

Međuispit iz Baza podataka

26. travnja 2013.

Zadaci 1 - 8 odnose se na bazu podataka INTERNET_TV prikazanu na **slici 1**. Na slici **nisu** prikazane sve n-torke sadržane u relacijama.

korisnik					serija			
sifKorisnik	prezime	ime	datRod	rejting	sifSerija	nazSerija	godPocEmit	prepDob
11	Perić	Pero	18.10.1978	3	101	Game of Thrones	2011	15
12	Jurić	Đurđa	11.06.1974	2	102	Big Bang Theory	2007	12
13	Jurić	Ivo	05.02.2001	4	103	Sulejman	2011	0
14	Ban	Josip	17.12.1953	3	104	The Wire	2002	18
15	Horvat	Ante	01.04.1973	2				

epizoda					gledanje			
sifEpizoda	rbrSezona	rbrEpizoda	sifSerija	trajanjeMin	sifEpizoda	sifKorisnik	datGled	ocjena
601	1	1	101	52	602	15	13.04.2013	NULL
602	1	2	101	51	604	12	29.03.2013	4
603	2	1	101	49	604	12	09.04.2013	5
604	2	4	102	19	604	15	09.04.2013	NULL
605	3	4	102	21	603	14	06.04.2013	3

Slika 1.

U relacije sa slike 1 pohranjuju se podaci o korisnicima internetske televizije (relacija **korisnik**) te o njihovom gledanju i ocjenjivanju (relacija **gledanje**) pojedinih epizoda (relacija **epizoda**) serija (relacija **serija**). Za svaku se seriju, osim šifre i naziva, evidentira godina početka emitiranja i preporučena najmanja dob gledatelja serije. Korisniku koji često gleda serije raste upisani rejting. Korisnik pojedinu epizodu serije može pogledati više puta, ali samo jednom u istom danu, a nakon svakog gledanja može je ocijeniti ocjenom od 1 do 5. Vrijednost NULL atributa *ocjena* znači da je korisnik odbio dati ocjenu.

U zadacima 1-4 napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

1. Ispisati šifru i naziv serije te datum prvog (najranijeg) gledanja za sve serije čija je godina početka emitiranja 2011., a prosječna ocjena gledatelja je minimalno 3.0. Serije ispisati poredane od one s najkasnijim datumom prvog gledanja, prema onoj s najranijim; u slučaju kad je taj datum istovjetan navesti abecedno po nazivu. **Zadatak riješiti bez podupita.** (2 boda)
2. Za sve serije čija je preporučena dob 18 ispisati šifru, naziv i broj različitih gledatelja rođenih 1990. ili ranije koji su pogledali barem jednu epizodu te serije (ispis i u slučaju kad seriju nije pogledao nitko iz opisane dobne skupine!). **Zadatak riješiti bez podupita.** (2.5 boda)
3. Ispisati šifru i naziv svake serije čija je ovogodišnja minutaža gledanja vikendom veća od minutaže gledanja tijekom svih pet radnih dana. (2.5 boda)
4. Smanjiti za 1 rejting korisnika koji su u tekućoj godini pogledali manje od 3 različite serije ili im je ukupna minutaža gledanih serija u tom vremenu bila manja od 100 minuta. (2.5 boda)
5. Napisati **izraz relacijske algebre (ne SQL upit)** koji odgovara sljedećem: za korisnike koji su ukupno odgledali više od 200 minuta programa ispisati njihovu šifru i spomenutu minutažu odgledanog programa. (2 boda)

U zadacima 6 i 7, uz pretpostavku da slika 1 prikazuje sve podatke pohranjene u bazi podataka, u **obliku tablice prikazati rezultate** obavljanja sljedećih operacija:

6. $[\pi_{\text{prezime, ime}} (\sigma_{\text{godPocEmit}=2011} (\text{serija} \bowtie \text{epizoda} \bowtie \text{gledanje} \bowtie \text{korisnik}))]$
 $\setminus [\pi_{\text{prezime, ime}} (\sigma_{\text{godPocEmit}=2007} (\text{serija} \bowtie \text{epizoda} \bowtie \text{gledanje} \bowtie \text{korisnik}))]$ (1 bod)
7. $\pi_{\text{sifKorisnik, ocjena}} (\sigma_{\text{rejting} \geq 3} (\text{korisnik} * \bowtie \text{gledanje}))$ (1 bod)

8. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B⁺ stabla) svih dolje navedenih upita. (2 boda)

- 1) SELECT * FROM korisnik WHERE datRod BETWEEN '01.01.1980.' AND '01.01.1990.';
- 2) SELECT * FROM korisnik WHERE rejting < 4;
- 3) SELECT * FROM korisnik WHERE prezime LIKE 'H%' AND ime LIKE 'H%';
- 4) SELECT * FROM korisnik ORDER BY datRod ASC, prezime DESC;
- 5) SELECT * FROM korisnik ORDER BY ime, prezime;
- 6) SELECT * FROM korisnik ORDER BY sifKorisnik, rejting;

9. Koji uvjet mora biti zadovoljen da bi relacije r(R) i s(S) bile unijski kompatibilne? Koje se operacije relacijske algebre mogu obaviti s operandima r(R) i s(S) onda i samo onda kada su relacije r(R) i s(S) unijski kompatibilne? (2 boda)

10. Produkcijaska kuća pohranjuje podatke o TV kvizovima koje snima. Relacijska shema EPIZODA_KVIZA sadrži sljedeće attribute:

- sifKviz - jedinstvena šifra kviza,
- nazKviz - naziv kviza,
- sifStudio - jedinstvena šifra TV studija,
- velStudio - veličina studija u m²,
- sifVod - jedinstvena šifra voditelja,
- prezVod - prezime voditelja,
- imeVod - ime voditelja,
- rbrSez - redni broj sezone kviza,
- budzetSez - budžet pojedine sezone kviza,
- rbrEpiz - redni broj epizode kviza unutar sezone (počinje od 1 za svaku novu sezonu),
- datSnimEpiz - datum snimanja epizode kviza.

Vrijede sljedeća pravila:

- kviz se uvijek snima u istom TV studiju,
- tijekom sezone kviza ne mijenja se voditelj,
- istog se dana može snimiti više epizoda pojedinog kviza, a svaka se epizoda snimi u jednom danu.

Odrediti ključ relacijske sheme EPIZODA_KVIZA tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF. (4 boda)

11. Relacija KORISNIK (*sifKorisnik*, *prez*, *ime*) sadrži n-torke sa sljedećim vrijednostima atributa *sifKorisnik*: 11, 17, 20, 25, 40, 42, 47. Nacrtati B⁺-stablo reda 5 za atribut *sifKorisnik* tako da popunjenost stabla bude **minimalna**. (2,5 boda)

12. Napisati SQL naredbu koja će kreirati relacije **zabEmisija** i **producent** nad shemama ZABEMISIJA= {*sifZabEm*, *nazZabEm*, *sifProducent*} i PRODUCENT={*sifProducent*, *OIB*, *prezime*, *ime*, *datRod*, *placa*}. Relacija **zabEmisija** zapisuje opće podatke o zabavnim emisijama, a relacija **producent** detalje o producentima zabavnih emisija. Kod kreiranja relacijske sheme smisleno odabrati tipove podataka te osigurati sljedeće: (4 boda)

- zabavna emisija je jedinstveno identificirana šifrom zabavne emisije (*sifZabEm*);
- svaka zabavna emisija mora imati drugačiji naziv;
- producent je jedinstveno identificiran šifrom producenta;
- producent je jedinstveno identificiran svojim OIB-om;
- za svaki zapis relacije **producent** vrijednosti atributa *sifProducent*, *OIB*, *prezime*, *ime* i *datRod* moraju biti poznate, dok vrijednost atributa *placa* ne mora biti poznata;
- za svaki zapis relacije **zabEmisija** moraju biti poznate vrijednosti atributa *sifZabEm* i *nazZabEm*;
- plaća producenta ne smije biti veća od 12000 kuna;
- atribut *sifProducent* u relaciji **zabEmisija** smije poprimiti isključivo vrijednosti istoimenog atributa u relaciji **producent**; ako se izbriše zapis iz relacije **producent**, brišu se i zapisi o zabavnim emisijama koje je on producirao.

13. Zadana je relacijska shema R = { A, B, C, D, E, F } i skup funkcijskih zavisnosti F = { AB → C, A → DE, C → F, E → D }. Ispitati vrijedi li funkcijska zavisnost ABC → DEF. Za svaki korak dokaza napisati pravilo koje se koristi. (2 boda)

Rješenja:

U zadacima 1-8 napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

1. Ispisati šifru i naziv serije te datum prvog (najranijeg) gledanja za sve serije čija je godina početka emitiranja 2011., a prosječna ocjena gledatelja je minimalno 3.0. Serije ispisati poredane od one s najkasnijim datumom prvog gledanja, prema onoj s najranijim; u slučaju kad je taj datum istovjetan navesti abecedno po nazivu. **Zadatak riješiti bez podupita. (2 boda)**

```
SELECT serija.sifSerija, nazSerija, MIN(datGled) najranijeGledanje
FROM serija JOIN epizoda ON serija.sifserija=epizoda.sifserija
      JOIN gledanje ON epizoda.sifepizoda=gledanje.sifepizoda
WHERE godPocEmit=2011
GROUP BY serija.sifSerija, nazSerija
HAVING AVG(ocjena)>=3.0
ORDER BY najranijeGledanje DESC, nazSerija;
```

2. Za sve serije čija je preporučena dob 18 ispisati šifru, naziv i broj različitih gledatelja rođenih 1990. ili ranije koji su pogledali barem jednu epizodu te serije (ispis i u slučaju kad seriju nije pogledao nitko iz opisane dobne skupine!). **Zadatak riješiti bez podupita. (2.5 boda)**

```
SELECT serija.sifserija, nazSerija, COUNT(DISTINCT korisnik.sifkorisnik)
FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
      JOIN korisnik ON gledanje.sifkorisnik=korisnik.sifkorisnik
      AND datrod<='31.12.1990'
      RIGHT JOIN serija ON epizoda.sifserija=serija.sifserija
WHERE prepDob=18
GROUP BY serija.sifserija, nazSerija;
```

```
SELECT serija.sifserija, nazSerija, COUNT(DISTINCT korisnik.sifkorisnik)
FROM serija LEFT JOIN epizoda ON serija.sifserija=epizoda.sifserija
      LEFT JOIN gledanje ON gledanje.sifepizoda=epizoda.sifepizoda
      LEFT JOIN korisnik ON gledanje.sifkorisnik=korisnik.sifkorisnik
      AND datrod<='31.12.1990'
WHERE prepDob=18
GROUP BY serija.sifserija, nazSerija;
```

3. Ispisati šifru i naziv svake serije čija je ovogodišnja minutaža gledanja vikendom veća od minutaže gledanja tijekom svih pet radnih dana. **(2.5 boda)**

```
SELECT serija.sifserija, nazSerija
FROM serija
WHERE
  (SELECT SUM(trajanjemin)
   FROM epizoda JOIN gledanje ON epizoda.sifepizoda=gledanje.sifepizoda
   WHERE WEEKDAY(datgled) IN (0, 6)
   AND YEAR(datgled)=YEAR(TODAY)
   AND epizoda.sifserija=serija.sifserija)
>
  (SELECT SUM(trajanjemin)
   FROM epizoda JOIN gledanje ON epizoda.sifepizoda=gledanje.sifepizoda
   WHERE WEEKDAY(datgled) BETWEEN 1 AND 5)
  AND YEAR(datgled)=YEAR(TODAY)
  AND epizoda.sifserija=serija.sifserija);

-- može i ... WHERE WEEKDAY(datgled) IN (1, 2, 3, 4, 5))
```

4. Smanjiti za 1 rejting korisnika koji su u tekućoj godini pogledali manje od 3 različite serije ili im je ukupna minutaža gledanih serija u tom vremenu bila manja od 100 minuta. **(2.5 boda)**

Napomena: uvjetima su obuhvaćeni korisnici koji u tekućoj godini nisu pogledali ništa (0 serija, 0 minuta) tj. nema zapisa u relaciji gledanje s njihovom korisničkom šifrom i datumom u tekućoj godini.

```
UPDATE korisnik
SET rejting=rejting-1
WHERE
(
    SELECT COUNT(DISTINCT sifserija)
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
)<3
OR (
    SELECT SUM(trajanjemin)
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
)<100;
```

```
UPDATE korisnik
SET rejting=rejting-1
WHERE sifkorisnik NOT IN (
SELECT sifkorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
GROUP BY sifKorisnik
HAVING COUNT(DISTINCT sifserija)>=3 AND SUM(trajanjemin)>=100
);
```

```
UPDATE korisnik
SET rejting=rejting-1
WHERE NOT EXISTS (
    SELECT sifKorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
GROUP BY sifKorisnik
HAVING COUNT(DISTINCT sifserija)>=3 AND SUM(trajanjemin)>=100
);
```

```
-- iduća dva upita smanjili bi rejting za korisnike koji jesu pogledali bar
-- jednu epizodu neke serije, a pritom je zadovoljen uvjet: manje od tri
-- različite serije ili manje od 100 minuta; rejting se ne bi smanjio
-- korisnicima koji u tekućoj godini nisu pogledali ništa
```

```
UPDATE korisnik
SET rejting=rejting-1
WHERE sifkorisnik IN(
    SELECT sifkorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
GROUP BY sifKorisnik
HAVING COUNT(DISTINCT sifserija)<3 OR SUM(trajanjemin)<100
);
```

```

UPDATE korisnik
SET rejting=rejting-1
WHERE EXISTS (
    SELECT sifKorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
    GROUP BY sifKorisnik
    HAVING COUNT(DISTINCT sifserija)<3 OR SUM(trajanjemin)<100
);

```

5. Napisati **izraz relacijske algebre** (ne SQL upit) koji odgovara sljedećem: za korisnike koji su ukupno odgledali više od 200 minuta programa ispisati njihovu šifru i spomenutu minutažu odgledanog programa. **(2 boda)**

$\sigma_{\text{ukupnoTrajanje} \geq 200} [\rho_{\text{ispis(sifKorisnik, ukupnoTrajanje)}}[\sigma_{\text{SUM(trajanjeMin)((gledanje \triangleright \triangleleft \text{epizoda})})}]]$

U zadacima 6 i 7, uz pretpostavku da slika 1 prikazuje sve podatke pohranjene u bazi podataka, u **obliku tablice prikazati rezultate** obavljanja sljedećih operacija:

6. $[\pi_{\text{prezime, ime}}(\sigma_{\text{godPocEmit}=2011}(\text{serija} \triangleright \triangleleft \text{epizoda} \triangleright \triangleleft \text{gledanje} \triangleright \triangleleft \text{korisnik}))] \setminus [\pi_{\text{prezime, ime}}(\sigma_{\text{godPocEmit}=2007}(\text{serija} \triangleright \triangleleft \text{epizoda} \triangleright \triangleleft \text{gledanje} \triangleright \triangleleft \text{korisnik}))]$ **(1 bod)**

prezime	ime
Ban	Josip

7. $\pi_{\text{sifKorisnik, ocjena}}(\sigma_{\text{rejting} \geq 3}(\text{korisnik} * \triangleright \triangleleft \text{gledanje}))$ **(1 bod)**

sifKorisnik	ocjena
11	NULL
13	NULL
14	3

8. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B⁺ stabla) svih dolje navedenih upita. **(2 boda)**

- 1) SELECT * FROM korisnik WHERE datRod BETWEEN '01.01.1980.' AND '01.01.1990.';
- 2) SELECT * FROM korisnik WHERE rejting < 4;
- 3) SELECT * FROM korisnik WHERE prezime LIKE 'H%' AND ime LIKE 'H%';
- 4) SELECT * FROM korisnik ORDER BY datRod ASC, prezime DESC;
- 5) SELECT * FROM korisnik ORDER BY ime, prezime;
- 6) SELECT * FROM korisnik ORDER BY sifKorisnik, rejting;

```

CREATE INDEX i1 ON korisnik (datRod, prezime DESC)      1 i 4
CREATE INDEX i2 ON korisnik (ime, prezime)              3 i 5
CREATE INDEX i3 ON korisnik (rejting)                   2
CREATE INDEX i4 ON korisnik (sifKorisnik, rejting)      6

```

9. Koji uvjet mora biti zadovoljen da bi relacije $r(R)$ i $s(S)$ bile unijski kompatibilne? **(1 bod)**

Relacije $r(R)$ i $s(S)$ su unijski kompatibilne ako vrijedi:

- relacije su istog stupnja
- korespondentni atributi su definirani nad istim domenama

Koje se operacije relacijske algebre mogu obaviti s operandima $r(R)$ i $s(S)$ onda i samo onda kada su relacije $r(R)$ i $s(S)$ unijski kompatibilne? **(1 bod)**

Unija, razlika, presjek.

10. Produkcijaska kuća pohranjuje podatke o TV kvizovima koje snima. Relacijska shema EPIZODA_KVIZA sadrži sljedeće attribute:

- sifKviz - jedinstvena šifra kviza,
- nazKviz - naziv kviza,
- sifStudio - jedinstvena šifra TV studija,
- velStudio - veličina studija u m^2 ,
- sifVod - jedinstvena šifra voditelja,
- prezVod - prezime voditelja,
- imeVod - ime voditelja,
- rbrSez - redni broj sezone kviza,
- budzetSez - budžet pojedine sezone kviza,
- rbrEpiz - redni broj epizode kviza unutar sezone (počinje od 1 za svaku novu sezonu),
- datSnimEpiz - datum snimanja epizode kviza.

Vrijede sljedeća pravila:

- kviz se uvijek snima u istom TV studiju,
- tijekom sezone kviza ne mijenja se voditelj,
- istog se dana može snimiti više epizoda pojedinog kviza, a svaka se epizoda snimi u jednom danu.

Odrediti ključ relacijske sheme EPIZODA_KVIZA tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF. **(4 boda)**

1NF

$K = \{sifKviz, rbrSez, rbrEpiz\}$

2NF

KVIZ (sifKviz, nazKviz, sifStudio, velStudio) $K = \{sifKviz\}$

SEZONA (sifKviz, brSez, budzetSez sifVod, prezVod, imeVod) $K = \{sifKviz, brSez\}$

EPIZODA (sifKviz, rbrSez, rbrEpiz, datSnimEpiz) $K = \{sifKviz, rbrSez, rbrEpiz\}$

3NF

KVIZ2 (sifKviz, nazKviz, sifStudio) $K = \{sifKviz\}$

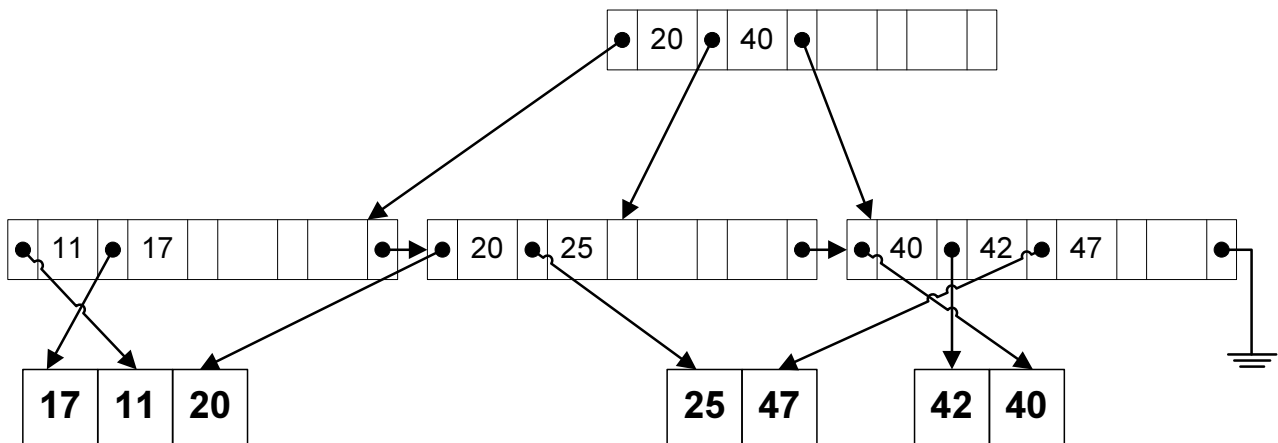
STUDIO (sifStudio, velStudio) $K = \{sifStudio\}$

SEZONA2 (sifKviz, brSez, budzetSez sifVod) $K = \{sifKviz, brSez\}$

VODITELJ (sifVod, prezVod, imeVod) $K = \{sifVod\}$

EPIZODA (sifKviz, rbrSez, rbrEpiz, datSnimEpiz) $K = \{sifKviz, rbrSez, rbrEpiz\}$

11. Relacija KORISNIK (*sifKorisnik*, *prez*, *ime*) sadrži n-torke sa sljedećim vrijednostima atributa *sifKorisnik*: 11, 17, 20, 25, 40, 42, 47. Nacrtati B⁺-stablo reda 5 za atribut *sifKorisnik* tako da popunjenost stabla bude minimalna. (2,5 boda)



12. Napisati SQL naredbu koja će kreirati relacije **zabEmisija** i **producent** nad shemama ZABEMISIJA= {*sifZabEm*, *nazZabEm*, *sifProducent*} i PRODUCENT={*sifProducent*, *OIB*, *prezime*, *ime*, *datRod*, *placa*}. Relacija **zabEmisija** zapisuje opće podatke o zabavnim emisijama, a relacija **producent** detalje o producentima zabavnih emisija. Kod kreiranja relacijske sheme smisleno odabrati tipove podataka te osigurati sljedeće: (4 boda)

- zabavna emisija je jedinstveno identificirana šifrom zabavne emisije (*sifZabEm*);
- svaka zabavna emisija mora imati drugačiji naziv;
- producent je jedinstveno identificiran šifrom producenta;
- producent je jedinstveno identificiran svojim OIB-om;
- za svaki zapis relacije **producent** vrijednosti atributa *sifProducent*, *OIB*, *prezime*, *ime* i *datRod* moraju biti poznate, dok vrijednost atributa *placa* ne mora biti poznata;
- za svaki zapis relacije **zabEmisija** moraju biti poznate vrijednosti atributa *sifZabEm* i *nazZabEm*;
- plaća producenta ne smije biti veća od 12000 kuna;
- atribut *sifProducent* u relaciji **zabEmisija** smije poprimiti isključivo vrijednosti istoimenog atributa u relaciji **producent**; ako se izbriše zapis iz relacije **producent**, brišu se i zapisi o zabavnim emisijama koje je on producirao.

```

CREATE TABLE producent (
    sifProducent INTEGER PRIMARY KEY,
    OIB DECIMAL(11) NOT NULL UNIQUE,
    prezime NCHAR(40) NOT NULL,
    ime NCHAR(40) NOT NULL,
    datRod DATE NOT NULL,
    placa DECIMAL(7,2) CHECK (placa <=12000)
);

```

```

CREATE TABLE zabEmisija (
    sifZabEm INTEGER PRIMARY KEY,
    nazZabEm NCHAR(50) NOT NULL UNIQUE,
    sifProducent INTEGER REFERENCES producent(sifProducent) ON DELETE CASCADE
);

```

13. Zadana je relacijska shema $R = \{ A, B, C, D, E, F \}$ i skup funkcijskih zavisnosti $F = \{ AB \rightarrow C, A \rightarrow DE, C \rightarrow F, E \rightarrow D \}$. Ispitati vrijedi li funkcijska zavisnost $ABC \rightarrow DEF$. Za svaki korak dokaza napisati pravilo koje se koristi. **(2 boda)**

1. $ABC \rightarrow ABC$ (refleksivnost):
2. $ABC \rightarrow ABC \wedge A \rightarrow DE \Rightarrow ABC \rightarrow ABCDE$ (akumulacija)
 $ABC \rightarrow ABCDE \wedge C \rightarrow F \Rightarrow ABC \rightarrow ABCDEF$ (akumulacija)
3. $ABC \rightarrow ABCDEF \Rightarrow ABC \rightarrow DEF$ (dekompozicija):

Može i obrnutim redom kod akumulacije

1. $ABC \rightarrow ABC$ (refleksivnost):
2. $ABC \rightarrow ABC \wedge C \rightarrow F \Rightarrow ABC \rightarrow ABCF$ (akumulacija)
 $ABC \rightarrow ABCF \wedge A \rightarrow DE \Rightarrow ABC \rightarrow ABCDEF$ (akumulacija)
3. $ABC \rightarrow ABCDEF \Rightarrow ABC \rightarrow DEF$ (dekompozicija)