

Dekanski rok iz Baza podataka (50 bodova)

17. rujna 2019.

Zadaci 1 – 6 odnose se na bazu podataka prikazanu na slici.

U bazi podataka **firstFM** servisa za praćenje slušanja glazbe evidentiraju se podatci o pjesmama (**pjesma**) za koje se bilježi naziv i trajanje (tipa INTERVAL). Pjesma se može nalaziti na samo jednom albumu (**album**). Ako pjesma doživi reizdanje na nekom drugom albumu, to se bilježi kao dva različita zapisa u relaciji **pjesma** (imaju različite vrijednosti atributa **sifPjesma**). Pjesmu može izvoditi (**izvodi**) više izvođača (**izvodac**). Za albume se evidentira naziv, godina izdanja i izvođač pod kojim je album izdan, dok se za izvođače evidentira njihovo ime. Za korisnike (**korisnik**) se evidentira korisničko ime (**klme**), datum rođenja i posljednja pjesma koju je korisnik slušao (**sifZadnjaPj**) i u koje vrijeme (**datVrZadnjeSl** tipa TIMESTAMP). Prilikom slušanja pjesme (**slusanje**) bilježi se korisnik, vrijeme početka i vrijeme završetka slušanja pjesme (**datVrPoc**, odnosno **datVrKraj** tipa TIMESTAMP).

Ključevi relacija su podcrtani. Atributi **klme**, **nazAlbum**, **nazIzvodac** i **nazPjesma** su tipa VARCHAR.

korisnik					album				izvodac	
<u>sifKorisnik</u>	klme	datRod	sifZadnjaPj	datVrZadnjeSl	<u>sifAlbum</u>	nazAlbum	godina	sifIzvodac	<u>sifIzvodac</u>	nazIzvodac
1	theFan	18.12.1983	100	28.08.2019 15:14:29	5005	The Works	1984	1001	1001	Queen
2	GaGa	21.07.1989	100	01.07.2019 21:49:34	5100	Let's Dance	1983	2100	2100	David Bowie
3	noob	12.02.2000	NULL	NULL	5200	Heathen	2002	2100	3500	The Cure
4	earz	15.07.2000	200	13.03.2017 17:18:01
...

slusanje				pjesma				izvodi	
<u>sifKorisnik</u>	<u>datVrPoc</u>	<u>sifPjesma</u>	<u>datVrKraj</u>	<u>sifPjesma</u>	nazPjesma	trajanje	sifAlbum	<u>sifPjesma</u>	sifIzvodac
1	26.08.2019 21:45:00	100	28.08.2019 21:49:06	100	Under Pressure	00:04:08	5100	100	1001
1	28.08.2019 15:00:10	300	28.08.2019 15:07:47	200	Radio Ga Ga	00:05:48	5005	100	2100
1	28.08.2019 15:14:29	300	28.08.2019 15:21:56	300	Let's Dance	00:07:37	5100	200	1001
2	29.06.2019 10:01:34	200	29.06.2019 10:01:44	300	2100
4	01.07.2017 21:49:37	100	01.07.2017 21:50:04
4	13.03.2017 17:18:01	100	NULL
...

Na slici **nisu** prikazane sve n-torke sadržane u relacijama.

- (4 boda)** Za sve korisnike koji su barem jednom slušali pjesme s albuma koje je **izdao** izvođač 'David Bowie' ispisati korisničko ime, koliko su puta slušali pjesme s albuma tog izvođača, te datum i vrijeme i naziv zadnje pjesme koju su slušali. Zapise poredati silazno s obzirom na broj slušanja.
- (5 bodova)** Za svakog izvođača kojega je tijekom prethodnog mjeseca slušalo ukupno 3 ili više različitih korisnika ispišite naziv, broj slušanja pjesama koje oni **izvode** a koje su slušane tijekom prethodnog mjeseca, te ukupan broj različitih slušatelja tog izvođača tijekom proteklog mjeseca. Ako nekog izvođača nije tijekom prethodnog mjeseca slušao niti jedan korisnik, prikazati i takve zapise u rezultatu, a kao broj slušanja i broj različitih slušatelja ispisati vrijednost 0.
Napomena: Upit mora ispravno vraćati rezultate za prethodni mjesec bez obzira na vrijeme pokretanja.
- (9 bodova)** Servis **firstFM** donio je pravilo da se evidentiraju samo ona slušanja prilikom kojih je odslušano barem pola ukupnog trajanja pjesme i Vaš je zadatak popraviti stanje baze podataka:
 - Napišite SQL naredbe kojima će se prilikom unosa ili brisanja zapisa iz relacije **slusanje** osigurati konzistentnost podataka o zadnjoj slušanoj pjesmi u relaciji **korisnik**. Vrijeme zadnjeg slušanja se postavlja prema vremenu početka slušanja pjesme. Pretpostavite za operaciju unosa da se mogu unijeti samo kronološki najnoviji zapisi (nema retrogradnog naknadnog unosa slušanja). Prilikom pokušaja unosa zapisa o slušanju gdje je odslušano manje od pola pjesme, odbiti unos uz ispis poruke: 'Error: Odslušano manje od pola pjesme!'.
 - Napišite SQL naredbu kojom će se izbrisati sva evidentirana slušanja pjesama prilikom kojih je odslušano manje od pola ukupnog trajanja pjesme.
- (3 boda)** Administrator baze podataka omogućio je svim korisnicima uspostavu korisničke sjednice, spajanje na bazu **firstFM** i pristup objektima sadržanim u shemi **public**. Napišite SQL naredbe kojima će administrator korisniku **theFan** dodijeliti ovlasti za pregled, unos i brisanje podataka o vlastitim slušanjima pjesama.
- (4 boda)** Napišite **izraz relacijske algebre** kojim će se za sve albume izdane 80-tih godina ispisati naziv i godina izdanja albuma, ukupan broj zabilježenih slušanja pjesama s tog albuma te ukupan broj različitih izvođača koji **izvode pjesme** s tog albuma. Ispis mora sadržavati i albume 80-ih koji nisu bili slušani, za koje će se 0 ispisati kao broj zabilježenih slušanja.
- (5 bodova)** Pretpostavite da su u bazi podataka **firstFM** definirana ograničenja primarnih ključeva (podcrtani na slici) i da su strani ključevi definirani isključivo među **istoimenim** atributima relacija, te da su posljedično automatski kreirani odgovarajući indeksi od strane SUBP-a. Optimizator upita raspolaže sljedećim statističkim podacima:

N(korisnik) = 15 000

V(sifZadnjaPj, korisnik) = 5 000

V(datVrZadnjeSl, korisnik) = 5 000

N(pjesma) = 4 500 000

V(nazPjesma, pjesma) = 3 000 000

V(trajanje, pjesma) = 1 000

N(album) = 500 000

V(nazAlbum, album) = 450 000

V(godina, album) = 100

Obavlja se operacija: $\sigma_{\text{trajanje} < 00:04:30 \wedge \text{godina} = 1998}(\text{korisnik} \bowtie \text{pjesma} \bowtie \text{album})$
 $\text{sifZadnjaPj} = \text{sifPjesma}$

Nacrtajte **inicijalno** i **konačno** stablo upita nakon provedene heurističke optimizacije. Prilikom crtanja inicijalnog stabla upita pretpostavite da je redoslijed spajanja relacija određen redoslijedom kojim su one navedene u izrazu relacijske algebre. Redoslijed spajanja relacija prilikom crtanja konačnog stabla upita odrediti temeljem procjene broja n-torki u rezultatima spajanja. Navesti izraze prema kojima je obavljena procjena broja n-torki u svim međurezultatima. U stablu upita naznačiti očekivani broj n-torki u svim međurezultatima.

7. (5 bodova) Kreirana je tablica **skladište** koja bilježi šifre artikla i količinu artikala pohranjenih u skladištu. Tablicom je prikazan sadržaj relacije **skladište** u trenutku započinjanja naredbe {1}. Odredite rezultate izvođenja naredbi {5}, {7}, {9} i {10}. Pojavljuju li se u transakcijama karakteristični problemi istodobnog pristupa? Ako da, navedite naziv problema, pri obavljanju koje naredbe je primijećen i kojom naredbom je uzrokovan.

skladište	
sifArtikl	kolicina
1	4
2	10

{ 1 }	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;	{3}	BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
{ 4 }	INSERT into skladište values (4, 20)				
{ 5 }	SELECT * FROM skladište;	{6}	UPDATE skladište SET kolicina = kolicina-1 WHERE sifArtikl = 1;	{7}	SELECT * FROM skladište;
		{8}	COMMIT TRANSACTION;	{9}	SELECT * FROM skladište;
{ 10 }	SELECT * FROM skladište;			{11}	UPDATE skladište SET kolicina = kolicina-1 WHERE sifArtikl = 1;
{ 12 }	COMMIT TRANSACTION;			{13}	COMMIT TRANSACTION;

8. (3 boda) Na relacijskoj shemi $R = \{ A, B, C, D, E, F, G \}$ zadan je skup funkcijskih zavisnosti:

$$F = \{ BF \rightarrow A, C \rightarrow GD, D \rightarrow F, E \rightarrow BC, G \rightarrow B \}$$

Dokažite vrijedi li funkcijska zavisnost $DE \rightarrow AC$. Navedite pravila korištena uz svaki korak dokaza.

9. (4 boda) Tablica **korisnik** u bazi podataka **firstFM** sadrži zapise s vrijednostima atributa **sifKorisnik** u intervalu od 1 do 11. Nad atributom **sifKorisnik** stvoren je indeks. Potrebno je nacrtati B^+ stablo reda 4 izgrađeno za atribut **sifKorisnik**, tako da popunjenost stabla bude:

- minimalna
- maksimalna

10. (8 bodova)

U bazi podataka evidentiraju se prijave učenika na ispite Državne mature i učenikove prijave studijskih programa.

Učenik odabire predmete iz kojih želi polagati ispit i razinu težine ispita kojeg želi polagati (niža razina ili viša razina). Za predmet i razinu evidentiraju se njihove šifre i nazivi te raspored polaganja ispita (npr. ispit iz više razine hrvatskog jezika održava se 18.6.2019.).

Učenik može različite predmete polagati na različitim razinama (npr. matematiku može pisati iz niže razine, a fiziku iz više razine).

Ispit iz istog predmeta moguće je polagati više puta (npr. ako prvi put nije položen ili za bolji rang) i pri tom ne nužno iz iste razine (npr. prvi put se polaže matematika iz više razine, a drugi put iz niže razine). Učenik polaže ispite iz predmeta prema definiranom rasporedu za željenu razinu te se evidentira broj ostvarenih bodova i rang kojeg je učenik ostvario polaganjem ispita.

Za učenike evidentira se OIB, ime i prezime te srednja škola koju pohađa. Učenik može pohađati samo jednu srednju školu. Također, evidentiraju se prijave učenika za upis na studije. Učenik može odabrati više studija, gdje za svaki odabrani studij navodi prioritet (npr. učenik koji prvenstveno želi upisati studij medicine, a alternativno studij stomatologije, navest će uz studij medicine prioritet 1., a uz studij stomatologije prioritet 2.).

Za studijske programe evidentira se šifra i naziv, trajanje u semestrima i fakultet koji ga izvodi. Jedan fakultet može izvoditi različite studije, dok određeni studij izvodi samo jedan fakultet.

Fakulteti i srednje škole su obrazovne ustanove za koje se evidentira njihova šifra i naziv. Za školu se evidentira trajanje obrazovnog programa u godinama i jedna vrsta škole (gimnazija, strukovna, umjetnička). Za fakultet se evidentira datum zadnje akreditacije i jedno znanstveno područje (prirodoslovno-matematičko, tehničko, biomedicinsko, društveno-humanističko, ...).

Nacrtajte ER model baze podataka. Navedite sheme entiteta i sheme veza (**označite ključeve**). Svaki entitet opisati **isključivo vlastitim atributima** (osim slabih entiteta, ako postoje u modelu). Nužno je da sve sheme zadovoljavaju **3NF**.

Rješenja

1. (4 boda)

```
SELECT kIme, COUNT(*) AS brSlusanja, datVrZadnjeSl, p2.nazPjesma
FROM korisnik JOIN slusanje USING (sifKorisnik)
              JOIN pjesma p1 USING (sifPjesma)
              JOIN album USING (sifAlbum)
              JOIN izvodac USING (sifIzvodac)
              JOIN pjesma p2 ON sifZadnjaPj = p2.sifPjesma
WHERE nazIzvodac = 'David Bowie'
GROUP BY sifKorisnik, kIme, datVrZadnjeSl, p2.sifPjesma, p2.nazPjesma
ORDER BY brSlusanja DESC;
```

2. (5 bodova)

```
SELECT nazIzvodac, COUNT(datVrPoc), COUNT(DISTINCT sifKorisnik)
FROM slusanje JOIN izvodi USING (sifPjesma)
      RIGHT JOIN izvodac ON izvodi.sifIzvodac = izvodac.sifIzvodac
                      AND EXTRACT(MONTH FROM datVrPoc) =
                          EXTRACT(MONTH FROM (CURRENT_DATE - '1 MONTH'::INTERVAL))
                      AND EXTRACT(YEAR FROM datVrPoc) =
                          EXTRACT(YEAR FROM (CURRENT_DATE - '1 MONTH'::INTERVAL))
GROUP BY izvodac.sifIzvodac, nazIzvodac
HAVING COUNT(DISTINCT sifKorisnik) NOT IN (1,2);
```

3. (9 bodova)

```
a) CREATE OR REPLACE FUNCTION upd_zadnjeSlusanje() RETURNS trigger AS $$      BEGIN
    IF (TG_OP = 'INSERT') THEN
        IF (NEW.datVrKraj - NEW.datVrPoc)*2 < (SELECT trajanje FROM pjesma
                                              WHERE sifPjesma = NEW.sifPjesma) THEN
            RAISE EXCEPTION 'Error: Odslusano manje od pola pjesme!';
        ELSE
            UPDATE korisnik SET sifZadnjaPj = NEW.sifPjesma, datVrZadnjeSl = NEW.datVrPoc
            WHERE sifKorisnik = NEW.sifKorisnik;
            RETURN NEW;
        END IF;
    ELSE -- TG_OP = 'DELETE'
        UPDATE korisnik SET datVrZadnjeSl = (SELECT MAX(datVrPoc) FROM slusanje
                                              WHERE sifKorisnik = OLD.sifKorisnik)
        WHERE sifKorisnik = OLD.sifKorisnik;
        UPDATE korisnik SET sifZadnjaPj = (SELECT sifPjesma FROM slusanje
                                              WHERE sifKorisnik = OLD.sifKorisnik
                                              AND datVrPoc = datVrZadnjeSl )
        WHERE sifKorisnik = OLD.sifKorisnik;
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER del_slusanje
AFTER DELETE ON slusanje
FOR EACH ROW EXECUTE PROCEDURE upd_zadnjeSlusanje();

CREATE TRIGGER ins_slusanje
BEFORE INSERT ON slusanje
FOR EACH ROW EXECUTE PROCEDURE upd_zadnjeSlusanje();
```

b)

```
DELETE FROM slusanje
WHERE datVrKraj-datVrPoc < (SELECT trajanje/2 FROM pjesma
                              WHERE pjesma.sifPjesma = slusanje.sifPjesma);
```

4. (3 boda)

```
CREATE VIEW userDataView AS
SELECT * FROM slusanje
WHERE sifKorisnik IN (SELECT sifKorisnik
                      FROM korisnik
                      WHERE kIme = SESSION_USER) -- moze i CURRENT_USER
WITH CHECK OPTION;

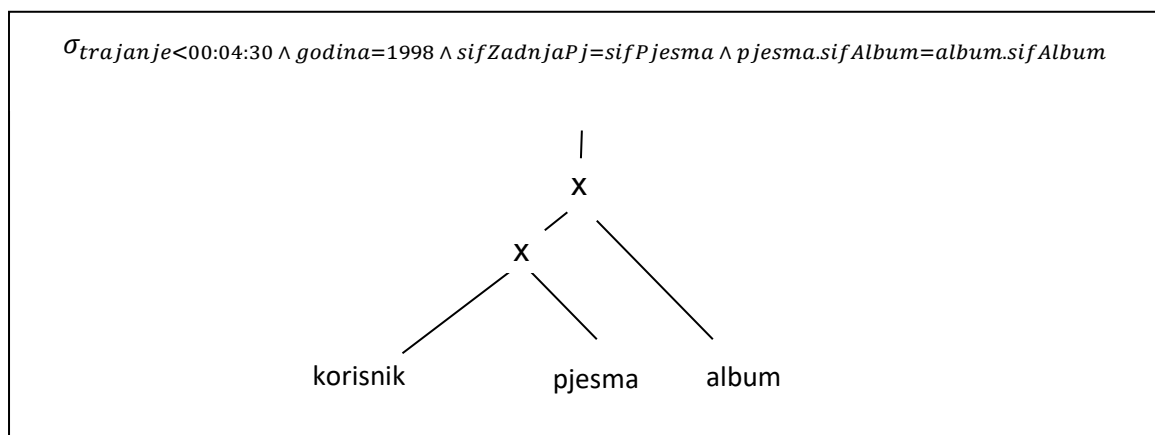
GRANT SELECT, INSERT, DELETE ON userDataView TO theFan;
```

5. (4 boda)

$nazAlbum, godina \bowtie COUNT(datVrPoc), COUNT-DISTINCT(sifIzvodac) \left(\left(\left(\sigma_{godina > 1979 \wedge godina < 1990} (album) \right) \right) \right) \bowtie pjesma \bowtie izvodi \bowtie slusanje$

6. (5 bodova)

Inicijalno stablo:



Potiskivanje selekcije – procjena međurezultata:

- $trajanje < 00:04:30 \rightarrow pjesma$

$$N(pjesma') = N(\sigma_{trajanje < 00:04:30}(pjesma)) = \frac{N(pjesma)}{3} = \frac{4.5 \cdot 10^6}{3} = 1.5 \cdot 10^6$$

- $godina = 1998 \rightarrow album$

$$N(album') = N(\sigma_{godina = 1998}(album)) = \frac{N(album)}{V(godina, album)} = \frac{500\,000}{100} = 5 \cdot 10^3$$

- $sifPjesma = sifZadnjaPj \rightarrow \theta$ -join relacija korisnik i pjesma (računa se u procjeni međurezultata)

Procjena međurezultata za različite redoslijede spajanja:

- *korisnik* i *pjesma*: θ -join

$$N(korisnik \bowtie_{sifZadnjaPj = sifPjesma} pjesma) = \frac{N(korisnik) * N(pjesma')}{MAX(V(sifZadnjaPj, korisnik), V(sifPjesma, pjesma))} = \frac{15 * 10^3 * 1.5 * 10^6}{MAX(5 * 10^3, 4.5 * 10^6)} = \frac{225 * 10^8}{4.5 * 10^6} = 5000$$

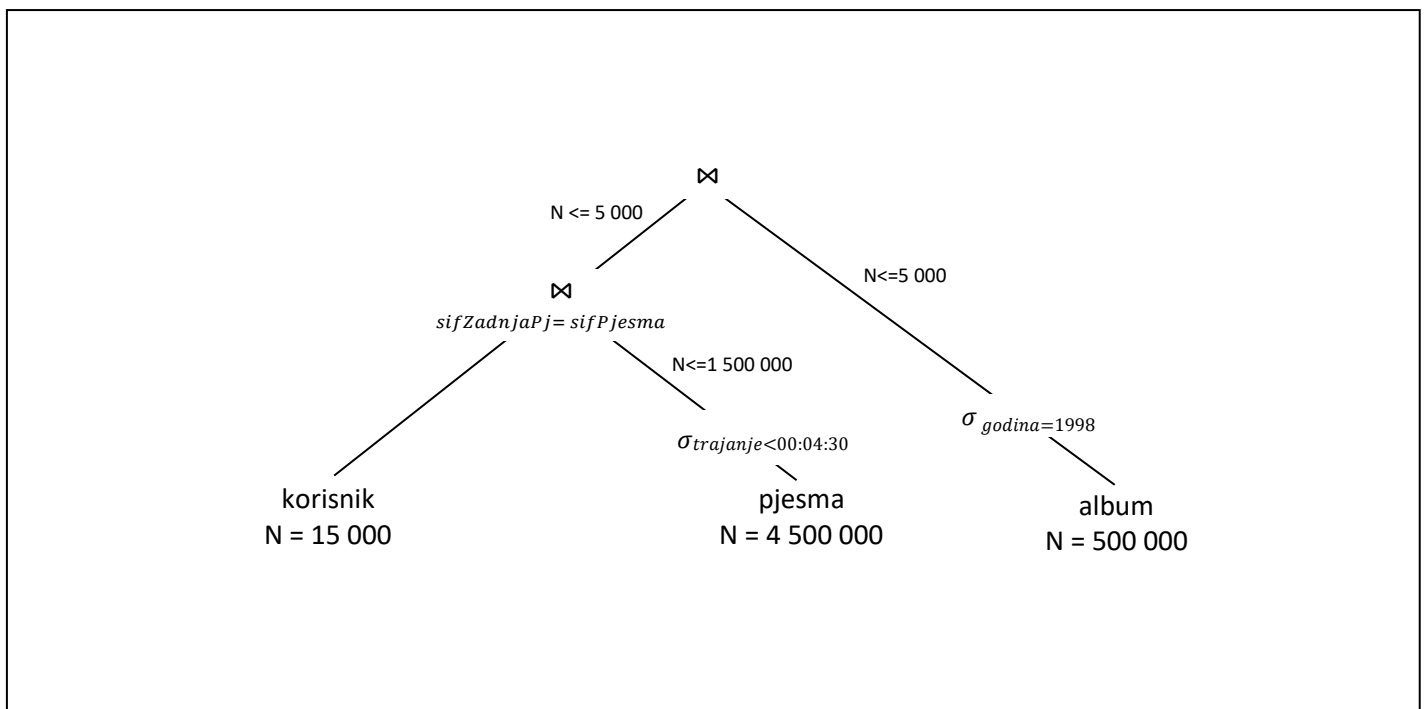
- *pjesma* i *album*: *sifAlbum* je FK u relaciji *pjesma*

$$N(pjesma \bowtie album) = N(pjesma') = 1.5 * 10^6$$

- *korisnik* i *album*: nema zajedničkih atributa \rightarrow izvodi se Kartezijev produkt

$$N(korisnik \bowtie album) = N(korisnik \times album') = 15 * 10^3 * 5 * 10^3 = 75 * 10^6$$

Optimalni redoslijed je: (*korisnik* \bowtie *pjesma*) \bowtie *album*



7. (5 bodova)

{5}

sifArtikl	kolicina
1	4
2	10
4	20

{7}

sifArtikl	kolicina
1	4
2	10

{9}

sifArtikl	kolicina
1	4
2	10

{10}

sifArtikl	kolicina
1	3
2	10
4	20

Vidljiva je pojava neponovljivog čitanja u naredbama {5} i {10} transakcije A, zbog naredbe {6} u transakciji B.

8. (3 boda)

DE \rightarrow DE (refleksivost)

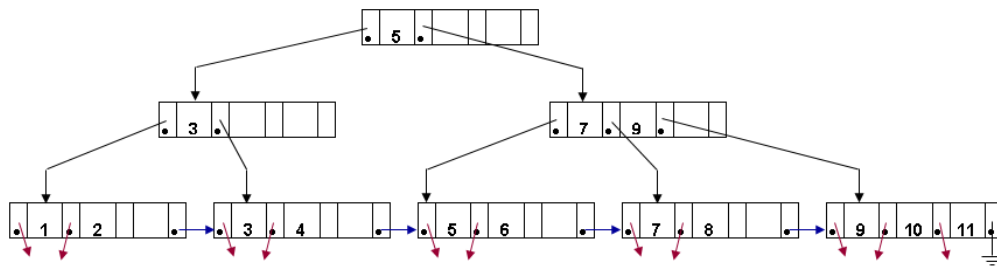
DE \rightarrow DE i D \rightarrow F \Rightarrow DE \rightarrow DEF (akumulacija)

DE \rightarrow DEF i E \rightarrow BC \Rightarrow DE \rightarrow BCDEF (akumulacija)

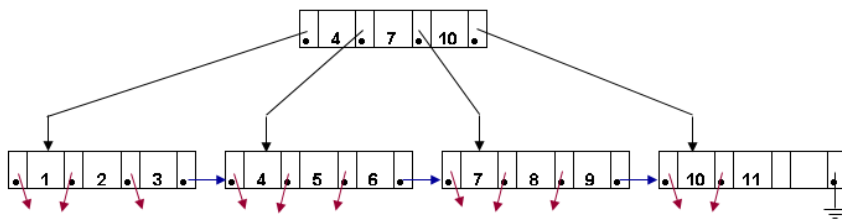
DE \rightarrow BCDEF i BF \rightarrow A \Rightarrow DE \rightarrow ABCDEF (akumulacija)

DE \rightarrow ABCDEF \Rightarrow DE \rightarrow AC (dekompozicija)

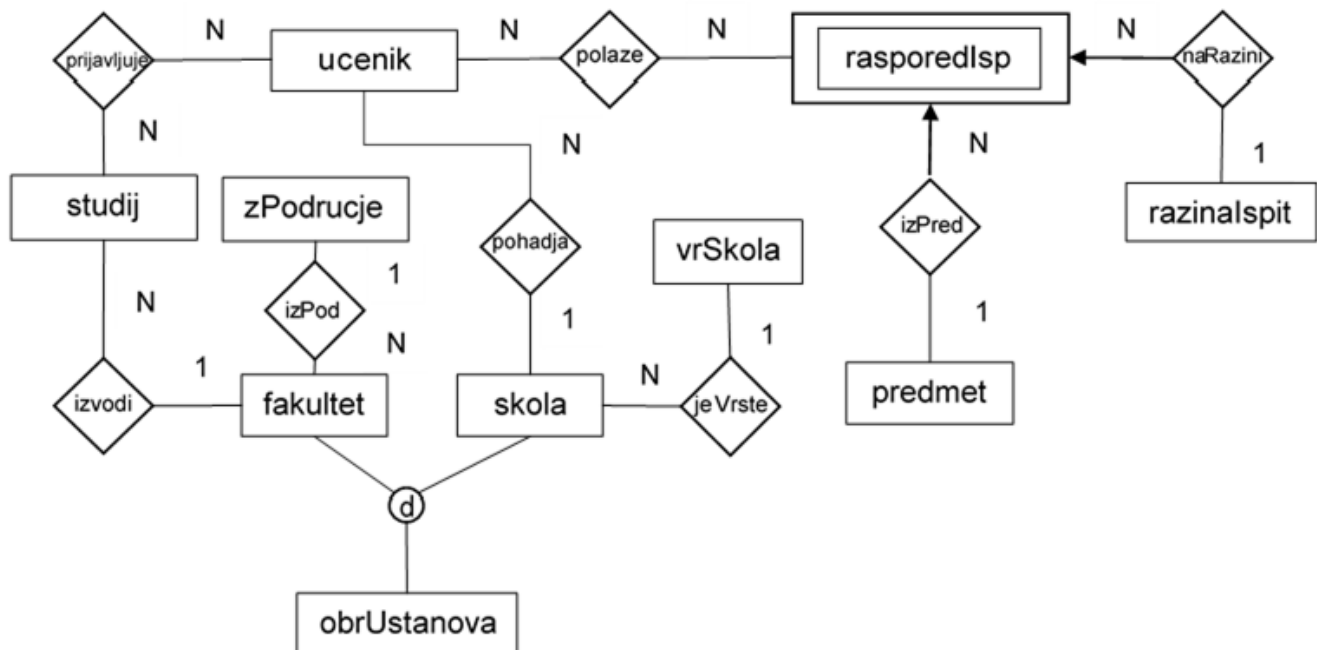
9. (4 boda)
a)



b)



10. (8 bodova)



vrSkola: (sifVrSkole, nazivVrSkole)
 zPodrucje: (sifZPodrucja, nazivZPodrucja)
 obrUstanova: (sifObrUstanova, nazivObrUstanova)
 fakultet: (sifObrUstanova, datAkred)
 skola: (sifObrUstanova, trajanje)

studij: (sifStudij, nazivStudij, trajanje)
 predmet: (sifPredmet, nazivPredmet)
 razinaIspit: (sifRazinaIspit, nazivRazinaIspit)
 ucenik: (OIBUcenik, ime, prezime)
 rasporedIsp: (sifPred, sifRazinaIspit, datIspit)

izvodi: (sifStudij, sifObrUstanova)
 pohadja: (OIBUcenik, sifObrUstanova)
 prijavljuje: (OIBUcenik, sifStudij, prioritet)
 polaze: (OIBUcenik, sifPredmet, sifRazinaIspit, datIspit, ostvarenoBodova, rang)
 naRazini: (sifPredmet, datIspit, sifRazinaIspit)
 izPred: (sifPredmet, datIspit, sifRazinaIspit)
 jeVrste: (sifObrUstanova, sifVrSkole)
 izPod: (sifObrUstanova, sifZPodrucja)