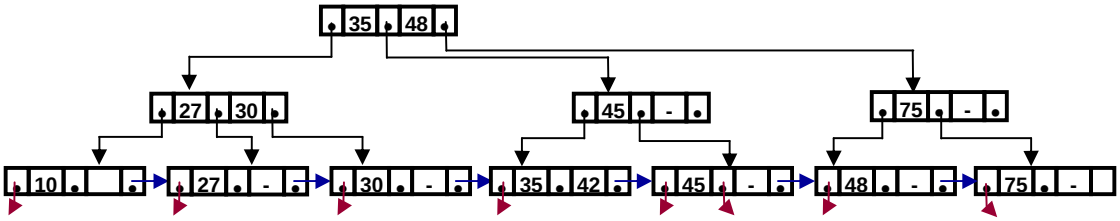
7.



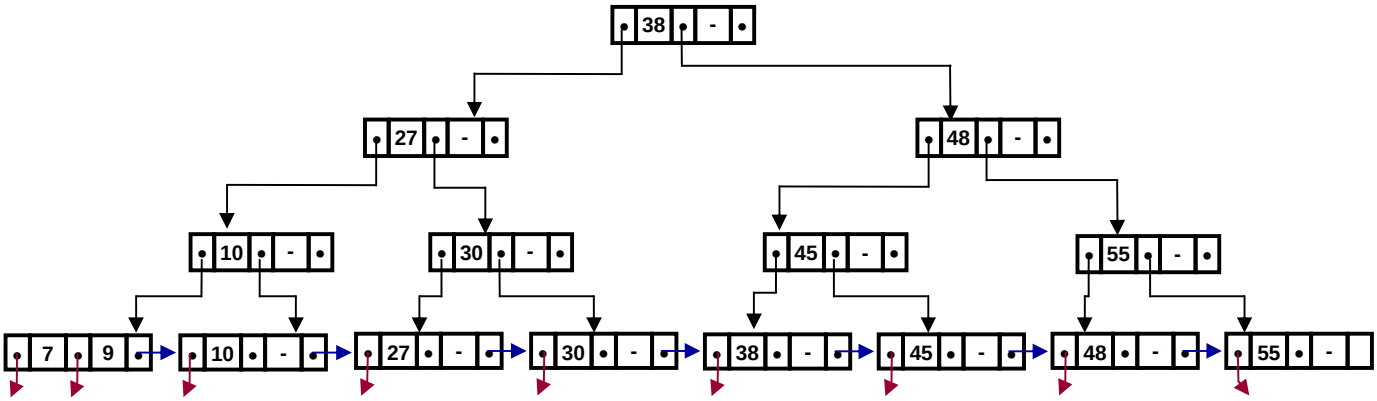
8. $2 \times 50^5 = 625\,000\,000$
9. $100^5 \times 99 = 990\,000\,000\,000$
10.



11.



12.



13.

```
CREATE INDEX mjestoNovoIdx1
         ON mjestoNovo(rbrMjesto, nazMjesto, sifZupanija);
CREATE INDEX mjestoNovoIdx2
          ON mjestoNovo(rbrMjesto, nazMjesto, sifZupanija DESC);
CREATE INDEX mjestoNovoIdx3 ON mjestoNovo(pbr, rbrMjesto);
   CREATE    INDEX    mjestoNovoIdx4    ON    mjestoNovo(rbrMjesto,
sifZupanija);
```

14..

```
CREATE UNIQUE INDEX stud2Ind ON stud(pbrRod, imeStud, prezStud);
```

15. .

```
DROP INDEX stud2Ind;
```

16.

```
CREATE INDEX stud_ime ON student(ime);
CREATE INDEX stud_prezime_ime ON student(prezime, ime);
```

2. Integritetska ograničenja

1.

```
CREATE TABLE TipVozilo(
    sifTipVozilo SMALLINT PRIMARY KEY CONSTRAINT pkTipVozilo
  , nazTipVozilo CHAR(50) NOT NULL
);

CREATE TABLE Vozilo(
    sifVozilo     INTEGER
  , sifTipVozilo  SMALLINT NOT NULL
  , nazVozilo     CHAR(50) NOT NULL
  , serijskiBroj  CHAR(20) NOT NULL UNIQUE CONSTRAINT unqSerijskiBroj
  , nosivost      INTEGER  NOT NULL -- ne smije biti SMALLINT!
  , PRIMARY KEY (sifVozilo) CONSTRAINT pkVozilo
  , FOREIGN KEY (sifTipVozilo) REFERENCES TipVozilo
                                    CONSTRAINT fkVoziloTipVozilo
  , CHECK (nosivost BETWEEN 100 AND 50000) CONSTRAINT chkNosivost
);

/* SELECT * FROM TipVozilo; */
```

Obavljati naredbe jednu po jednu i gledati rezultat obavljanja:

```
INSERT INTO TipVozilo VALUES (1, 'Osobni automobil');
INSERT INTO TipVozilo VALUES (1, 'Kombi'); /* Greška: pkTipVozilo */
INSERT INTO TipVozilo VALUES (2, 'Kombi');
INSERT INTO TipVozilo VALUES (3, 'Kamion');

/*
SELECT *
  FROM Vozilo, TipVozilo
 WHERE Vozilo.sifTipVozilo = TipVozilo.sifTipVozilo;
*/
INSERT INTO Vozilo VALUES (100, 1, 'V8 interceptor'    , 'MMX-123'  ,
1200);
INSERT INTO Vozilo VALUES (100, 2, 'Kamion interceptor', 'KX-123'   ,
12000); /* Greška: pkVozilo */
INSERT INTO Vozilo VALUES (101, 3, 'Izmisljeni kamion' , 'KX-123-56',
62000); /* Greška: chkNosivost */
INSERT INTO Vozilo VALUES (102, 2, 'Caddy'             , 'MMX-123'  ,
1500); /* Greška: unqSerijskiBroj */
INSERT INTO Vozilo VALUES (103, 5, 'Caddy'             , 'CDY-0-123',
1500); /* Greška: fkVoziloTipVozilo */
INSERT INTO Vozilo VALUES (104, 2, 'Caddy'             , 'CDY-1-123',
1500);
INSERT INTO Vozilo VALUES (105, 2, NULL                , 'CDY-2-123',
2500);  /* Greška: naziv vozila ne smije biti NULL */
```

Na kraju obrisati relacije:

```
DROP TABLE TipVozilo;
DROP TABLE Vozilo;
```

2. Integritetska ograničenja se mogu naknadno (nakon što su relacije već stvorene) definirati pomoću naredbe ALTER TABLE.
Pri definiciji ograničenja sustav provjerava zadovoljava li trenutačna instanca baze podataka integritetsko ograničenje:

- ukoliko zadovoljava, integritetsko ograničenje se upisuje u rječnik podataka i provjerava od tog trenutka nadalje
- ukoliko ne zadovoljava, sustav odbija prihvatiti definiciju integritetskog ograničenja

```
CREATE TABLE TipVozilo(
    sifTipVozilo SMALLINT
  , nazTipVozilo CHAR(50) NOT NULL
);

CREATE TABLE Vozilo(
    sifVozilo     INTEGER
  , sifTipVozilo  SMALLINT NOT NULL
  , nazVozilo     CHAR(50) NOT NULL
  , serijskiBroj  CHAR(20) NOT NULL
  , nosivost      INTEGER  NOT NULL -- ne smije biti SMALLINT!
);
```

Ukoliko u relacije unesete neki zapis koji ne zadovoljava integritetska ograničenja koja želite napraviti – ograničenja neće biti stvorena već ćete dobiti poruku o pogrešci.

a) Osigurati entitetski integritet i integritet ključa u obje relacije relacije.

Naredba ima oblik:

```
ALTER TABLE imeRelacije
   ADD CONSTRAINT PRIMARY KEY (atribut [, ...])
      [CONSTRAINT imeOgranicenja]
```

```
ALTER TABLE TipVozilo
   ADD CONSTRAINT PRIMARY KEY (sifTipVozilo)
      CONSTRAINT pkTipVozilo;
ALTER TABLE Vozilo
   ADD CONSTRAINT PRIMARY KEY (sifVozilo)
      CONSTRAINT pkVozilo;
```

b) Osigurati integiritet ključa (K2) u relaciji VOZILO.

Naredba ima oblik:

```
ALTER TABLE imeRelacije
   ADD CONSTRAINT UNIQUE (atribut [, ...])
      [CONSTRAINT imeOgranicenja]
```

```
ALTER TABLE Vozilo
   ADD CONSTRAINT UNIQUE(serijskiBroj)
      CONSTRAINT unqSerijskiBroj;
```

c) Atribut nosivost u relaciji VOZILO se odnosi na broj kilograma koje vozilo može nositi – osigurati da se može unijeti samo broj u intervalu [100, 50000].

Naredba ima oblik:

```
ALTER TABLE imeRelacije
   ADD CONSTRAINT CHECK (izraz)
      [CONSTRAINT imeOgranicenja]

ALTER TABLE Vozilo
   ADD CONSTRAINT  CHECK (nosivost BETWEEN 100 AND 50000)
      CONSTRAINT chkNosivost;
```

d) Osigurati referencijski integritet pozivajuće relacije VOZILO s obzirom na pozivanu relaciju TIPVOZILO.

Naredba ima oblik:

```
ALTER TABLE pozivajucaRel
   ADD CONSTRAINT FOREIGN KEY (atribut [, ...]) REFERENCES pozivanaRel
      [CONSTRAINT imeOgranicenja]

ALTER TABLE Vozilo
   ADD CONSTRAINT FOREIGN KEY (sifTipVozilo) REFERENCES TipVozilo
      CONSTRAINT fkVoziloTipVozilo;
```

e) Uklonite integritetska ograničenja definirana u zadacima a) – d):

Naredba ima oblik:

```
ALTER TABLE ImeRelacije DROP CONSTRAINT imeOgranicenja

ALTER TABLE Vozilo    DROP CONSTRAINT fkVoziloTipVozilo;
ALTER TABLE Vozilo    DROP CONSTRAINT chkNosivost;
ALTER TABLE Vozilo    DROP CONSTRAINT unqSerijskiBroj;

ALTER TABLE TipVozilo DROP CONSTRAINT pkTipVozilo;
ALTER TABLE Vozilo    DROP CONSTRAINT pkVozilo;
```

Napomena: *ukoliko bi prvo uklonili primarni ključ u relaciji TipVozila samim time bi bio uklonjen i strani ključ fkVoziloTipVozilo jer ne može postojati ukoliko ne postoji ključ na pozivanoj relaciji.*
Probajte, na primjer, ***prvo*** *obaviti naredbu iz d) dijela rješenja (dok još niste obavili naredbu iz a) dijela odnosno dok još ne postoji primarni ključ u relaciji TipVozilo) i pogledajte poruku pogreške.*

Napomena: *naknadno se mogu dodati i NOT NULL ograničenja. Neformalno, u IBM Informix SUBP-u ta naredba ima oblik:*

```
ALTER TABLE imeRelacija
   MODIFY PonovljenaDefinicijaAtributa
```

Npr.

```
ALTER TABLE Vozilo
   MODIFY nazVozilo CHAR(60) NOT NULL
      CONSTRAINT nnNazVozilo;
```

Primijetite da je na ovaj način moguće redefinirati i tip podatka (kao što je u primjeru promijenjen `CHAR(50)` u `CHAR(60)`).

Na kraju obrisati relacije:

```
DROP TABLE TipVozilo;
DROP TABLE Vozilo;
```

3.

```
CREATE TABLE tecaj(
    sifTecaj INTEGER PRIMARY KEY CONSTRAINT pkTecaj
  , nazTecaj CHAR(50) NOT NULL
);

CREATE TABLE osoba(
    sifOsoba INTEGER PRIMARY KEY CONSTRAINT pkOsoba
  , imeOsoba CHAR(50) NOT NULL
  , prezOsoba CHAR(50) NOT NULL
);

CREATE TABLE polaznik_tecaj(
    sifOsoba INTEGER
  , sifTecaj INTEGER
  , datumUpis DATE NOT NULL
  , datumIspis DATE
  , PRIMARY KEY (sifOsoba, sifTecaj) pkPolaznikTecaj
  , FOREIGN KEY (sifOsoba) REFERENCES osoba (sifOsoba)
      ON DELETE CASCADE
  , FOREIGN KEY (sifTecaj) REFERENCES tecaj (sifTecaj)
      ON DELETE CASCADE
  , CHECK (datumIspis IS NULL or
           datumIspis IS NOT NULL and datumIspis > datumUpis)
);

CREATE TABLE polaznik_tecaj_rata(
    sifOsoba INTEGER
  , sifTecaj INTEGER
  , datumRata DATE NOT NULL
```

```
  , iznosRata DECIMAL (7,2) NOT NULL
  , PRIMARY KEY (sifOsoba, sifTecaj, datumRata) pkPolaznikTecajRata
  , FOREIGN KEY (sifOsoba, sifTecaj)
     REFERENCES polaznik_tecaj (sifOsoba, sifTecaj)
       ON DELETE CASCADE
  , CHECK (iznosRata > 0)
);
```

Obaviti naredbe za unos sadržaja:

```
INSERT INTO tecaj VALUES (1, 'Baze podataka');
INSERT INTO tecaj VALUES (2, 'Uvod u baze podataka');

INSERT INTO osoba VALUES (1, 'Ante', 'Zenić');
INSERT INTO osoba VALUES (2, 'Laura', 'Krolo');
INSERT INTO osoba VALUES (3, 'Dino', 'Balić');

INSERT INTO polaznik_tecaj VALUES (1, 1, '10.12.2007', NULL);
INSERT INTO polaznik_tecaj VALUES (2, 2, '10.6.2007', NULL);
INSERT INTO polaznik_tecaj VALUES (3, 1, '10.12.2007', '10.12.2009');
INSERT INTO polaznik_tecaj VALUES (3, 2, '10.12.2007', '10.12.2009');

INSERT INTO polaznik_tecaj_rata VALUES (3, 2, '10.12.2007', 100);
INSERT INTO polaznik_tecaj_rata VALUES (2, 2, '10.12.2007', 150);
INSERT INTO polaznik_tecaj_rata VALUES (3, 1, '10.12.2007', 200);
INSERT INTO polaznik_tecaj_rata VALUES (3, 2, '5.12.2007', 600);

/* primjer ON DELETE CASCADE*/
DELETE FROM osoba WHERE sifOsoba = 2;
SELECT * FROM polaznik_tecaj;
SELECT * FROM polaznik_tecaj_rata;


DELETE FROM tecaj WHERE sifTecaj = 1;
SELECT * FROM polaznik_tecaj;
SELECT * FROM polaznik_tecaj_rata;
```

Na kraju obrisati relacije:

```
DROP TABLE tecaj;
DROP TABLE osoba;
DROP TABLE polaznik_tecaj;
DROP TABLE polaznik_tecaj_rata;
```