

# 1. Kontrolna zadaća iz Uvoda u baze podataka

20. studenog 2002.

1. Navedite pravilo dekompozicije (pravilo o projektivnosti) i napišite dokaz tog pravila. (4 boda)
2. a) Navedite moguće strategije održavanja referencijskog integriteta pri brisanju n-torke.  
b) Koje od navedenih strategija se ne smiju koristiti ukoliko se briše n-torka čiji je primarni ključ strani ključ neke druge relacije i ujedno dio njezinog primarnog ključa. (4 boda)
3. Među učenicima koji završavaju srednju školu i žele se upisati na fakultet provedena je anketa. Evidentiraju se podaci o srednjoj školi, fakultetu, podaci o učenicima, te fakultetu na kojeg se žele upisati. U bazi podataka nalaze se relacije (**primarni ključevi relacija su podcrtani**):

## MJESTO

<u>pbr</u>	integer	not null	poštanski broj
nazMjesto	nchar(40)	not null	naziv mjesta

## SKOLA

<u>sifSkola</u>	integer	not null	šifra škole
nazSkola	nchar(60)	not null	naziv škole
pbrSkola	integer	not null	poštanski broj mjesta u kojem se nalazi škola

## FAKULTET

<u>sifFakultet</u>	integer	not null	šifra fakulteta
nazFakultet	nchar(60)	not null	naziv fakulteta
pbrFakultet	integer	not null	poštanski broj mjesta u kojem se nalazi fakultet
adresaFakultet	nchar(60)		adresa fakulteta
upisnaKvota	smallint		upisna kvota (kapacitet upisa)

## UCENIK

<u>jmbg</u>	char(13)	not null	JMBG učenika
prezUcenik	nchar(25)	not null	prezime učenika
imeUcenik	nchar(25)	not null	ime učenika
sifSkola	integer	not null	šifra srednje škole koju učenik pohađa
sifFakultet	integer	not null	šifra fakulteta na kojeg se učenik želi upisati
opciUspjeh	decimal(3,2)	not null	opći uspjeh (ocjena 2-5)

 Napisati *po jednu* SQL naredbu za svaki sljedeći zadatak:

- a) Ispisati ukupan broj i prosječan opći uspjeh učenika koji pohađaju srednju školu u mjestima čiji naziv započinje slovima S do Z. (2 boda)
  - b) Ispisati sve podatke o mjestima u kojima ne postoji niti jedna srednja škola koju pohađa neki anketirani učenik. (3 boda)
  - c) Ispisati jmbg, ime i prezime učenika, naziv mjesta srednje škole i mjesta fakulteta za učenike koji ne pohađaju srednju školu u istom mjestu u kojem se nalazi fakultet na kojeg se žele upisati. Ne koristiti podupite. (3 boda)
  - d) Ispisati šifru, prvih 20 znakova naziva i duljinu naziva fakulteta za koje je interes dvostruko veći od kapaciteta (tj. na fakultet se želi upisati dvostruko više učenika nego što je upisna kvota za fakultet). (3 boda)
  - e) Za svaki fakultet za kojeg nije evidentirana adresa ispisati šifru, naziv, te jmbg, ime i prezime učenika koji se žele na njega upisati. Fakulteti na koje se želi upisati više od jednog učenika naći će se u popisu više puta. U listi se trebaju nalaziti i fakulteti za koje nema zainteresiranih učenika. (3 boda)
4. Zadana je relacijska shema R (A, B, C, D, E, F, G, H, I, J). Pretpostavljeni ključ je atribut A. Postoji skup funkcijskih zavisnosti  $F = \{ ABCD \rightarrow EFGHIJ, CD \rightarrow GHJ, GH \rightarrow J, E \rightarrow F \}$

Normalizirajte relaciju na:

a) 1 NF (1 NF ostvariti proširenjem ključa relacije R)	(2 boda)
b) 2 NF	(2 boda)
c) 3 NF	(2 boda)

5. Zadane su relacije:

$r$	( A	B	C )	$s$	( B	C )
	null	null	4		2	3
	4	4	2		2	1
	null	2	3		3	1
	1	1	null		1	null
	2	1	null			
	1	2	3			

Ispišite rezultate obavljanja operacija pod a), b) i c). SQL upite ne treba pisati.

a)  $\pi_{B,C}(r) \cap s$  (2 boda)

b)  $\sigma_{A=B}(r) \bowtie_{r.C > s.C} s$  (3 boda)

c)  $\text{SELECT SUM(DISTINCT C), COUNT(C) FROM } r$   
 $\text{WHERE } C \neq 2$  (2 boda)

6. Zadana je relacija:

$s$	( D	E	F	G)
	2	1	2	1
	4	4	1	2
	3	4	2	1
	1	2	1	9
	4	2	1	2

Vrijedi li za trenutni sadržaj relacije  $s$  funkcijska zavisnost  $FG \rightarrow D$ . **Objasnite zašto!** (2 boda)

7. Zadane su relacije  $r$ ,  $s$  i  $t$  s pripadnim shemama:  $R(A, B, C)$ ,  $S(A, B)$ ,  $T(B, C)$ . U relacijama **nema** NULL vrijednosti. Napišite jednu SQL naredbu kojom će se obaviti operacija:

$\pi_{A,B}(r \bowtie t) \cap s$  (3 boda)

# 1. Kontrolna zadaća iz Uvoda u baze podataka – rješenja

20. studenog 2002.

1) PRAVILO DEKOMPOZICIJE (pravilo o projektivnosti): predavanja!

2) a) Moguće strategije su:

- ciljna n-torka se ne može brisati ako postoje odgovarajuće pozivajuće n-torke
- obavlja se brisanje ciljne n-torke uz kaskadno brisanje svih pozivajućih n-torki kojima je vrijednost stranog ključa jednaka vrijednosti primarnog ključa ciljne n-torke
- postavljanje vrijednosti stranog ključa u pozivajućoj n-torki na NULL vrijednost

b) Nije moguće postavljanje vrijednosti stranog ključa u pozivajućoj n-torki na NULL vrijednost.

3) a) `SELECT COUNT(*), AVG(opciUspjeh)`

```
FROM ucenik, skola, mjesto
WHERE ucenik.sifSkola = skola.sifSkola
AND skola.pbrSkola = mjesto.pbr
AND mjesto.nazMjesto MATCHES '[S-Z]*'
```

b) `SELECT * FROM mjesto`

```
WHERE pbr NOT IN (SELECT DISTINCT pbrSkola FROM skola, ucenik
WHERE skola.sifSkola = ucenik.sifSkola)
```

c) `SELECT jmbg, imeUcenik, prezUcenik, mjestoSkola.nazMjesto, mjestoFakultet.nazMjesto`

```
FROM ucenik, skola, fakultet, mjesto mjestoSkola, mjesto mjestoFakultet
WHERE ucenik.sifSkola = skola.sifSkola
AND ucenik.sifFakultet = fakultet.sifFakultet
AND fakultet.pbrFakultet = mjestoFakultet.pbr
AND skola.pbrSkola = mjestoSkola.pbr
AND fakultet.pbrFakultet <> skola.pbrSkola
```

d) `SELECT sifFakultet, SUBSTRING (nazFakultet from 1 for 20), LENGTH (nazFakultet)`

```
FROM fakultet
WHERE upisnaKvota*2 = (SELECT COUNT(*) FROM ucenik
WHERE ucenik.sifFakultet = fakultet.sifFakultet)
```

e) `SELECT fakultet.sifFakultet, nazFakultet, jmbg, imeUcenik, prezUcenik`

```
FROM fakultet LEFT OUTER JOIN ucenik
ON ucenik.sifFakultet = fakultet.sifFakultet
WHERE adresaFakultet IS NULL
```

4) a)  $ABCD \rightarrow EFGHIJ$        $R_1 (A, B, C, D, E, F, G, H, I, J)$        $K_{R1} = \{A, B, C, D\}$

b)  $CD \rightarrow GHJ$        $R_{21} (C, D, G, H, J)$        $K_{R21} = \{C, D\}$   
 $ABCD \rightarrow EFI$        $R_{22} (A, B, C, D, E, F, I)$        $K_{R22} = \{A, B, C, D\}$

c)  $GH \rightarrow J$        $R_{31} (G, H, J)$        $K_{R31} = \{G, H\}$   
 $CD \rightarrow GH$        $R_{32} (C, D, G, H)$        $K_{R32} = \{C, D\}$   
 $E \rightarrow F$        $R_{33} (E, F)$        $K_{R33} = \{E\}$   
 $ABCD \rightarrow EI$        $R_{34} (A, B, C, D, E, I)$        $K_{R34} = \{A, B, C, D\}$

5) a) 

B	C
2	3
1	null

c) 

$\frac{SUM(DISTINCT C)}{7}$	$\frac{COUNT(C)}{3}$
-----------------------------	----------------------

b) 

r.A	r.B	r.C	s.B	s.C
4	4	2	2	1
4	4	2	3	1
1	1	null	null	null

- 6) Za relaciju s funkcijska zavisnost  $FG \rightarrow D$  bila bi zadovoljena ako za svaku fg vrijednost od FG

$$\pi_D (\sigma_{F=f \wedge G=g} (s))$$

ima najviše jednu n-torku.

Funkcijska zavisnost  $FG \rightarrow D$  ne vrijedi, jer u 1. i 3. n-torki za istu vrijednost atributa FG, vrijednost atributa D nije ista.

- 7) 

```
SELECT DISTINCT r.A, r.B
FROM r, t
WHERE r.B = t.B
AND r.C = t.C
AND EXISTS (SELECT * FROM s
            WHERE r.A = s.A
            AND r.B = s.B)
```

**UVOD U BAZE PODATAKA - 1. Međuispit**  
**26. studeni 2003.**

1. Zadane su relacije **r** i **s** s pripadnim shemama **R** i **S**. U obliku izraza predikatnog računa definirajte sljedeće operacije:

a)  $r \bowtie_{A \theta B} s$  pri čemu je  $A \in R, B \in S$  (theta-pridruživanje ili spajanje uz uvjet)

b)  $r \bowtie s$  (prirodno pridruživanje ili prirodno spajanje) **(4 boda)**

2. Uz pretpostavku da na relaciji **r** (A, B, C, D, E, F) vrijede funkcijske zavisnosti iz skupa **F**

$$F = \{ A \rightarrow EDF, AB \rightarrow ECF, D \rightarrow CB \}$$

dokažite da na relaciji **r** vrijedi funkcijska zavisnost  $AF \rightarrow ECF$ . Za dokaz možete koristiti Armstrongove aksiome i pravila koja iz njih proizlaze, osim pravila o pseudotranzitivnosti.

**Napomena:** dokaz mora biti jasno napisan, kao niz izjava oblika:

- prema **p**, iz **x** i **y**  $\Rightarrow$  **w**
- ...

pri čemu je **p** naziv nekog pravila ili aksioma, **x**, **y**, **w** su neke funkcijske zavisnosti. Definicije pravila i aksioma nije potrebno pisati, dovoljno je navesti samo naziv.

**(4 boda)**

3. Zadana je relacijska shema **R** (A, B, C, D, E, F, G, H). Pretpostavljeni ključ je atribut **A**. Postoji skup funkcijskih zavisnosti  $F = \{ AB \rightarrow EFGH, ABC \rightarrow DEFG, EF \rightarrow GH, EF \rightarrow AB \}$

Normalizirajte relaciju na

- a) 1 NF (1 NF ostvariti BEZ rastavljanja sheme relacije **R**, proširenjem ključa)
- b) 2 NF
- c) 3 NF

**(4 boda)**

4. Zadane su relacije:

$r(A \quad B)$	$s(B \quad C \quad E)$	$t(E \quad F)$
a b	b 3 1	1 10
a m	null 5 null	10 null
c b	c null 3	1 9
d null		null 9
d c		null null
null null		null 1

Napišite rezultate obavljanja sljedećih operacija. SQL upite nije potrebno pisati

a)  $\pi_B(r) \cap \pi_B(s)$

b)  $\sigma_{E+F < 10}(t)$

c)  $s * \bowtie t$  **(6 bodova)**

5. Zadane su relacije s pripadnim shemama: **r** (A, B, C, D), **s** (B, C) i **t** (C, D, E). Svaki zadatak riješite uporabom jedne SQL naredbe (u relacijama nema NULL vrijednosti)

a)  $\sigma_{A=E}(r \bowtie t)$

b)  $\pi_{A,B}(r) \setminus \pi_{A,B}(r \bowtie s)$  **(6 bodova)**

→ Sljedeća stranica

6. U bazi podataka LETOVI prate se podaci o letovima aviona. Podvučeni atributi čine primarni ključ relacije. Vrijednost niti jednog atributa ne može poprimiti NULL vrijednost.

**AVION** - opisuje pojedinačne avione

<u>sifAvion</u>	integer	šifra aviona
tipAvion	char(15)	oznaka tipa aviona
maxDolet	smallint	najveća udaljenost ( <b>u kilometrima</b> ) koju avion može preletjeti bez punjenja goriva

**LET** - koji avion je letio kamo i kada

<u>sifLet</u>	integer	šifra leta
datum	date	datum leta
vrijemePol	char(5)	vrijeme polijetanja u obliku SS:MM (sati i minute)
sifAvion	integer	šifra aviona
sifAerPol	integer	šifra aerodroma polijetanja
sifAerSlij	integer	šifra aerodroma slijetanja

**AERODROM** - opisuje aerodrome

<u>sifAer</u>	integer	šifra aerodroma
nazAer	char(20)	naziv aerodroma

**UDALJENOST** - udaljenost upisana za svaki par aerodroma: npr. PLESO-DUBROVNIK i DUBROVNIK-PLESO

<u>sifAerOd</u>	integer	šifra aerodroma
<u>sifAerDo</u>	integer	šifra aerodroma
udalj	smallint	udaljenost ( <b>u kilometrima</b> ) između aerodroma sa šifrom sifAerOd do aerodroma sa šifrom sifAerDo

Napisati **po jedan** SQL upit za svaki od zadataka:

- Ispisati broj različitih tipova aviona koji su letjeli tijekom zadnjih 15 dana i ukupnu udaljenost izraženu u nautičkim miljama koju su pri tome preletjeli. Upit mora biti neovisan o datumu izvođenja. (1 nautička milja = 1.852 km). **Uporaba podupita nije dopuštena.**
- Ispisati popis letova (šifru i tip aviona, datum i vrijeme polijetanja) za koje je aerodrom polijetanja bio **Only** ili **Gatwick**, a aerodrom slijetanja **LaGuardia**. **Uporaba podupita nije dopuštena.**
- Ispisati popis aviona (šifru i tip aviona) koji su tijekom prošle godine (prošle u odnosu na tekuću godinu) s aerodroma **Le Bourget** barem jednom poletjeli na puni ili polovicu sata (u 00:00 ili u 00:30 ili 01:00 ili 01:30, itd.). Upit mora biti neovisan o datumu izvođenja. **Uporaba podupita nije dopuštena.**
- Ispisati popis aviona (samo šifru aviona) koji mogu bez ponovnog punjenja goriva preletjeti udaljenost između svaka dva aerodroma u jednom i drugom smjeru (dakle "tamo i natrag" bez punjenja goriva). Treba uzeti u obzir da, zbog korištenja različitih koridora letenja, udaljenost između aerodroma X i Y ne mora biti jednaka udaljenosti između aerodroma Y i X.

**(12 bodova)**

7. Navedite nazive (**nije potrebno pisati definicije**) općih pravila integriteta. Za svako od navedenih pravila integriteta navedite po jedan primjer temeljen na relaciji UDALJENOST iz prethodnog zadatka. **Ne treba** pisati SQL naredbe za implementaciju tih pravila.

**(4 boda)**

**UVOD U BAZE PODATAKA - 1. Međuispit**  
**26. studeni 2003.**

**Rješenja:**

1. a)  $r \bowtie s = \{ t_r \wedge t_s \mid t_r \in r \wedge t_s \in s \wedge t_r(A) \theta t_s(B) \}$   
 $A \theta B$   
 b)  $r \bowtie s = q(R \cup S) = \{ t \mid t_r \in r \wedge t_s \in s, t(R) = t_r \wedge t(S) = t_s \}$

2.  
 prema pravilu o dekompoziciji:  $A \rightarrow EDF \Rightarrow A \rightarrow D$   
 prema aksiomu o tranzitivnosti:  $A \rightarrow D \wedge D \rightarrow CB \Rightarrow A \rightarrow CB$   
 prema pravilu o uniji:  $A \rightarrow EDF \wedge A \rightarrow CB \Rightarrow A \rightarrow EDFCB$   
 prema pravilu o dekompoziciji:  $A \rightarrow EDFCB \Rightarrow A \rightarrow ECF$   
 prema aksiomu o uvećanju:  $A \rightarrow ECF \Rightarrow AF \rightarrow ECF$

3. 1NF:  $R = ABCDEFGH$   $K = ABC$   
 2NF:  $R_1 = ABCD$   $K_1 = ABC$   
 $R_2 = ABEFGH$   $K_2 = AB$   
 3NF:  $R_1 = ABCD$   $K_1 = ABC$   
 $R_2 = ABEFGH$   $K_{21} = AB$   $K_{22} = EF$

4.a) **B**  
 ===  
 b  
 null  
 c

b) **E F**  
 =====  
 1 10

c) **B C E E' F**  
 =====  
 b 3 1 1 10  
 b 3 1 1 9  
 null 5 null null null  
 c null 3 null null

5.a) `SELECT r.*, t.E  
 FROM r, t  
 WHERE r.C = t.C  
 AND r.D = t.D  
 AND r.A = t.E`

b) `SELECT DISTINCT A, B FROM r rOut  
 WHERE NOT EXISTS  
 (SELECT DISTINCT r.A, r.B FROM r, s  
 WHERE r.B = s.B  
 AND r.C = s.C  
 AND r.A = rOut.A  
 AND r.B = rOut.B)`

- 6.a)**

```
SELECT COUNT(DISTINCT tipAvion)
      , SUM(udalj)/1.852
FROM avion, let, udaljenost
WHERE avion.sifAvion = let.sifAvion
      AND let.sifAerPol = udaljenost.sifAerOd
      AND let.sifAerSlij = udaljenost.sifAerDo
      AND datum > TODAY - 15
```
- b)**

```
SELECT avion.sifAvion, tipAvion, datum, vrijemePol
FROM avion, let, aerodrom aPol, aerodrom aSlij
WHERE avion.sifAvion = let.sifAvion
      AND let.sifAerPol = aPol.sifAer
      AND let.sifAerSlij = aSlij.sifAer
      AND (aPol.nazAer = 'Orly' OR aPol.nazAer = 'Gatwick')
      AND aSlij.nazAer = 'LaGaurdia'
```
- c)**

```
SELECT DISTINCT avion.sifAvion, tipAvion
FROM avion, let, aerodrom
WHERE avion.sifAvion = let.sifAvion
      AND let.sifAerPol = aerodrom.sifAer
      AND aerodrom.nazAer = 'Le Bourget'
      AND (vrijemePol LIKE '%:00' OR vrijemePol LIKE '%:30')
      AND YEAR(datum) = YEAR(TODAY) -1
```
- d)**

```
SELECT sifAvion
FROM avion
WHERE maxDolet >= ALL (
SELECT tamo.udalj + natrag.udalj
FROM udaljenost tamo, udaljenost natrag
WHERE tamo.sifAerDo = natrag.sifAerOd
      AND natrag.sifAerDo = tamo.sifAerOd)
```

## 7. Pravilo entitetskog integriteta

vrijednosti atributa *sifAerOd* i *sifAerDo* ne smiju poprimiti NULL vrijednost niti za jednu n-torku.

### Pravilo referencijskog integriteta

svaka vrijednost atributa *sifAerOd* i *sifAerDo* mora biti jednaka vrijednosti primarnog ključa neke n-torke iz relacije aerodrom



**UVOD U BAZE PODATAKA – 1. međuispit**  
**24. studeni 2004.**

1. Zadana je relacijska shema  $R(A, B, C, D)$ . Navedite primjer sadržaja relacije  $r(R)$  na kojoj vrijedi funkcijska zavisnost  $AB \rightarrow CD$ , a ne vrijede funkcijske zavisnosti  $A \rightarrow CD$  i  $B \rightarrow CD$ . Relacija treba sadržavati 4 do 5 n-torki. (2 boda)

2. Relacijska shema CDTEKA sastoji se od sljedećih atributa:

- sifCD	- šifra CD-a
- nazCD	- naziv CD-a
- cijena	- cijena CD-a
- sifIzd	- šifra izdavača CD-a
- nazIzd	- naziv izdavača CD-a
- sifPj	- šifra pjesme
- nazPj	- naziv pjesme
- sifZanr	- šifra žanra pjesme
- nazZanr	- naziv žanra pjesme
- trajanje	- trajanje pjesme

Pretpostavljeni ključ relacije je sifCD (šifra CD-a). Svaki CD izdaje jedan izdavač. Izdavači sami određuju šifru CD-a (različiti CD-ovi mogu imati iste šifre ako ih izdaju različiti izdavači).

Pjesma ima šifru, naziv i trajanje, a pripada jednom žanru. Šifra pjesme jednoznačno određuje pjesmu.

Na jednom CD-u se može nalaziti više pjesama, ali se ista pjesma može pojaviti samo jednom. Ista se pjesma može pojaviti na različitim CD-ovima bilo kojeg izdavača.

Normalizirajte relaciju DISKOTEKA na:

- 1NF (1 NF ostvariti proširenjem ključa, BEZ rastavljanja sheme relacije CDTEKA)
- 2NF
- 3NF

(6 bodova)

3. Navedite **nazive** i **definicije** pravila koja proizlaze iz Armstrongovih aksioma. (4 boda)

4. Zadane su relacije:

$r(A \quad B)$	$s(A \quad B \quad C \quad D)$	$t(D \quad E)$
a m	b m null null	3 1
null m	a n null 5	5 null
b n	b n 4 9	7 1
null null	a m 6 3	null null
b null	c null 4 5	5 4
b h		

Napišite rezultate obavljanja slijedećih operacija (SQL upite **nije** potrebno pisati):

- $\pi_B(r) \cup \pi_B(s)$
- $\sigma_{C \leq E}(s \bowtie t)$
- $(r \bowtie s) * \bowtie t$

(6 bodova)

5. Zadane su relacije  $r$  i  $s$  s pripadnim shemama  $R$  i  $S$ . U kojem slučaju će rezultat  $r \bowtie s$  biti jednak njihovom Kartezijevom produktu? (2 boda)

6. Zadane su relacije s pripadnim relacijskim shemama:  $r(A, B, C)$ ,  $s(B, D, E)$ ,  $t(C, F)$ . Svaki zadatak riješite **upotrebom jedne SQL naredbe** (u relacijama nema NULL vrijednosti).

a)  $\pi_A (r \bowtie s) \cap \pi_A (r \bowtie t)$

b)  $\sigma_{B>5 \vee A>F} (r \bowtie^* t)$

(6 bodova)

7. Baza podataka sadrži podatke o knjigama. Podvučeni atributi čine primarni ključ relacije. U relacijama nema NULL vrijednosti.

**OSOBA** - opisuje osobe

<u>sifOsoba</u>	INTEGER	šifra osobe
ime	CHAR(30)	ime osobe
prezime	CHAR(30)	prezime osobe

**IZDAVAC** - opisuje izdavače knjiga

<u>sifIzdavac</u>	INTEGER	šifra izdavača
naziv	CHAR(50)	naziv izdavača

**KNJIGA** - opisuje knjiga

<u>sifKnjiga</u>	INTEGER	šifra knjige
naslov	CHAR(100)	naslov knjige (različite knjige mogu imati jednak naslov)
sifIzdavac	INTEGER	šifra izdavača knjige
datum	DATE	datum 1. izdanja knjige

**AUTORI** - opisuje autore knjiga

<u>sifKnjiga</u>	INTEGER	šifra knjige
<u>sifOsoba</u>	INTEGER	šifra osobe koja je autor knjige

Napisati po jedan SQL upit za svaki od zadataka. **Upiti moraju biti neovisni o datumu izvođenja. Neće se priznavati rješenja u kojima se nepotrebno koriste podupiti.**

- a) Ispisati imena, prezimena i inicijale (oblika I.P.) svih osoba koje nisu autori niti jedne knjige. (2 boda)
- b) Ispisati sve podatke o knjigama koje su prvi puta izdane nakon svih knjiga koje je izdao izdavač "Školska knjiga". Pretpostavlja se da je izdavač "Školska knjiga" izdao barem jednu knjigu. (3 boda)
- c) Za sve izdavače ispisati naziv izdavača, ukupan broj knjiga koje je izdao taj izdavač, te broj knjiga tog izdavača izdanih u zadnjih 10 godina. (3 boda)
- d) Ispisati broj **različitih naslova** knjiga koje je u prva 3 mjeseca prošle godine (prošle u odnosu na tekuću godinu) izdao izdavač "Školska knjiga". (3 boda)
- e) Za svaku osobu čije prezime ne počinje samoglasnikom, ispisati šifru, ime i prezime, te naslov knjige čiji je autor. U ispisu se moraju pojaviti i osobe koje nisu autor niti jedne evidentirane knjige: u tom je slučaju za naslov knjige potrebno ispisati tekst 'Nije autor knjige'. (3 boda)

**UVOD U BAZE PODATAKA – rješenja 1. međuispita  
24. studeni 2004.**

1.

r(A)	B	C	D)
b	m	1	1
a	n	1	2
b	c	4	9
a	c	6	3

2.

**1NF:** CDTEKA = (sifCD, siflzd, sifPj, nazCD, cijena, nazlzd, nazPj, sifZanr, nazZanr, trajanje)  
K<sub>CDTEKA</sub>={sifCD, siflzd, sifPj}

**2NF:** CD = (sifCD, siflzd, nazCD, cijena) K<sub>CD</sub>={sifCD, siflzd }

IZDAVAC = (siflzd, nazlzd) K<sub>IZDAVAC</sub>={siflzd}

PJESMA= (sifPj, nazPj, sifZanr, nazZanr, trajanje) K<sub>PJESMA</sub>={sifPj}

CDTEKA1 = (sifCD, siflzd, sifPj) K<sub>CDTEKA1</sub>={sifCD, siflzd, sifPj }

**3NF:** CD = (sifCD, siflzd, nazCD, cijena) K<sub>CD</sub>={sifCD, siflzd }

IZDAVAC = (siflzd, nazlzd) K<sub>IZDAVAC</sub>={siflzd}

PJESMA1= (sifPj, nazPj, sifZanr, trajanje) K<sub>PJESMA1</sub>={sifPj}

ZANR= (sifZanr, nazZanr) K<sub>ZANR</sub>={sifZanr}

CDTEKA1 = (sifCD, siflzd, sifPj) K<sub>CDTEKA1</sub>={sifCD, siflzd, sifPj }

3. Pogledati predavanja...

4.

a)

B
m
n
null
h

b)

A	B	C	D	E
a	m	6	3	1

c)

A	B	C	D	D'	E
a	m	6	3	3	1
b	n	4	9	null	null

5. U slučaju u kojem relacije r i s nemaju istoimene atribute.

6.

a) 

```
SELECT DISTINCT r.a
FROM r,s
WHERE r.b = s.b
AND r.a IN (SELECT DISTINCT rIn.a
            FROM r rIn, t
            WHERE rIn.c = t.c)
```

b) 

```
SELECT *
FROM t LEFT OUTER JOIN r
ON r.c = t.c
WHERE r.b > 5 OR r.a > t.f
```

7.

a) 

```
SELECT ime, prezime
      , SUBSTRING(ime FROM 1 FOR 1) || '.' ||
      SUBSTRING(prezime FROM 1 FOR 1) || '.'
FROM osoba
WHERE osoba.sifOsoba NOT IN (SELECT DISTINCT autori.sifOsoba FROM autori)
```

b) 

```
SELECT knjiga.*
FROM knjiga
WHERE datum > ALL(SELECT datum
                   FROM izdovac, knjiga drugaKnjiga
                   WHERE naziv = 'Školska knjiga'
                   AND drugaKnjiga.sifIzdovac = izdovac.sifIzdovac)
```

ili

```
SELECT knjiga.*
FROM knjiga
WHERE datum > (SELECT MAX(datum)
               FROM izdovac, knjiga drugaKnjiga
               WHERE naziv = 'Školska knjiga'
               AND drugaKnjiga.sifIzdovac = izdovac.sifIzdovac)
```

c) 

```
SELECT naziv,
       (SELECT COUNT(*)
        FROM knjiga
        WHERE knjiga.sifIzdovac = izdovac.sifIzdovac),
       (SELECT COUNT(*)
        FROM knjiga
        WHERE knjiga.sifIzdovac = izdovac.sifIzdovac
        AND YEAR(TODAY) - YEAR(datum) <= 10)
FROM izdovac
```

d) 

```
SELECT COUNT(DISTINCT naslov)
FROM knjiga, izdovac
WHERE knjiga.sifIzdovac = izdovac.sifIzdovac
AND naziv = 'Školska knjiga'
AND YEAR(TODAY)-1 = YEAR(datum)
AND MONTH(datum) <= 3
```

e) 

```
SELECT osoba.sifOsoba, ime, prezime, NVL (naslov, 'Nije autor knjige')
FROM osoba
LEFT OUTER JOIN autori
      INNER JOIN knjiga
      ON autori.sifKnjiga = knjiga.sifKnjiga
      ON autori.sifOsoba = osoba.sifOsoba
WHERE prezime MATCHES '[^AEIOU]*'
```

**UVOD U BAZE PODATAKA – 1. međuispit**  
**23. studeni 2005.**

1. Zadana je relacija **r** s pripadnom relacijskom shemom **R(A, B, C)**. A, B i C su atributi tipa integer. (Npr. (1, 2, 3) može biti n-torka relacije r.)

a) Pretpostavite da vrijedi funkcijska zavisnost **F: B → C**. Napravite primjer sadržaja relacije r(R) koji neće zadovoljavati funkcijsku zavisnost **F**. Primjer mora sadržavati najmanji broj n-torki potreban da se pokaže da funkcijska zavisnost ne vrijedi. (2 boda)

b) Pretpostavite da sadržaj relacije **r** trenutno zadovoljava funkcijsku zavisnost **F: B → C**. Koje od sljedećih operacija ni u kojem slučaju neće narušiti tu funkcijsku zavisnost? Objasnite zašto. (2 boda)

- 1) izmjena vrijednosti atributa A
- 2) izmjena vrijednosti atributa B
- 3) izmjena vrijednosti atributa C
- 4) unos n-torke
- 5) brisanje n-torke

2. Baza podataka sadrži podatke o mečevima teniskog turnira. Podvučeni atributi čine primarni ključ relacije. NULL vrijednosti mogu poprimiti samo atributi uz koje je to specificirano.

**NATJECATELJ** - opisuje natjecatelje

<u>sifNat</u>	INTEGER	šifra natjecatelja
imeNat	CHAR(30)	ime natjecatelja
prezimeNat	CHAR(30)	prezime natjecatelja

**MEC** - opisuje pojedinačni meč

<u>sifMec</u>	SMALLINT	šifra meča
datumMec	DATE	datum održavanja meča
vrijemeMec	CHAR(5)	vrijeme početka meča (npr. 15:00)
trajanjeMec	SMALLINT	trajanje meča u minutama, za meč u tijeku vrijednost atributa je NULL

**NATJ\_MEC** - opisuje natjecatelje u meču (za svaki meč sadrži dva zapisa)

<u>sifMec</u>	SMALLINT	šifra meča
<u>sifNat</u>	INTEGER	šifra natjecatelja
dobioSetova	SMALLINT	broj osvojenih setova u meču

**NATJ\_SET** - opisuje rezultat natjecatelja u svakom setu meča

<u>sifMec</u>	SMALLINT	šifra meča
<u>sifNat</u>	INTEGER	šifra natjecatelja
<u>rbrSet</u>	SMALLINT	redni broj seta
dobioGemova	SMALLINT	broj dobivenih gemova natjecatelja u setu meča

Napisati po jedan SQL upit za svaki od zadataka. **Upiti moraju biti neovisni o datumu izvođenja. Neće se priznavati rješenja u kojima se nepotrebno koriste podupiti.**

a) Ispisati prezime i inicijal imena (oblika: Ljubičić, I.) natjecatelja koji su završili barem jedan meč, ali niti u jednom meču nisu osvojili više od jednog seta. (3 boda)

b) Ukoliko je prosjek trajanja svih mečeva do tri sata (uključivo), ispisati 'kratki', ukoliko je više od tri sata ispisati 'dugi', a ukoliko nije ništa od navedenog ispisati 'nepoznato'. **Objasnite** u kojem će se slučaju za prosjek trajanja svih mečeva ispisati 'nepoznato', uz pretpostavku da relacija **mec** nije prazna (3 boda)

c) Ispisati broj mečeva odigranih u zadnjih sedam dana, koji su trajali barem 4 seta. (2 boda)

d) Za svakog natjecatelja čije ime ne započinje samoglasnikom, a duljina imena ili prezimena je veća od 5, ispisati šifru, prezime, ime, datum meča, vrijeme meča i broj osvojenih setova, ukoliko je taj broj veći od 1. Ukoliko za nekog natjecatelja ne postoji niti jedan meč na kojem je osvojio više od jednog seta, za datum meča, vrijeme meča i broj osvojenih setova ispisati NULL vrijednosti. Napomena: poznato je da za svaku šifru natjecatelja iz relacije **natj\_mec**, sigurno postoji n-torka u relaciji **mec**. Niti jedno ime niti prezime ne započinje prazninom. (3 boda)

3. Zadana je relacijska shema  $R(A, B, C, D, E, F, G)$ . Vrijede funkcijske zavisnosti iz skupa funkcijskih zavisnosti:  $F = \{ABC \rightarrow DEFG, AB \rightarrow DG, BC \rightarrow EF, A \rightarrow G\}$

- a) Ako je  $K_R = \{AB\}$  pretpostavljeni ključ relacijske sheme  $R$  ona nije u 1 NF. Objasnite zašto. (1 boda)
- b) U procesu normalizacije relacijske sheme  $R$  nastala je relacijska shema:  
 $R_1(A, B, C, D, G)$   $K_{R_1} = \{ABC\}$  (2 boda)  
 $R_1$  nije u 2 NF. Navedite razlog(e).

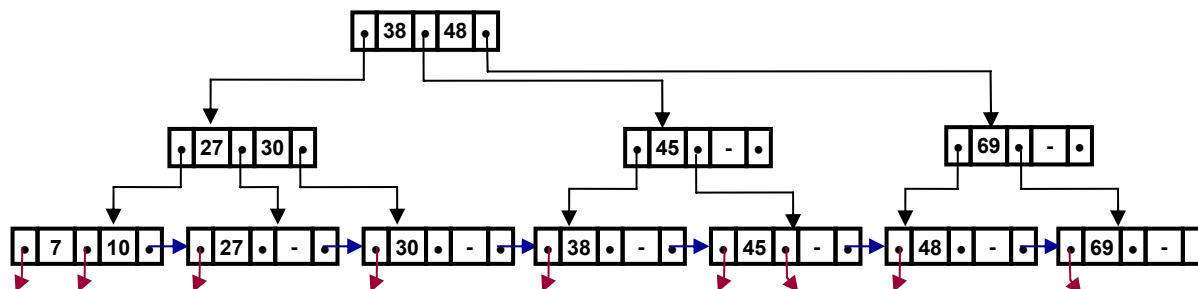
4. Zadane su relacije:

$r(A \quad B \quad C)$	$s(A \quad B \quad C \quad D)$	$t(D \quad E)$
1 X ab	4 Z ac 1.2	5.3 null
2 null null	null Z null 0.5	null 4.6
null X ab	3 null aa null	4.6 1
4 Z null	2 null bb 0.2	5.3 4
4 Z ac	null X ab null	
3 Y aa	1 X ab 5.3	
2 null bb	1 X ab null	

Napišite rezultate obavljanja sljedećih operacija (SQL upite **nije** potrebno pisati):

- a)  $t \bowtie (r \bowtie s)$  (2 boda)
- b)  $\pi_{A,B}(r) \cap \pi_{A,B}(s)$  (2 boda)
- c) Napišite razlog zbog kojeg nije moguće obaviti operaciju  $\pi_B(r) \cup \pi_{B,C}(s)$  (2 boda)

5. a) U  $B^+$ -stablo na slici dodati zapis s ključem 8. (Ne treba crtati sve korake.) (4 boda)



- b) Uz pretpostavku da se radi o  $B^+$ -stablu reda  $n$  ( $n \gg 1$ ) koje sadrži kazaljke na  $m$  zapisa podataka, koliki će biti najmanji mogući broj razina stabla? (3 boda)

6. U bazi podataka kreirane su relacije: Rezervacija(sifraJedrilicar, sifraJedrilica, datum)  
 Jedrilicar (sifraJedrilicar, ime)  
 Jedrilica (sifraJedrilica, naziv, boja)

Primjer sadržaja relacija:

Jedrilicar		Jedrilica			Rezervacija		
sifraJedrilicar	Ime	sifraJedrilica	naziv	boja	sifraJedrilicar	sifraJedrilica	datum
22	Ivan	101	Mornar	crvena	22	101	10.10.2005.
58	Leo	103	Galeb	zelena	58	103	11.10.2005
		104	Adria	zelena			

- a) Sljedeće zadatke riješite uporabom **jedne** SQL naredbe:
- $\pi_{ime}(\sigma_{sifraJedrilica=103}(jedrilica \bowtie rezervacija \bowtie jedrilicar))$  (2 boda)
  - $\pi_{naziv, datum}(\sigma_{boja='zelena'}(jedrilica \bowtie rezervacija))$  (2 boda)
- b) Sljedeće zadatke riješite uporabom **jednog** izraza relacijske algebre:
- Pronađite imena jedrilicara koji su rezervirali i crvenu i zelenu jedrilicu. (3 boda)
  - Pronađite nazive jedrilica koje je rezervirao jedrilicar imena Ivan i nazive jedrilica crvene boje. (2 boda)

**UVOD U BAZE PODATAKA – 1. međuispit - rješenja**  
**23. studeni 2005.**

1. a) (1,2,3) i (1,2,4)  
 b) 1 i 5.  
 1 - ne mijenja vrijednost atributa B i C, tako da ne može utjecati na funkcijsku zavisnost  $B \rightarrow C$ .  
 5 – brisanje n-torke ne može narušiti funkcijsku zavisnost, ako je funkcijska zavisnost već bila zadovoljena.  
 2,3,4 mogu narušiti funkcijsku zavisnost npr.  
 $R = \{(1,2,3), (1,3,4)\}$ , nakon izvođenja SQL naredbe `UPDATE B = 2 WHERE B = 3` dobit će se rezultat  
 $R_1 = \{(1,2,3), (1,2,4)\}$   
 $R = \{(1,2,3), (8,2,3)\}$ , nakon izvođenja SQL naredbe `UPDATE C = 3 WHERE A = 8` dobit će se rezultat  
 $R_1 = \{(1,2,3), (8,2,4)\}$   
 $R = \{(1,2,3)\}$ , nakon izvođenja SQL naredbe `INSERT INTO R VALUES (1,2,4)` dobit će se rezultat  
 $R_1 = \{(1,2,3), (1,2,4)\}$

2. a)  

```
SELECT DISTINCT TRIM(natjecatelj.prezNat) || ', ' ||
                SUBSTRING(imeNat FROM 1 FOR 1) || ' '
FROM natjecatelj, mec, natj_mec
WHERE natjecatelj.sifNat = natj_mec.sifNat
      AND natj_mec.sifMec = mec.sifMec
      AND mec.trajanjeMec IS NOT NULL
      AND sifNat NOT IN (SELECT DISTINCT sifNat
                        FROM natj_mec
                        WHERE dobioSetova > 1)
```

- b)  

```
SELECT CASE WHEN AVG(trajanjeMec) <= 180 THEN 'kratki'
            WHEN AVG(trajanjeMec) > 180 THEN 'dugi'
            ELSE 'nepoznato'
        END CASE
FROM mec
```

Za prosjek trajanja svih mečeva ispisat će se 'nepoznato' kada su svi mečevi u toku (`mec.trajanjeMec IS NULL`).

- c)  

```
SELECT COUNT(DISTINCT mec.sifMec)
FROM mec, natj_set
WHERE mec.sifMec = natj_mec.sifMec
      AND mec.datumMec > TODAY -7
      AND natj_set.rbrSet = 4
```

- d)  

```
SELECT natjecatelj.sifNat, imeNat, prezNat, mec.datumMec
      , mec.vrijemeMec, natj_mec.dobioSetova
FROM natjecatelj
      LEFT OUTER JOIN natj_mec
            INNER JOIN mec
                  ON natj_mec.sifMec = mec.sifMec
            ON natjecatelj.sifNat = natj_mec.sifNat AND natj_mec.dobioSetova > 1
WHERE imeNat MATCHES '[^AEIOU]*'
      AND (LENGTH (imeNat) > 5 OR LENGTH (prezNat) > 5)
```

3. a) R nije uz 1NF jer pretpostavljeni ključ  $K_R = \{AB\}$  ne određuje attribute C, E i F tj. nisu svi zavisni atributi funkcijski ovisni o ključu.  
 b)  $R_1$  nije u 2NF zato što atributi D i G nisu potpuno funkcijski ovisni o cijelom ključu ABC. D je potpuno funkcijski ovisan o AB a G je potpuno funkcijski ovisan o A.

4. a)  $r \bowtie s$ 
 $t \bowtie (r \bowtie s)$ 

A	B	C	D
1	X	ab	5.3
1	X	ab	null
4	Z	ac	1.2

D	E	A	B	C	D
5.3	null	1	X	ab	5.3
null	4.6	null	null	null	null
4.6	1	null	null	null	null
5.3	4	1	X	ab	5.3

 b)  $\pi_{A,B}(r)$ 

A	B
1	X
2	null
null	X
4	Z
3	Y

 $\pi_{A,B}(s)$ 

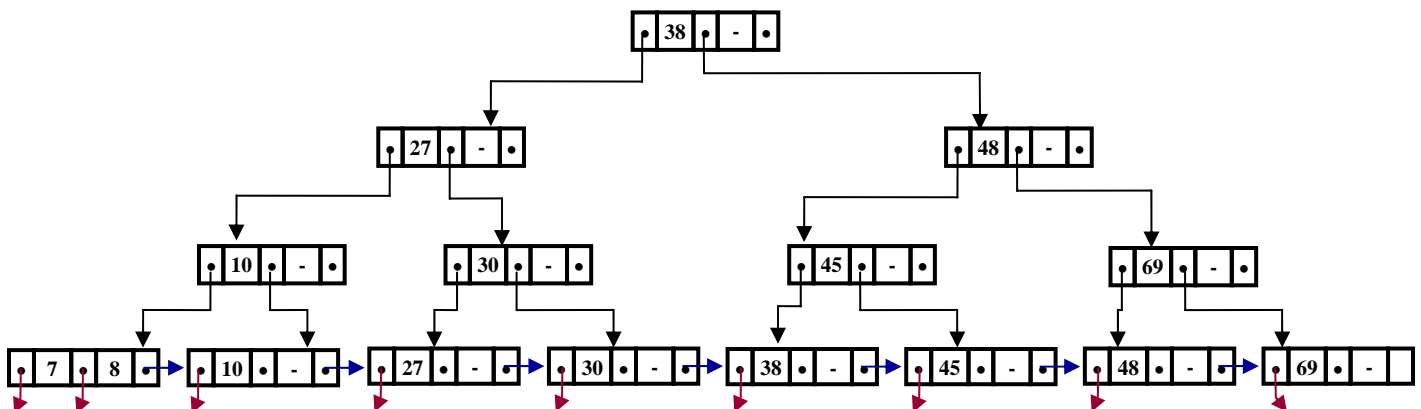
A	B
4	Z
null	Z
3	null
2	null
null	X
1	X

 $\pi_{A,B}(r) \cap \pi_{A,B}(s)$ 

A	B
1	X
2	null
null	X
4	Z

c) Da bi se operacija unije ( $\cup$ ) mogla obaviti nad relacijama one moraju biti unijski kompatibilne (istog stupnja; korespondentni atributi definirani nad istim domenama).  $\pi_B(r)$  je stupnja 1, a  $\pi_{B,C}(s)$  je stupnja 2.

5. a)



b) Pogledati u predavanjima.

6. a) `SELECT DISTINCT ime  
FROM jedrilica, rezervacija, jedrilicar  
WHERE sifraJedrilica = 103  
AND jedrilica.sifraJedrilica = rezervacija.sifraJedrilica  
AND jedrilicar.sifraJedrilicar = rezervacija.sifraJedrilicar`

`SELECT DISTINCT naziv, datum  
FROM jedrilica  
LEFT OUTER JOIN rezervacija  
ON jedrilica.sifraJedrilica = rezervacija.sifraJedrilica  
WHERE jedrilica.boja = 'zelena'`

b)  $\pi_{ime}((\pi_{sifraJedrilicar}(\sigma_{boja='crvena'}(jedrilica) \bowtie rezervacija) \cap \pi_{sifraJedrilicar}(\sigma_{boja='zelena'}(jedrilica) \bowtie rezervacija)) \bowtie jedrilicar)$

$\pi_{naziv}(\sigma_{ime='Ivan'}(jedrilicar) \bowtie rezervacija \bowtie jedrilica) \cup \pi_{naziv}(\sigma_{boja='crvena'}(jedrilica))$



## 2. Kontrolna zadaća iz Uvoda u baze podataka

15. siječnja 2003.

1. Među učenicima koji završavaju srednju školu i žele se upisati na fakultet provedena je anketa. U bazi podataka nalaze se relacije (**primarni ključevi relacija su potcrtni**):

### MJESTO

<u>pbr</u>	integer	not null	poštanski broj
nazMjesto	nchar(40)	not null	naziv mjesta

### SKOLA

<u>sifSkola</u>	integer	not null	šifra škole
nazSkola	nchar(60)	not null	naziv škole
pbrSkola	integer	not null	poštanski broj mjesta u kojem se nalazi škola

### FAKULTET

<u>sifFakultet</u>	integer	not null	šifra fakulteta
nazFakultet	nchar(60)	not null	naziv fakulteta
pbrFakultet	integer	not null	poštanski broj mjesta u kojem se nalazi fakultet
adresaFakultet	nchar(60)		adresa fakulteta

### UCENIK

<u>jmbg</u>	char(13)	not null	JMBG učenika
prezUcenik	nchar(25)	not null	prezime učenika
imeUcenik	nchar(25)	not null	ime učenika
sifSkola	integer	not null	šifra srednje škole koju učenik pohađa
sifFakultet	integer	not null	šifra fakulteta na kojeg se učenik želi upisati
opciUspjeh	smallint	not null	opći uspjeh (ocjena 2-5)

Napisati **po jednu** SQL naredbu za svaki sljedeći zadatak:

- a) Za svako mjesto u kojem pohađa srednju školu neki od anketiranih učenika ispisati poštanski broj, naziv mjesta, te ukupan broj različitih fakulteta na koje se žele upisati anketirani učenici iz tog mjesta. Rezultate poredati tako da se mjesta s većim brojem fakulteta na koje se žele upisati anketirani učenici nalaze bliže početku liste. Mjesta u kojima ima jednak broj različitih fakulteta poredati po abecedi. Riješiti bez korištenja podupita. (3 boda)
- b) Ispisati nazive škola za one škole čiji naziv sadrži niz **Gimnazija** ili **gimnazija**, pohađa ih više od 100 anketiranih učenika, a prosječan opći uspjeh učenika koji ju pohađaju je barem za 0.5 veći od prosječnog općeg uspjeha svih anketiranih učenika. Više škola može imati isti naziv. Dozvoljeno je ugraditi samo jedan podupit. Riješiti bez korištenja relacijskog operatora OR. (4 boda)
- c) Napišite naredbu za kreiranje pogleda **neanketiranaSkola** koji treba sadržavati podatke o školama u kojima nije anketiran niti jedan učenik. Shema tog pogleda jednaka je shemi relacije **skola** uz dodatni atribut **brojSkolauMjestu** koji sadrži ukupan broj škola u mjestu u kojem se škola nalazi. (4 boda)
2. Datoteka **konverzija.unl** sadrži sljedeće podatke za svaku školu: stara šifra škole, nova šifra škole (pri tome može postojati škola čija nova šifra odgovara staroj šifri neke druge škole), npr:

```
1234#4321#
1356#1234#
```

Koristi se baza podataka iz 1. zadatka u kojoj su kreirani **samo** primarni ključevi (strani ključevi, niti bilo koji dodatni indeks niti ograničenje nisu kreirani). Nazivi ograničenja kojim se definiraju primarni ključevi dodjeljivani su prema nazivu relacije uz prefix **pk** (npr. **pkUcenik** za relaciju **ucenik**).

Napišite niz SQL naredbi kojima će se kreirati privremena relacija u koju ćete učitati podatke iz **konverzija.unl**, te koristeći te podatke zamijeniti stare šifre škola u nove. Nakon obavljanja promjena, konzistentnost baze mora biti očuvana. (7 bodova)

3. Nad relacijom **fakultet** (iz 1. zadatka) kreiran je primarni ključ i alternativni ključ kojeg čine atributi **nazFakultet**, **pbrFakultet** (prilikom kreiranja alternativnog ključa korišten je ovaj redosljed atributa). Strani ključevi, niti bilo koji dodatni indeks nisu kreirani. Hoće li se prilikom obavljanja sljedećeg upita za pristup do n-torke koristiti indeks? **Objasnite zašto.** (2 boda)

```
SELECT * FROM fakultet
WHERE pbrFakultet = 10000;
```

4. Kreiranjem potrebnih objekata u bazi iz 1. zadatka, osigurati da SUBP implementira sljedeće pravilo prilikom brisanja (DELETE) iz relacije **skola**:
- ako ima manje od 5 anketiranih učenika koji pohađaju tu školu, brisanjem zapisa o školi obrisati i anketirane studente iz te škole.
- Napomena:** zadatak riješiti bez kreiranja pohranjenih procedura! (4 boda)
5. Zadane su relacije **r** i **s** s pripadnim shemama: **R (A, B)**, **S (A, B)**. U relacijama postoje i NULL vrijednosti. Napišite jednu SQL naredbu kojom će se ispisati rezultat za operaciju: **r \ s** (4 boda)

6. Uz pretpostavku da na relaciji  $r$  (A, B, C, D, E, F) vrijedi skup funkcijskih zavisnosti  $F = \{ A \rightarrow BCD, CD \rightarrow EF \}$  dokažite da na relaciji  $r$  vrijedi funkcijska zavisnost  $AE \rightarrow F$ . U svakom koraku dokaza navedite koji ste aksiom ili pravilo za izvođenje funkcijskih zavisnosti upotrijebili (6 bodova)
7. U bazi podataka evidentiraju se podaci o nastavi i prisustvovanju studenata nastavi.

Na početku semestra evidentiraju se nastavne jedinice koje čine plan izvođenja nastave. Nastavna jedinica je identificirana vrstom nastave (laboratorijske vježbe, predavanja, ...), predmetom i rednim brojem nastavnog tjedna. Za nastavnu jedinicu se evidentira još i opis teme koja se obrađuje. Npr:

sifPred	nazPred	oznVrNast	opisVrNast	rbrTjedan	opisTema
101	Uvod u baze podataka	P	predavanje	1	Uvod u baze
101	Uvod u baze podataka	A	auditorne vježbe	1	Uvod u SQL
101	Uvod u baze podataka	P	predavanje	2	Relacijski model pod.
101	Uvod u baze podataka	A	auditorne vježbe	2	SELECT naredba
101	Uvod u baze podataka	L	laboratorijske vježbe	2	Uvod u SQL
105	Operacijska istraživanja	P	predavanje	1	Uvod u op.istraživanja
...					

Pretpostavka je da se u istom tjednu ista vrsta nastave iz predmeta odvija samo jednom.

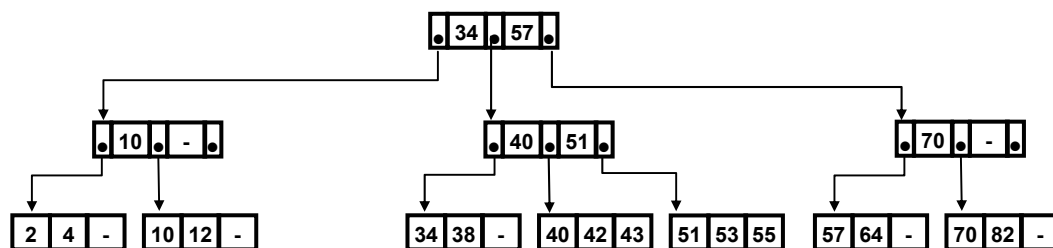
Predmet ima svoju šifru i naziv. Vrsta nastave ima oznaku i opis vrste nastave (npr: L-laboratorijske vježbe, A-auditorne vježbe, P-predavanja).

Evidentira se je li student prisustvovao nastavnoj jedinici. Osim oznake prisustvovanja nastavi (D/N) čuva se i broj bodova koje je pri tom osvojio (npr. u 2. tjednu laboratorijskih vježbi iz Uvoda u baze podataka student s matičnim brojem 0036381073 osvojio je 3 boda). Za studenta se čuva matični broj, ime i prezime.

Za nastavnike se evidentira šifra, ime i prezime i organizacijska jedinica u kojoj su zaposleni. Svaka organizacijska jedinica ima šifru i naziv, ima najviše jednu nadređenu organizacijsku jedinicu, a može imati više podređenih organizacijskih jedinica.

Predmet može imati jednog ili više nastavnika koji su nositelji predmeta. Isti nastavnik može biti nositelj na više predmeta.

- a) Nacrta ER model i opisati entitete i veze. Entitete (osim slabih entiteta) opisati isključivo vlastitim atributima. **Sve sheme moraju zadovoljavati 3NF.** (8 bodova)
- b) Za veze koje postoje među entitetima **nastavnik** i **predmet** napisati uvjete spajanja. (2 boda)
- c) Za segment ER modela koji obuhvaća entitete **nastavnik** i **organizacijska jedinica**, te veze koje postoje među tim entitetima, napisati relacijski model u obliku SQL naredbi za kreiranje relacija s ugrađenim općim pravilima integriteta. Tipove podataka odredite sami. (4 boda)
8. U B-stablo prikazano na slici ubaciti zapis s ključem **56**. **Ne** koristiti algoritam koji je modificiran u svrhu upisa predsortiranih zapisa u stablo. (4 boda)



9. Navedite Armstrongov aksiom o uvećanju i napišite dokaz tog aksioma. (5 bodova)
10. Pod kojim uvjetima se dekompozicija relacijske sheme obavlja bez gubitka informacija? (3 boda)

## 2. Kontrolna zadaća iz Uvoda u baze podataka – rješenja

15. siječnja 2003.

- 1.a) 

```
SELECT mjesto.pbr, nazMjesto, COUNT(DISTINCT sifFakultet)
FROM mjesto, skola, ucenik
WHERE ucenik.sifSkola = skola.sifSkola
AND skola.pbrSkola = mjesto.pbr
GROUP BY 1,2
ORDER BY 3 DESC, 2
```
- b) 

```
SELECT nazSkola
FROM skola, ucenik
WHERE skola.sifSkola = ucenik.sifSkola
AND nazSkola MATCHES '*[Gg]imnazija*'
GROUP BY skola.sifSkola, nazSkola
HAVING AVG(opciUspjeh) > (SELECT AVG(opciUspjeh) FROM ucenik) + 0.5
AND COUNT(*) > 100
```
- c) 

```
CREATE VIEW neanketiranaSkola (sifSkola, nazSkola, pbrSkola, brojSkolaUmjestu) AS
SELECT skola.*, (SELECT COUNT(*) FROM skola skolal
WHERE skola.pbrSkola = skolal.pbrSkola)
FROM skola
WHERE sifSkola NOT IN (SELECT DISTINCT sifSkola FROM ucenik)
```
2. 

```
CREATE TEMP TABLE konverzija (
staraSifSkola INTEGER
, novaSifSkola INTEGER);

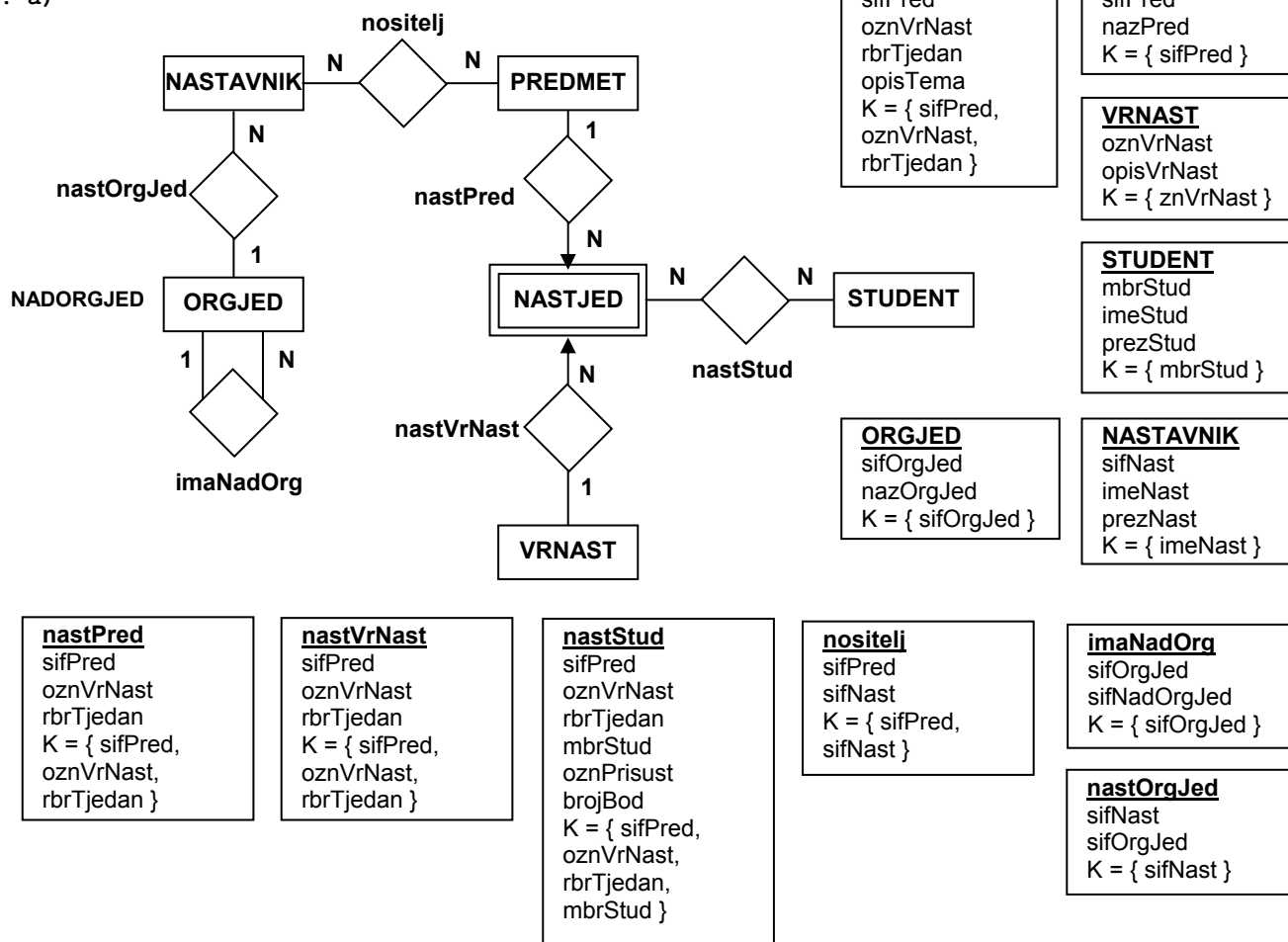
LOAD FROM 'konverzija.unl' DELIMITER '#' INSERT INTO konverzija;
ALTER TABLE skola DROP CONSTRAINT pkSkola;
UPDATE skola SET sifSkola = (SELECT novaSifSkola FROM konverzija
WHERE konverzija.staraSifSkola = skola.sifSkola);
ALTER TABLE skola ADD CONSTRAINT PRIMARY KEY (sifSkola) CONSTRAINT pkSkola;
UPDATE ucenik SET sifSkola = (SELECT novaSifSkola FROM konverzija
WHERE konverzija.staraSifSkola = ucenik.sifSkola);
```
3. Indeks se neće koristiti. Ako je nad relacijom R kreiran kompozitni indeks za npr. attribute A, B, C, do n-torke se može pristupiti korištenjem bilo koje od navedenih kombinacija atributa kao ključa za dohvat - A, AB, ABC. Dakle, da bi se prilikom dohvata zapisa mogao koristiti indeks koji nastaje kreiranjem alternativnog ključa nazFakultet, pbrFakultet uvjet dohvata je potrebno postaviti i na atribut nazFakultet.
4. 

```
CREATE TRIGGER delSkola
DELETE ON skola
REFERENCING OLD AS oldSkola
FOR EACH ROW
WHEN ( (SELECT COUNT(*) FROM ucenik
WHERE ucenik.sifSkola = oldSkola.sifSkola) < 5)
(DELETE FROM ucenik WHERE sifSkola = oldSkola.sifSkola);
```
5. 

```
SELECT r.A, r.B
FROM r
WHERE NOT EXISTS (SELECT * FROM s
WHERE (r.a = s.a OR r.a IS NULL AND s.a IS NULL)
AND (r.b = s.b OR r.b IS NULL AND s.b IS NULL))
```
6. 

$A \rightarrow BCD \Rightarrow A \rightarrow CD$	- pravilo dekompozicije
$A \rightarrow CD, CD \rightarrow EF \Rightarrow A \rightarrow EF$	- aksiom o tranzitivnosti
$A \rightarrow EF \Rightarrow A \rightarrow F$	- pravilo dekompozicije (pravilo o projektivnosti)
$A \rightarrow F \Rightarrow AE \rightarrow F$	- aksiom o uvećanju

7. a)



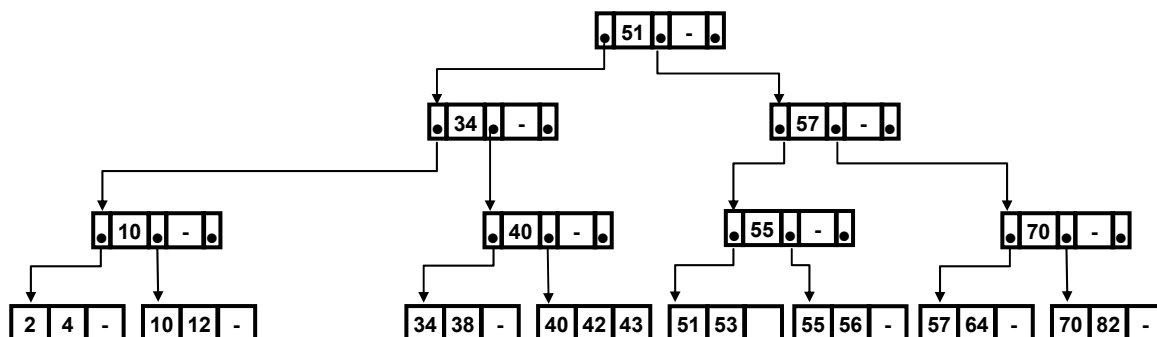
b) NASTAVNIK.sifNast = nositelj.sifNast AND nositelj.sifPred = PREDMET.sifPred

c) 

```
CREATE TABLE orgJed (sifOrgJed    INTEGER
, nazOrgJed    CHAR(20)
, sifNadOrgJed INTEGER
, PRIMARY KEY (sifOrgJed)
, FOREIGN KEY (sifNadOrgJed) REFERENCES orgJed (sifOrgJed));

CREATE TABLE nastavnik (sifNast    INTEGER
, imeNast    CHAR(20)
, prezNast    CHAR(20)
, sifOrgJed    INTEGER
, PRIMARY KEY (sifNast)
, FOREIGN KEY (sifOrgJed) REFERENCES orgJed (sifOrgJed));
```

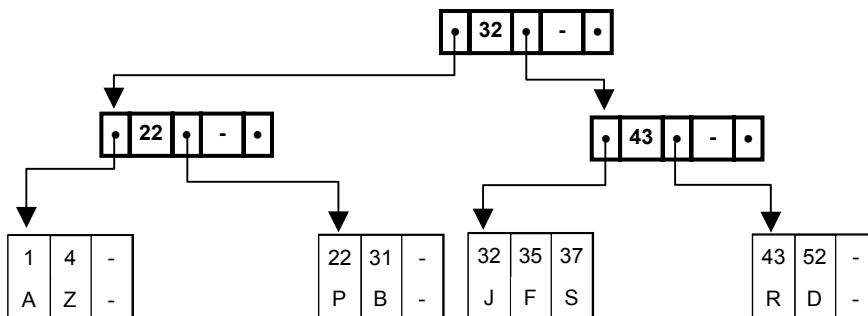
8.



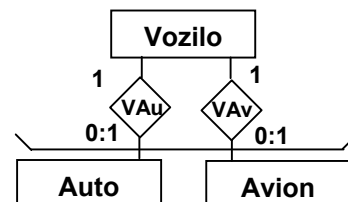
9. vidjeti predavanja

10. vidjeti predavanja

1. Pomoću B-stabla na slici moguće je na temelju zadanog poštanskog broja efikasno dohvatiti podatke o gradu. Podaci su sortirani prema poštanskom broju. Osim poštanskog broja, za svaki je grad evidentiran i naziv grada (npr. naziv grada s poštanskim brojem 35 je F).



- a) iz stabla sa slike obrisati zapis s poštanskim brojem 31. Dovoljno je nacrtati samo konačno rješenje. (4 boda)
- b) nacrtajte **ново** B-stablo koje će imati **isti broj zapisa** u indeksnim i podatkovnim blokovima i sadržavati **iste podatke** kao stablo dobiveno u a) dijelu zadatka, ali tako da su **podaci sortirani prema nazivu grada u rastućem poretku**. To **ново** B-stablo mora imati strukturu kojom će biti osigurano sljedeće:
- i dalje je moguće efikasno dohvaćati podatke prema zadanom **poštanskom broju**
  - omogućeno je efikasno obavljanje upita oblika
- `SELECT * FROM grad WHERE pbr BETWEEN 4 AND 35` (Obratite pozornost na BETWEEN). (4 boda)
2. Za vlasnika trkaćeg konja evidentira se jmbg, prezime i ime. Za konja treba evidentirati ime konja, datum kad se oždrijebio i kojem vlasniku pripada. Pretpostavlja se da jedan vlasnik može imati više konja, ali svi **njegovi** konji imaju različita imena.
- Svaki džokej (džokej je jahač trkaćeg konja) ima šifru koja ga identificira, ime i prezime. Nije potrebno evidentirati je li džokej vlasnik konja ili je li neki vlasnik konja ujedno i džokej. Evidentirati sljedeće: džokej može (ali ne mora) imati svog preferencijalnog konja (najviše jednog); konj može (ali ne mora) imati svog preferencijalnog džokeja (najviše jednog). Preferencijalni znači: onaj kojeg se najradije (ali ne isključivo) odabire.
- Za utrke koje se održavaju na zagrebačkom hipodromu evidentira se datum, trenutak početka utrke (u obliku hh:mm, tj. sat i minuta), te stanje staze tijekom utrke (staza može biti ili "mokra" ili "suha"). U istom trenutku može početi samo jedna utrka. Treba evidentirati koji je konj sudjelovao u kojoj utrci i koji je džokej pri tome na njemu jahao. Ne postoji ograničenje prema kojem džokej može jahati samo na svojem preferencijalnom konju ili prema kojem će na konju jahati samo njegov preferencijalni džokej.
- a) nacrtati ER model, opisati entitete i veze (njihove atribute i ključeve). Sve sheme moraju zadovoljavati 3NF. **Ne treba** preslikavati model u relacijski model niti opisivati tipove podataka i uvjete spajanja u vezama.
- b) navesti nazive (samo nazive) svih relacija koje se dobiju **na kraju** postupka transformacije vašeg ER modela u relacijski model (12 bodova)
3. Zadan je ER model na slici. Primarni ključ za **Vozilo** je tvornički broj. Primarni ključ za **Auto** je registarski broj. Primarni ključ za **Avion** je registarski broj. Za vozilo treba evidentirati godinu proizvodnje, za auto maksimalnu brzinu, a za avion duljinu aviona i raspon krila.



Napisati niz SQL naredbi kojima će se kreirati svi potrebni objekti za realizaciju prikazanog ER modela u relacijskoj (**NE** objektno-relacijskoj) bazi podataka. Pri tome:

- tipove podataka odaberite prema želji
- moraju biti zadovoljena sva ograničenja integriteta koja proizlaze iz modela
- registarski broj auta mora sadržavati minimalno 7 znakova, a aviona minimalno 5 znakova
- raspon krila ne smije biti veći od duljine aviona
- pokušaj **unos**a podataka u relaciju **auto** za vozilo koje je već opisano kao avion, spriječiti dojavom poruke: "ovo vozilo je već opisano kao avion"

(10 bodova)

4. U bazi podataka LETOVI prate se podaci o letovima aviona. Podvučeni atributi čine primarni ključ relacije. Vrijednost niti jednog atributa ne može poprimiti NULL vrijednost.

**AVION** - opisuje pojedinačne avione

<b>sifAvion</b>	integer	šifra aviona
tipAvion	char(15)	oznaka tipa aviona
maxDolet	smallint	najveća udaljenost ( <b>u kilometrima</b> ) koju avion može preletjeti bez punjenja goriva

**AERODROM** - opisuje aerodrome

<b>sifAer</b>	integer	šifra aerodroma
nazAer	char(20)	naziv aerodroma
oznDrzave	char(2)	oznaka države u kojoj se nalazi aerodrom

**LET** - koji avion je letio kamo i kada (ovo su **ostvareni** letovi, a ne planirani letovi)

<b>sifLet</b>	integer	šifra leta
datum	date	datum leta
vrijemePol	char(5)	vrijeme polijetanja u obliku SS:MM (sati i minute)
sifAvion	integer	šifra aviona
sifAerPol	integer	šifra aerodroma polijetanja
sifAerSlij	integer	šifra aerodroma slijetanja

**UDALJENOST** - udaljenost upisana za svaki par aerodroma: npr. PLESO-DUBROVNIK i DUBROVNIK-PLESO

<b>sifAerOd</b>	integer	šifra aerodroma
<b>sifAerDo</b>	integer	šifra aerodroma
udalj	smallint	udaljenost ( <b>u kilometrima</b> ) između aerodroma sa šifrom sifAerOd do aerodroma sa šifrom sifAerDo

- a) Napisati SQL naredbu za kreiranje relacije **noviAerodrom**. Relacija ima 3 atributa: šifru aerodroma, naziv aerodroma, popis aviona koji su **tijekom posljednjih 45 dana** sletjeli na taj aerodrom (pod popisom aviona misli se na popis **šifara** aviona). Primarni ključ relacije je šifra aerodroma. Avion koji je na neki aerodrom sletio više puta, naći će se u popisu aviona za taj aerodrom više puta. Redoslijed aviona u popisu nije bitan.

Nakon naredbe za kreiranje relacije, napisati naredbu za punjenje relacije **noviAerodrom** i to tako da se po jedna n-torka upiše za **svaki** aerodrom iz relacije **aerodrom**. Pri tome se popis aviona svakog aerodroma formira na temelju podataka koji se nalaze u relaciji **let**. **(6 bodova)**

Napisati **po jedan** SQL upit za svaki od sljedećih zadataka (**ne koristiti** relaciju **noviAerodrom**):

- b) Udaljenost između svakog para aerodroma u Republici Hrvatskoj povećati za 5%. Oznaka države za Republiku Hrvatsku je HR.

**(5 bodova)**

- c) Ispisati listu za održavanje aviona. Za svaki avion treba ispisati šifru aviona, tip aviona, uputu za održavanje. Za avione koji su imali **više od 500** polijetanja, uputa za održavanje glasi: **tvornički servis**; za avione koji su imali od **100 do 500** polijetanja uputa za održavanje glasi: **redovni servis**; za avione koji su imali od **0 do 99** polijetanja (primijetite da je **0** uključena) uputa za održavanje glasi: **redovni pregled** (naravno, smatra se da jedan let aviona znači jedno polijetanje). Nije dopuštena upotreba podupita niti UNION operatora.

**(5 bodova)**

5. Zadane su relacije s pripadnim shemama:  $r(A, B)$ ,  $s(A, B)$  i  $t(A, B)$ . Zadatak riješite upotrebom jedne SQL naredbe (u relacijama nema NULL vrijednosti)

$r \cap s \cap t$

**(4 boda)**

6. Zadana je relacijska shema  $R(A, B, C, D, E, F, G)$ . Pretpostavljeni ključ je atribut A. Postoji skup funkcijskih zavisnosti  $F = \{ ABC \rightarrow DEFG, AB \rightarrow DFG, B \rightarrow EF, E \rightarrow BF \}$

Normalizirajte relaciju na

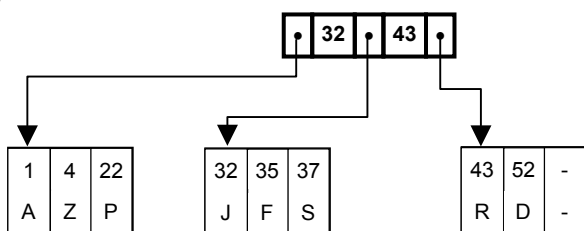
- 1 NF (1 NF ostvariti BEZ rastavljanja sheme relacije R, proširenjem ključa)
- 2 NF
- 3 NF

**(6 bodova)**

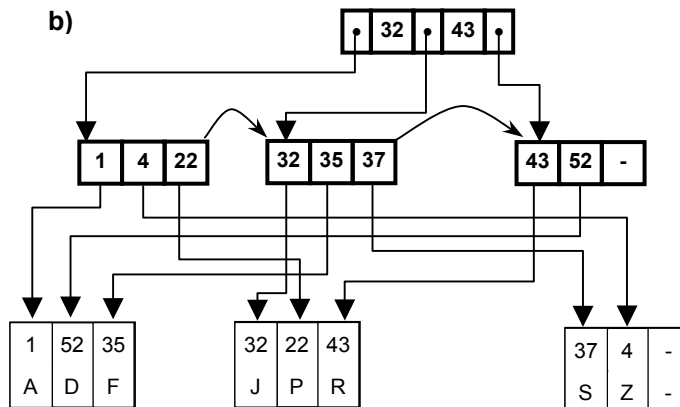
7. Nabrojite zadatke sustava za upravljanje bazama podataka. **Ne treba ih opisivati.**

**(4 boda)**

1.a)

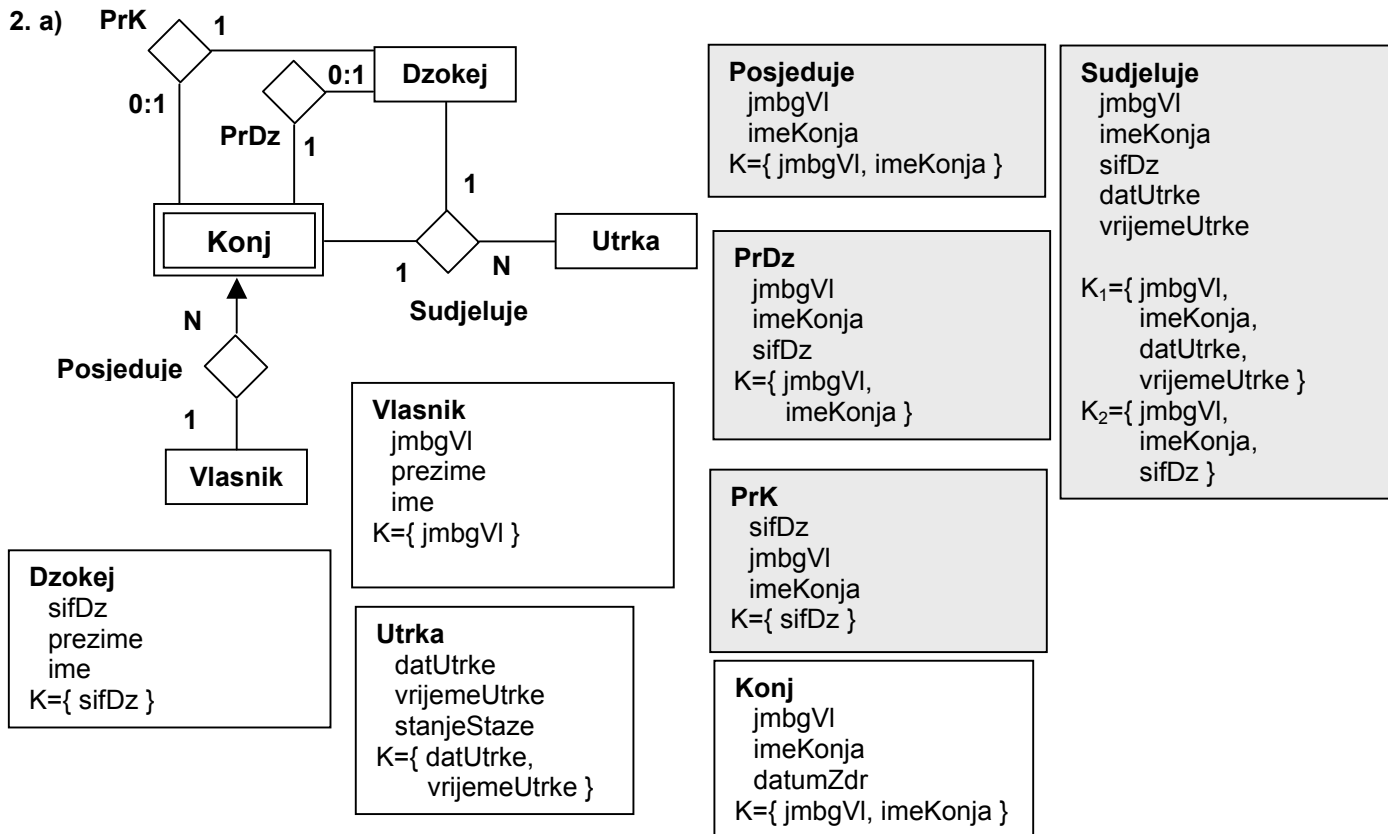


b)



**komentar uz rješenje:** u listovima B-stabla ne nalaze se podaci, nego kazaljke na podatke čime je omogućeno da podaci budu nesortirani; listovi B-stabla povezani su kazaljka čime je omogućen dohvat iz zadanog intervala vrijednosti;

2. a) PrK



b) Svi entiteti i veza Sudjeluje preslikavaju se u relacije: Vlasnik, Konj, Dzozej, Utrka, Sudjeluje

```

3. CREATE TABLE vozilo (
    tvBroj      CHAR(40)
    , godProizv  SMALLINT
    , PRIMARY KEY (tvBroj) CONSTRAINT pkVozilo);

CREATE TABLE auto (
    regBroj     CHAR(20) CHECK (LENGTH(regBroj) >= 7) CONSTRAINT cAutoRegBroj
    , maxBrzina  INTEGER
    , tvBroj     CHAR(40) NOT NULL
    , PRIMARY KEY (regBroj) CONSTRAINT pkAuto
    , UNIQUE (tvBroj) CONSTRAINT uAutoTvBroj
    , FOREIGN KEY (tvBroj) REFERENCES vozilo (tvBroj) CONSTRAINT fkAutoVozilo
);

CREATE TABLE avion (
    regBroj     CHAR(20) CHECK (LENGTH(regBroj) >= 5) CONSTRAINT cAvionRegBroj
    , duljina    DECIMAL(6,3)
    , raspKrila  DECIMAL(6,3)
    , tvBroj     CHAR(40) NOT NULL
    , PRIMARY KEY (regBroj) CONSTRAINT pkAvion
    , UNIQUE (tvBroj) CONSTRAINT uAvionTvBroj
    , FOREIGN KEY (tvBroj) REFERENCES vozilo (tvBroj) CONSTRAINT fkAvionVozilo
    , CHECK (duljina >= raspKrila) CONSTRAINT cAvionDuljina
);

CREATE PROCEDURE pogreska()
    RAISE EXCEPTION -746, 0, 'Ovo vozilo je već opisano kao avion';
END PROCEDURE;

CREATE TRIGGER autoInsert
    INSERT ON auto
    REFERENCING NEW AS noviAuto
    FOR EACH ROW
        WHEN (EXISTS (SELECT * FROM avion WHERE avion.tvBroj = noviAuto.tvBroj))
            (EXECUTE PROCEDURE pogreska());

4. a) CREATE TABLE noviAerodrom (
    sifAer      INTEGER
    , nazAer     CHAR(20)
    , avioni     MULTISSET(INTEGER NOT NULL)
    , PRIMARY KEY (sifAer) CONSTRAINT pkNoviAerodrom
);
INSERT INTO noviAerodrom
    SELECT sifAer
        , nazAer
        , MULTISSET(SELECT ITEM sifAvion FROM let
                    WHERE sifAerSlij = aerodrom.sifAer
                    AND datum > TODAY - 45)
    FROM aerodrom;

b) UPDATE udaljenost SET udalj = 1.05*udalj
    WHERE udaljenost.sifAerOd IN
        (SELECT sifAer FROM aerodrom
         WHERE oznDrzave = 'HR')
    AND udaljenost.sifAerDo IN
        (SELECT sifAer FROM aerodrom
         WHERE oznDrzave = 'HR')

c) SELECT avion.sifAvion
    , tipAvion
    , CASE WHEN COUNT(let.sifLet) > 500 THEN 'tvornički servis'
          WHEN COUNT(let.sifLet) BETWEEN 100 AND 500 THEN 'redovni servis'
          WHEN COUNT(let.sifLet) < 100 THEN 'redovni pregled'
    END
    FROM avion LEFT OUTER JOIN let
    ON avion.sifAvion = let.sifAvion
    GROUP BY 1, 2

```



5. 

```
SELECT r.* FROM r
WHERE EXISTS
  (SELECT * FROM s
   WHERE s.A = r.A
     AND s.B = r.B
     AND EXISTS
      (SELECT * FROM t
       WHERE t.A = s.A
         AND t.B = s.B)
  )
```

iii

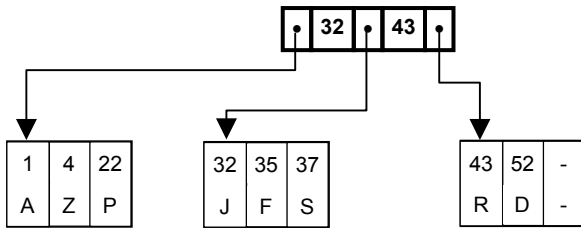
```
SELECT r.* FROM r, s, t
WHERE r.A = s.A
  AND r.B = s.B
  AND s.A = t.A
  AND s.B = t.B
```

6.	1NF:	R = ABCDEFG	K = ABC	
	2NF:	R <sub>1</sub> = ABDG R <sub>2</sub> = BEF R <sub>3</sub> = ABC	K <sub>1</sub> = AB K <sub>2</sub> = B K <sub>3</sub> = ABC	
	3NF:	R <sub>1</sub> = ABDG R <sub>2</sub> = BEF R <sub>3</sub> = ABC	K <sub>1</sub> = AB K <sub>21</sub> = B K <sub>3</sub> = ABC	K <sub>22</sub> = E

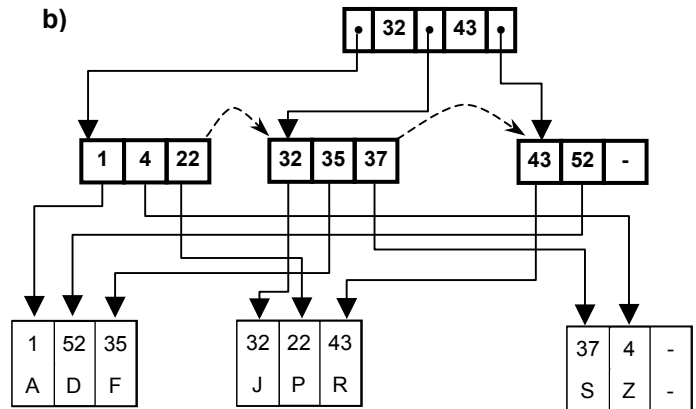
- 7.
- sakriva od korisnika detalje fizičke pohrane podataka
  - omogućuje definiciju i rukovanje s podacima
    - DDL - Data Definition Language
    - DML - Data Manipulation Language
  - obavlja funkciju zaštite podataka
    - integritet podataka
    - pristup podacima - autorizacija, sigurnost
    - kontrola paralelnog pristupa
    - obnova u slučaju razrušenja
  - optimiranje upita

- Rješenja -

1.a)

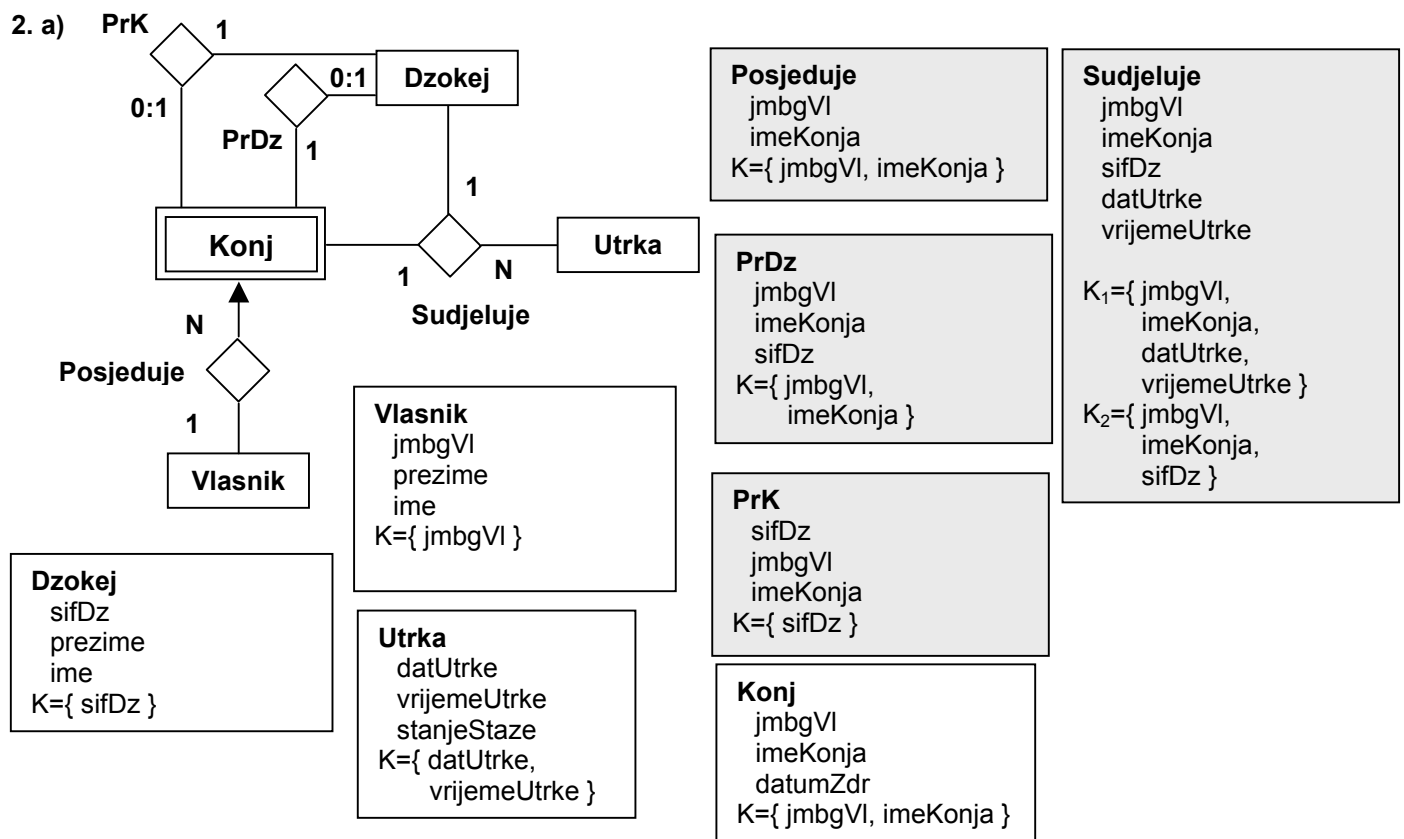


b)



**komentar uz rješenje:** u listovima B-stabla ne nalaze se podaci, nego kazaljke na podatke čime je omogućeno da podaci budu nesortirani; listovi B-stabla povezani su kazaljka čime je omogućen dohvat iz zadanog intervala vrijednosti;

2. a)



b) Svi entiteti i veza Sudjeluje preslikavaju se u relacije: Vlasnik, Konj, Dzozej, Utrka, Sudjeluje

```

3. CREATE TABLE vozilo (
    tvBroj      CHAR(40)
  , godProizv   SMALLINT
  , PRIMARY KEY (tvBroj) CONSTRAINT pkVozilo);

CREATE TABLE auto (
    regBroj     CHAR(20) CHECK (LENGTH(regBroj) >= 7) CONSTRAINT cAutoRegBroj
  , maxBrzina   INTEGER
  , tvBroj      CHAR(40) NOT NULL
  , PRIMARY KEY (regBroj) CONSTRAINT pkAuto
  , UNIQUE (tvBroj) CONSTRAINT uAutoTvBroj
  , FOREIGN KEY (tvBroj) REFERENCES vozilo (tvBroj) CONSTRAINT fkAutoVozilo
);

CREATE TABLE avion (
    regBroj     CHAR(20) CHECK (LENGTH(regBroj) >= 5) CONSTRAINT cAvionRegBroj
  , duljina     DECIMAL(6,3)
  , raspKrila   DECIMAL(6,3)
  , tvBroj      CHAR(40) NOT NULL
  , PRIMARY KEY (regBroj) CONSTRAINT pkAvion
  , UNIQUE (tvBroj) CONSTRAINT uAvionTvBroj
  , FOREIGN KEY (tvBroj) REFERENCES vozilo (tvBroj) CONSTRAINT fkAvionVozilo
  , CHECK (duljina >= raspKrila) CONSTRAINT cAvionDuljina
);

CREATE PROCEDURE pogreska()
    RAISE EXCEPTION -746, 0, 'Ovo vozilo je već opisano kao avion';
END PROCEDURE;

CREATE TRIGGER autoInsert
    INSERT ON auto
    REFERENCING NEW AS noviAuto
    FOR EACH ROW
        WHEN (EXISTS (SELECT * FROM avion WHERE avion.tvBroj = noviAuto.tvBroj))
            (EXECUTE PROCEDURE pogreska());

4. a) CREATE TABLE noviAerodrom (
    sifAer      INTEGER
  , nazAer      CHAR(20)
  , avioni      MULTISSET(INTEGER NOT NULL)
  , PRIMARY KEY (sifAer) CONSTRAINT pkNoviAerodrom
);
INSERT INTO noviAerodrom
    SELECT sifAer
      , nazAer
      , MULTISSET(SELECT ITEM sifAvion FROM let
                  WHERE sifAerSlij = aerodrom.sifAer
                  AND datum > TODAY - 45)
    FROM aerodrom;

b) UPDATE udaljenost SET udalj = 1.05*udalj
    WHERE udaljenost.sifAerOd IN
        (SELECT sifAer FROM aerodrom
         WHERE oznDrzave = 'HR')
    AND udaljenost.sifAerDo IN
        (SELECT sifAer FROM aerodrom
         WHERE oznDrzave = 'HR')

c) SELECT avion.sifAvion
      , tipAvion
      , CASE WHEN COUNT(let.sifLet) > 500 THEN 'tvornički servis'
            WHEN COUNT(let.sifLet) BETWEEN 100 AND 500 THEN 'redovni servis'
            WHEN COUNT(let.sifLet) < 100 THEN 'redovni pregled'
            END
    FROM avion LEFT OUTER JOIN let
    ON avion.sifAvion = let.sifAvion
    GROUP BY 1, 2

```

5. 

```
SELECT r.* FROM r
WHERE EXISTS
  (SELECT * FROM s
   WHERE s.A = r.A
     AND s.B = r.B
     AND EXISTS
      (SELECT * FROM t
       WHERE t.A = s.A
         AND t.B = s.B)
  )
```

iii

```
SELECT r.* FROM r, s, t
WHERE r.A = s.A
  AND r.B = s.B
  AND s.A = t.A
  AND s.B = t.B
```

6.	1NF:	R = ABCDEFG	K = ABC	
	2NF:	R <sub>1</sub> = ABDG R <sub>2</sub> = BEF R <sub>3</sub> = ABC	K <sub>1</sub> = AB K <sub>2</sub> = B K <sub>3</sub> = ABC	
	3NF:	R <sub>1</sub> = ABDG R <sub>2</sub> = BEF R <sub>3</sub> = ABC	K <sub>1</sub> = AB K <sub>21</sub> = B K <sub>3</sub> = ABC	K <sub>22</sub> = E

- 7.
- sakriva od korisnika detalje fizičke pohrane podataka
  - omogućuje definiciju i rukovanje s podacima
    - DDL - Data Definition Language
    - DML - Data Manipulation Language
  - obavlja funkciju zaštite podataka
    - integritet podataka
    - pristup podacima - autorizacija, sigurnost
    - kontrola paralelnog pristupa
    - obnova u slučaju razrušenja
  - optimiranje upita

## UVOD U BAZE PODATAKA – 2. međuispit 19.01.2005.

1. Prilikom rješavanja ovog zadatka koristiti sljedeće relacije i atribute iz rječnika podataka (podvučeni atributi čine primarni ključ relacije):

Relacija	Atributi
systables	<u>tabid</u> , tabname, tabtype
syscolumns	<u>tabid</u> , <u>colno</u> , colname, coltype
sysviews	<u>tabid</u> , <u>segno</u> , viewtext
systriggers	<u>trigid</u> , trigname, tabid, event
sysconstraints	<u>constrid</u> , tabid, constrname, constrtype, idxname

Napisati po **jednu SQL naredbu** za svaki od zadataka od a) do e). **Neće se priznavati rješenja u kojima se nepotrebno koriste podupiti.**

- a) Kreirati i napuniti privremenu tablicu *numTrigg* koja će za svaku baznu relaciju sadržavati **tabid**, **tabname** i ukupan broj okidača definiranih nad njom. Za bazne relacije nad kojima nije definiran niti jedan okidač, za broj okidača u tablici *numTrigg* treba biti upisana vrijednost 0. (4 boda)
- b) Iz privremene tablice *numTrigg* definirane u a) dijelu zadatka obrisati podatke o svim relacijama nad kojima nije definirano niti jedno ograničenje. (3 boda)
- c) Za sve relacije koje nisu dio rječnika podataka ispisati **tabid**, **tabname**, ukupan broj okidača definiranih nad tom relacijom i ukupan broj ograničenja definiranih nad tom relacijom. Relacije trebaju biti silazno poredane po broju okidača, a relacije s jednakim brojem okidača trebaju biti uzlazno poredane po broju ograničenja. (3 boda)
- d) Kreirati pogled *vAttrib* koji će za svaki tip atributa (**coltype**) prikazati broj baznih relacija u kojima se taj tip atributa pojavljuje. Pogled treba sadržavati samo one tipove atributa koji se pojavljuju u barem pet različitih relacija. (4 boda)
- e) Ispisati sve podatke o relacijama koje **ne sadrže** najmanji broj atributa. (4 boda)

2. U bazi podataka pohranjuju se podaci o održavanju predstava u nekom kazalištu. Predstava ima svoju šifru (koja ju jednoznačno određuje) i naziv predstave. Svaka predstava pripada nekoj od kategorija (drama, balet, opera, ...). Kategorija ima svoju oznaku i opis (D – drama, B – balet, ...).

Jedno održavanje predstave naziva se izvedbom. Svaka predstava može imati više izvedbi. Za izvedbu se evidentira datum i točno vrijeme početka (sat i minuta) te broj prodanih karata. Jedna predstava ne može imati više izvedbi koje počinju istog dana u isto vrijeme.

Izvedbe se ocjenjuju od strane kritičara. Ocjena može poprimiti vrijednost od 1 do 10. Jednu izvedbu može ocijeniti više kritičara, a jedan kritičar može ocijeniti više izvedbi. Za kritičara se evidentira šifra (koja ga jednoznačno određuje), ime i prezime.

Za svaku predstavu evidentirani su zadaci zaposlenika kazališta (glumac, režiser, scenograf ...). Svaki zaposlenik može na jednoj predstavi obavljati samo jedan zadatak. Isti zadatak na jednoj predstavi može obavljati nekoliko zaposlenika (predstava može imati više glumaca, scenografa ...). Za svaki zadatak evidentira se šifra (koja ga jednoznačno određuje) i naziv. Za svakog zaposlenika evidentira se JMBG, ime i prezime.

- a) Nacrtati ER model i opisati entitete i veze. Entitete (osim slabih entiteta) opisati isključivo vlastitim atributima. **Sve sheme moraju zadovoljavati 3NF.** (10 bodova)
- b) Dobiveni ER model transformirati u relacijski model. Za svaku relaciju navesti naziv relacije, atribute i ključeve. Odabrati relaciju koja ima najveći broj stranih ključeva i napisati SQL naredbu za kreiranje te relacije s ugrađenim **općim** pravilima integriteta. (5 bodova)

3. Na temelju čega se određuje ispravnost (*validation*) XML dokumenta? (4 boda)

4. a) Na relaciji  $r(A, B, C, D, E)$  vrijede funkcijske zavisnosti iz skupa  $F = \{A \rightarrow E, A \rightarrow BC, B \rightarrow DE\}$ . Korištenjem Armstrongovih aksioma i pravila koja iz njih proizlaze dokažite da na relaciji  $r$  vrijedi funkcijska zavisnost  $AE \rightarrow D$ .  
Napomena: definicije pravila i aksioma nije potrebno pisati, dovoljno je navesti samo naziv. (3 boda)

b) Zadana je relacijska shema  $R(A, B, C, D, E, F, G, H)$ . Relacijska shema  $R$  ima ključ ABCD i u prvoj je normalnoj formi. Uz pretpostavku da na  $R$  vrijedi skup funkcijskih zavisnosti  $F = \{AB \rightarrow EF, CD \rightarrow G, ABCD \rightarrow EFGH, B \rightarrow F\}$ , normalizirati  $R$  na drugu normalnu formu. (5 bodova)

5. Zadane su relacije s pripadnim shemama:  $r(A,B)$ ,  $s(A,C)$ . U relacijama nema NULL vrijednosti. (4 boda)

Izvršavanje SQL naredbi:	vraća sljedeći broj n-torki:
SELECT COUNT(A) FROM r	250
SELECT COUNT(DISTINCT A) FROM r	250
SELECT COUNT(A) FROM s	100
SELECT COUNT(DISTINCT A) FROM s	100
SELECT COUNT(*) FROM r,s WHERE r.A = s.A	50

Koliko n-torki će vratiti sljedeći upiti:

a) 

```
select A from r
      UNION ALL
select A from r
```

b) 

```
select A from r
      UNION
select A from s
```

6. Za relaciju **pred** (sifPred, kratPred, nazPred, sifOrgJed, upisanoStud, brojSatiTjedno) napišite SQL naredbe za kreiranje **najmanjeg mogućeg** broja indeksa koji će omogućiti efikasno obavljanje (pri tome se misli na mogućnost obavljanja upita pomoću B-stabla) svih sljedećih upita:

```
SELECT * FROM pred ORDER BY nazPred, sifPred, upisanoStud
SELECT * FROM pred WHERE nazPred = 'Baze podataka' AND upisanoStud = 200
SELECT * FROM pred WHERE sifPred = 121 AND nazPred = 'Programiranje'
SELECT * FROM pred ORDER BY nazPred, sifPred, upisanoStud DESC
SELECT * FROM pred WHERE brojSatiTjedno BETWEEN 4 AND 8
SELECT * FROM pred ORDER BY brojSatiTjedno DESC, sifPred DESC
SELECT * FROM pred ORDER BY nazPred DESC, sifPred DESC, upisanoStud ASC
```

(4 boda)

7. a) Definirati DISTINCT korisničke tipove podataka *tKglZnos* i *tFlZnos* koji služe za evidenciju mase u kilogramima i funtama (1 funta = 0.45 kg), a temelje se na ugrađenom tipu DECIMAL(5,2). (1 bod)

b) Napisati CAST operator za pretvaranje *tKglZnos* u *tFlZnos*. (3 boda)

c) Kreirati relaciju *ArtiklKg* za pohranu podataka o artiklima. Relacija treba sadržavati oznaku i naziv artikla te masu u kilogramima. Koristiti korisnički tip podataka definiran u a) dijelu zadatka. (1 bod)

d) Napisati SELECT naredbu kojom će se za svaki artikl ispisati njegova oznaka, masa u kilogramima i masa u funtama. (2 boda)

**UVOD U BAZE PODATAKA – 2. međuispit - rješenja**  
**19.01.2005.**

1.

- a) 

```
SELECT systables.tabid, tabname, COUNT(trigid) brojOkidaca
FROM systables
LEFT OUTER JOIN systriggers
ON systables.tabid = systriggers.tabid
WHERE systables.tabtype = 'T'
GROUP BY 1, 2
INTO TEMP numTrigg
```
- b) 

```
DELETE FROM numTrigg
WHERE NOT EXISTS
(SELECT * FROM sysconstraints
WHERE sysconstraints.tabid = numTrigg.tabid)
```

  
ili  

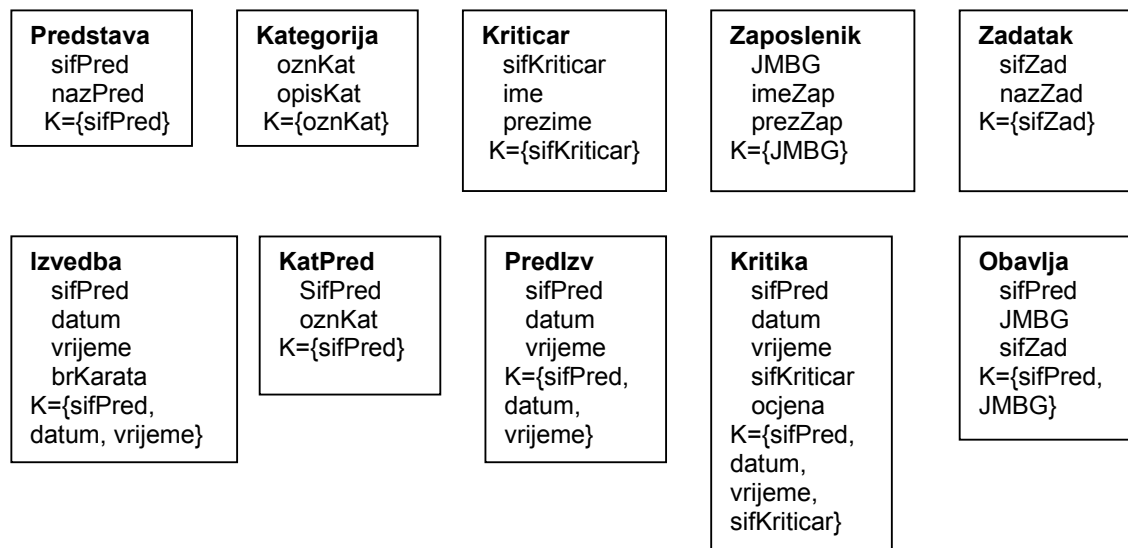
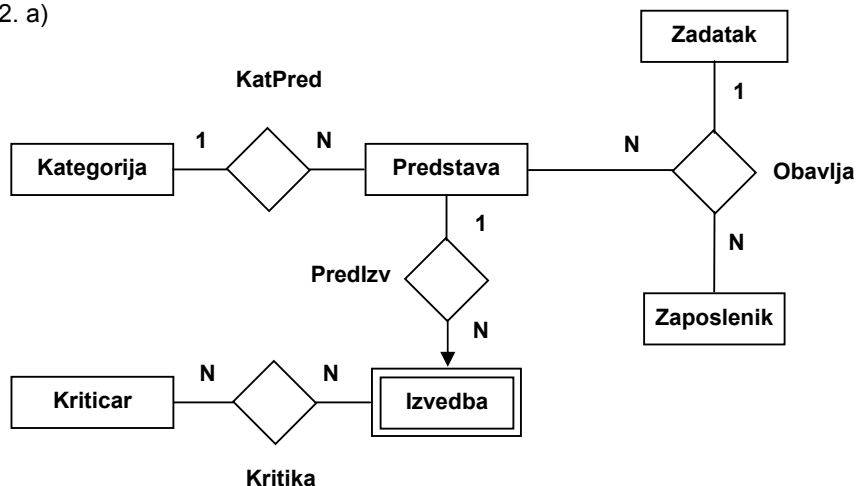
```
DELETE FROM numTrigg
WHERE tabid NOT IN
(SELECT tabid FROM sysconstraints)
```
- c) 

```
SELECT tabid, tabname,
(SELECT COUNT(*) FROM systriggers
WHERE systriggers.tabid = systables.tabid),
(SELECT COUNT(*) FROM sysconstraints
WHERE sysconstraints.tabid = systables.tabid)
FROM systables
WHERE tabid >= 100
ORDER BY 3 DESC, 4
```
- d) 

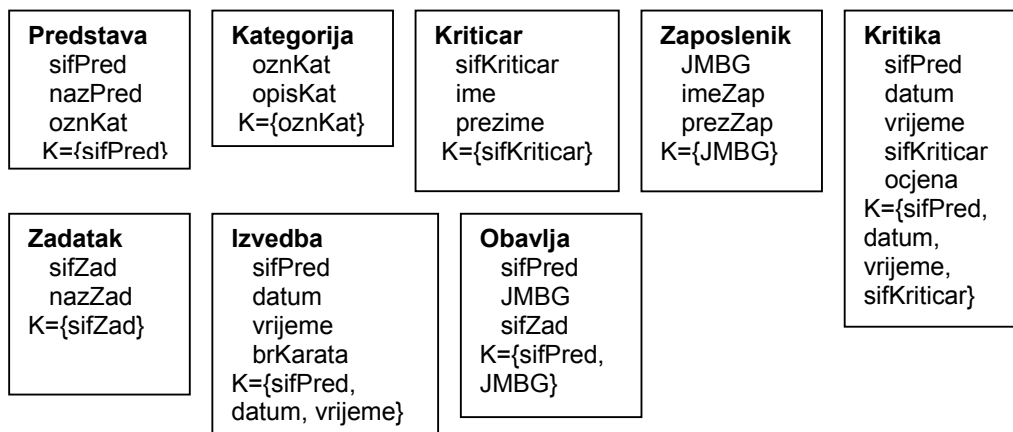
```
CREATE VIEW vAttrib (coltype, brojPojavljivanja) AS
SELECT coltype, COUNT(DISTINCT syscolumns.tabid)
FROM syscolumns, systables
WHERE syscolumns.tabid = systables.tabid
AND systables.tabtype = 'T'
GROUP BY 1
HAVING COUNT(DISTINCT tabid) >= 5
```
- e) 

```
SELECT systables.*
FROM systables
WHERE (SELECT COUNT(*) FROM syscolumns
WHERE systables.tabid = syscolumns.tabid) >
ANY (SELECT COUNT(*) FROM syscolumns
GROUP BY syscolumns.tabid)
```

2. a)



b)





```
CREATE TABLE obavlja (
    sifPred      INTEGER
  , JMBG        CHAR(13)
  , sifZadatak  INTEGER
  , PRIMARY KEY (sifPred, JMBG)
  , FOREIGN KEY (sifPred) REFERENCES predstava(sifPred)
  , FOREIGN KEY (JMBG) REFERENCES zaposlenik(JMBG)
  , FOREIGN KEY (sifZadatak) REFERENCES zadatak(sifZadatak)
)
```

3. Vidjeti predavanja.

4. a)  $B \rightarrow DE \Rightarrow B \rightarrow D$  - pravilo dekompozicije  
 $A \rightarrow BC \Rightarrow A \rightarrow B$  - pravilo dekompozicije  
 $A \rightarrow B \wedge B \rightarrow D \Rightarrow A \rightarrow D$  - aksiom tranzitivnosti  
 $A \rightarrow D \Rightarrow AE \rightarrow D$  - aksiom uvećanja  
 b)  $R_1(A, B, E), K_1 = AB$   
 $R_2(B, F), K_2 = B$   
 $R_3(C, D, G), K_3 = CD$   
 $R_4(A, B, C, D, H), K_4 = ABCD$
5. a) 500  
 b) 300
6. 

```
CREATE INDEX predIdx1 ON pred(nazPred, sifPred, upisanoStud);
CREATE INDEX predIdx2 ON pred(nazPred, upisanoStud);
CREATE INDEX predIdx3 ON pred(nazPred, sifPred, upisanoStud DESC);
CREATE INDEX predIdx4 ON pred(brojSatiTjedno DESC, sifPred DESC);
```
7. a) 

```
CREATE DISTINCT TYPE tKgIznos AS DECIMAL(5,2);
CREATE DISTINCT TYPE tFIznos AS DECIMAL(5,2);
```

  
 b) 

```
CREATE FUNCTION KgUFunte (iznos tKgIznos) RETURNING tFIznos;
    RETURN (1/0.45 * iznos::DECIMAL(5,2))::tFIznos;
END FUNCTION;
```

```
CREATE EXPLICIT CAST (tKgIznos AS tFIznos WITH KgUFunte);
```

  
 c) 

```
CREATE TABLE ArtiklKg( oznArtikl CHAR(5),
                        , nazivArtikl CHAR(50),
                        , masa tKgIznos
                        , PRIMARY KEY (oznArtikl))
```

  
 d) 

```
SELECT oznArtikl, iznos, iznos::tFIznos FROM ArtiklKg
```

**UVOD U BAZE PODATAKA – 2. međuispit  
18. siječanj 2006.**

1. a) Na primjerima ilustrirati anomalije koje se mogu pojaviti prilikom rada s prikazanom relacijom **Projekt** (atributi koji čine primarni ključ relacije su potcrtani). Atribut **datumIzvjestaj** predstavlja datume kada je osoba na projektu podnijela svoj izvještaj o radu na tom projektu. (4 boda)

<u>sifraOsoba</u>	<u>sifraProjekt</u>	nazivProjekt	datumIzvjestaj
123456789	4	Projekt A	1.1.2005, 15.6.2005
123456789	3	Projekt B	null
111222333	4	Projekt A	1.1.2005, 15.6.2005, 10.12.2005

b) Je li relacijska shema na kojoj je definirana relacija **Projekt** u 1NF, te je li u 2NF? Ukoliko relacijska shema nije u 1NF i/ili u 2NF, potrebno je navesti **sve razloge** zbog kojih nije u toj normalnoj formi. (3 boda)

2. Oblikovati bazu podataka o agencijama za prodaju nekretnina. Agencija ima šifru i naziv. Nekretnine su šifrirane, a osim šifre evidentira se površina i vrsta nekretnine: stan, kuća ili zemljište. Za stanove se dodatno evidentira broj soba, a za kuću broj soba i površina okućnice. Istu nekretninu može prodavati samo jedna agencija.

Osoba se o nekretnini može informirati na više načina, npr. telefonskim pozivom, preko web stranice (pretpostavka je da su osobe registrirane), osobnim dolaskom u agenciju, ... Načini informiranja su šifrirani, a evidentira se šifra i opis načina informiranja.

Agencija vodi evidenciju o upitima osoba: tko se, kada i na koji način informirao o kojoj nekretnini. Vrijeme informiranja je pohranjeno kao DATETIME podatkovni tip. O istoj se nekretnini osoba u istom trenutku može informirati više puta, ali samo na različite načine (npr. gledanjem informacije o nekretnini na web stranici i telefonskim razgovorom s agencijom). Za osobu se evidentira šifra, jmbg, ime, prezime i telefonski broj. Osim šifre, osobu jednoznačno identificira i jmbg.

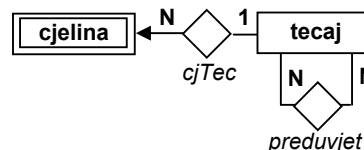
Nacrtati E-R model, opisati entitete i veze (njihove attribute i ključeve). Sve sheme moraju zadovoljavati 3NF. Uvijete spajanja u vezama, relacijski model i naredbe za kreiranje relacija **nije potrebno pisati**. (9 bodova)

3. Na slici je prikazan E-R model za segment baze podataka koji obuhvaća evidenciju tečajeva. Za slušanje tečaja potrebna su znanja stečena na drugim tečajevima, stoga neki tečaj može predstavljati preduvjet za praćenje drugih tečaja. Tečaj je podijeljen u tematske cjeline.

**entiteti**                      **atributi**                      **ključevi**

tečaj                      sifTecaj, nazTecaj                      K = { sifTecaj }

cjelina                      sifCjelina, rbrCjelina, sadrzaj                      K = { sifTecaj, rbrCjelina }



Napisati niz SQL naredbi kojima će se kreirati svi potrebni objekti za realizaciju prikazanog E-R modela u relacijskoj bazi podataka. Redni broj cjeline je pozitivna cijeli broj. Ostale tipove podataka odabrati prema želji. Pri tome:

- moraju biti zadovoljena sva ograničenja integriteta koja proizlaze iz modela
- ne mogu postojati dva tečaja istog naziva
- brisanjem tečaja moraju biti obrisane i sve cjeline tog tečaja
- tečaj može imati najviše 30 cjelina
- tematske cjeline tečaja moraju se unositi redom (tj. da bi se upisala 2. cjelina tečaja mora postojati 1. cjelina tog tečaja, a da bi se evidentirala 1. cjelina tečaja, za tečaj ne smije biti evidentirana niti jedna cjelina). Pokušaj unosa tematske cjeline koja ne zadovoljava navedeno pravilo spriječiti pozivom **postojeće** pohranjene procedure **pogreska** koja dojavljuje odgovarajuću pogrešku (procedure **pogreska ne treba pisati**). (9 bodova)

4. Na relaciji  $r(A, B, C, D, E)$  vrijede funkcijske zavisnosti iz skupa  $F = \{ AD \rightarrow C, CD \rightarrow E \}$ . Korištenjem Armstrongovih aksioma i pravila koja iz njih proizlaze (**osim pravila o pseudotranzitivnosti**), dokažite da na relaciji  $r$  vrijedi funkcijska zavisnost  $AD \rightarrow E$ .

**Napomena:** dokaz mora biti jasno napisan, kao niz izjava oblika: *prema p, iz x i y  $\Rightarrow$  z*

pri čemu je **p naziv** nekog pravila ili aksioma, **x, y, z** su neke funkcijske zavisnosti. Definicije pravila i aksioma nije potrebno pisati, dovoljno je navesti samo **naziv**. (4 boda)

5. Nacrtati  $B^+$ -stablo reda 5, čiji su čvorovi najmanje popunjeni. Stablo sadrži kazaljke na zapise čiji su ključevi cijeli brojevi u intervalu [1, 12]. (4 boda)

6. Na koje načine se može XML dokument pohraniti u relacijsku bazu podataka? Za svaki način navesti prednosti i nedostatke. (4 boda)

7. Zadane su relacije:

<u>r</u> (A D)	<u>s</u> (A B C)	<u>t</u> (C D)
A 1	A null null	5 null
null 1	B 4 1	null 2
B 2	null 2 2	1 2
		1 1

Napisati **rezultat** obavljanja sljedeće operacije (SQL naredbe **nije** potrebno pisati). **Obavezno napisati i međurezultat.**

a)  $t \bowtie_{D < B}^* s \bowtie_{D < B} r$  (3 boda)

Sljedeći zadatak riješiti **upotrebom jedne SQL naredbe** (koriste se prikazane relacije):

b)  $\pi_{C,D}(r \bowtie_{D < B} s) \cap t$  (4 boda)

8. Baza podataka sadrži podatke o projektima i vremenima utrošenim na rad projekata od strane osoba uključenih u razvoj projekata. Podvučeni atributi čine primarni ključ relacije. Niži jedan atribut ne može poprimiti NULL vrijednost.

**PROJEKT** - opisuje projekte

<u>sifProjekt</u>	INTEGER	šifra projekta
<u>nazivProjekt</u>	CHAR(50)	naziv projekta
<u>opisProjekt</u>	CHAR(150)	opis projekta
<u>datumPocetak</u>	DATE	datum početka projekta
<u>datumZavrsetak</u>	DATE	datum završetka projekta

**ZADATAK** - opisuje zadatke na projektu

<u>sifZadatak</u>	INTEGER	šifra zadatka
<u>sifProjekt</u>	INTEGER	šifra projekta
<u>nazivZadatak</u>	CHAR(50)	naziv zadatka
<u>opisZadatak</u>	CHAR(150)	opis zadatka

**OSOBA** - opisuje osobe uključene u razvoj projekata

<u>sifOsoba</u>	INTEGER	šifra osobe
<u>imeOsoba</u>	CHAR(50)	ime osobe
<u>prezOsoba</u>	CHAR(50)	prezime osobe
<u>emailOsoba</u>	CHAR(30)	e-mail adresa osobe

**DNEVNI\_PLAN** - opisuje dnevno utrošeno vrijeme osobe na zadacima

<u>sifZadatak</u>	INTEGER	šifra zadatka
<u>datumZadatak</u>	DATE	datum obavljanja zadatka
<u>sifOsoba</u>	INTEGER	šifra osobe
<u>brojSati</u>	SMALLINT	broj sati utrošenih na rad na zadatku

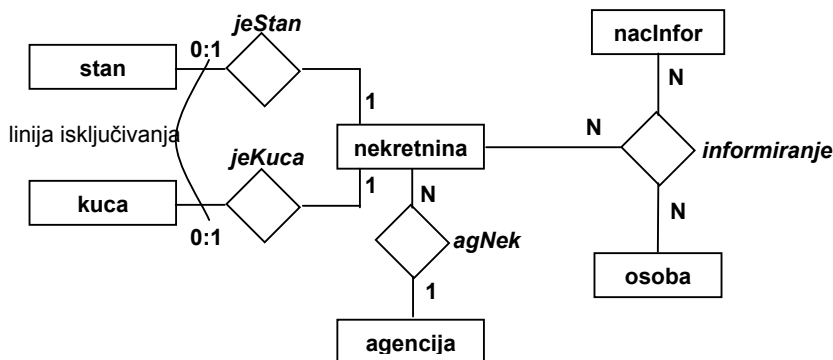
Napisati po jedan SQL upit za svaki od zadataka. Upiti moraju biti neovisni o datumu izvođenja. Neće se priznavati rješenja u kojima se nepotrebno koriste podupiti.

- Za svaki projekt ispisati njegov naziv i ukupan broj sati utrošenih na izradu tog projekta tekuće godine. Bliže početku liste trebaju se nalaziti projekti sa većim brojem utrošenih sati. Projekte s jednakim brojem utrošenih sati poredati po abecedi. (4 boda)
- Ispišite sve podatke o osobama koje su utrošile više vremena na radu na projektima u mjesecu ožujku bilo koje godine, nego ijedna druga osoba radeći na projektima radnim danima. (4 boda)
- Svim projektima koji su završeni prije današnjeg dana postaviti vrijednost datuma završetka na datum najkasnijeg dnevnog plana koji se odnosi na taj projekt. (4 boda)
- Ispisati šifru, naziv i opis projekta za sve projekte za koje postoji više od 10 definiranih zadataka i sve one projekte koji su trajali manje od 100 dana i na njima je radilo više od 15 ljudi. Projekt se u listi pojavljuje samo jednom. (4 boda)

UVOD U BAZE PODATAKA – RJEŠENJA 2. međuispita  
18. siječanj 2006.

1. a) – anomalija unosa – ne mogu se unijeti podaci o osobama koje nisu na projektima, niti o projektima kojima nije još dodijeljena niti jedna osoba
  - anomalija izmjena - promjenu naziva projekta potrebno je provesti u svim zapisima koji se odnose na taj projekt
  - anomalija brisanja - brisanjem svih podataka o raspoređenim osobama na projektu, gube se podaci o projektu
- b) – relacijska shema nije u 1NF – domena atributa **datumlzvještaj** ne sadrži jednostavne vrijednosti već skup vrijednosti
  - relacijska shema nije u 2NF – nije u 1NF i postoji nepotpuna funkcijska ovisnost atributa **nazivProjekt** o ključu, jer **sifraProjekt** → **nazivProjekt**

2.



<b>agencija</b> sifAgencija nazAgencija K={sifAgencija }	<b>nekretnina</b> sifNek vrstaNek površina K={ sifNek }	<b>stan</b> sifNek brSoba K={ sifNek }	<b>kuca</b> sifNek brSoba površinaOkuc K={ sifNek }	<b>osoba</b> sifOsoba jmbg prezOsoba imeOsoba telOsoba K1={ sifOsoba } K2={ jmbg }	<b>nacInfor</b> sifNacInfor opisNacInfor K={ sifNacInfor }
<b>agNek</b> sifNek sifAgencija K={ sifNek }	<b>jeStan</b> sifNek K={ sifNek }	<b>jeKuca</b> sifNek K={ sifNek }	<b>informiranje</b> sifOsoba sifNek sifNacInfor vrijemeInfor K={ sifKupac, sifNek, sifNacInfor, vrijemeInfor }		

```

3. CREATE TABLE tecaj ( sifTecaj  INTEGER PRIMARY KEY
                        , nazTecaj  CHAR(40) UNIQUE)

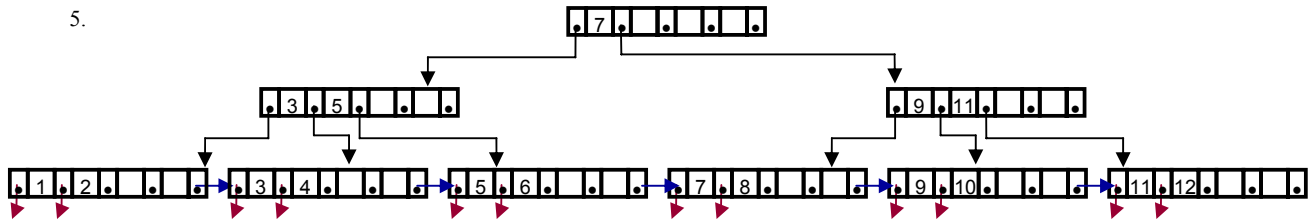
CREATE TABLE preduvjet ( sifTecaj  INTEGER REFERENCES tecaj(sifTecaj)
                        , sifTecajPred INTEGER REFERENCES tecaj (sifTecaj)
                        , PRIMARY KEY (sifTecaj, sifTecajPred ))

CREATE TABLE cjelina (sifTecaj  INTEGER REFERENCES tecaj(sifTecaj) ON DELETE CASCADE
                        , rbrCjelina SMALLINT CHECK (rbrCjelina BETWEEN 1 AND 30)
                        , sadrzaj  CHAR(40)
                        , PRIMARY KEY (sifTecaj, rbrCjelina))

CREATE TRIGGER insCjelina
INSERT ON cjelina
REFERENCING NEW AS nova
FOR EACH ROW
WHEN (nova.rbrCjelina != (SELECT COUNT(*) FROM cjelina
                        WHERE cjelina.sifTecaj = nova.sifTecaj))
(EXECUTE PROCEDURE pogreska ());
    
```

4.  $D \rightarrow D$  – aksiom o refleksivnosti
- $D \rightarrow D \Rightarrow AD \rightarrow D$  – aksiom o uvećanju
- $AD \rightarrow D \wedge AD \rightarrow C \Rightarrow AD \rightarrow CD$  – pravilo unije (aditivnosti)
- $AD \rightarrow CD \wedge CD \rightarrow E \Rightarrow AD \rightarrow E$  – aksiom o tranzitivnosti

5.



6. vidjeti predavanja

7. a)

$t \triangleright \triangleleft^* s =$ $D < B$	(C'	D	A	B	C)
	null	null	A	null	null
	null	2	B	4	1
	1	2	B	4	1
	1	1	B	4	1
	1	1	null	2	2

$t \triangleright \triangleleft^* s \triangleright \triangleleft r =$ $D < B$	(C'	D	A	B	C)
	null	2	B	4	1
	1	2	B	4	1

b) SELECT DISTINCT r.D, s.C FROM r, s, t

WHERE r.A = s.A

AND (s.C = t.C OR s.C IS NULL AND t.C IS NULL) -- presjek se može riješiti i pomoću EXISTS

AND (r.D = t.D OR r.D IS NULL AND t.D IS NULL)

8. a) SELECT nazivProjekt, SUM (brojSati)  
FROM projekt, zadatak, dnevni\_plan  
WHERE projekt.sifProjekt = zadatak.sifProjekt  
AND zadatak.sifZadatak = dnevni\_plan.sifZadatak  
AND YEAR(datumZadatak) = YEAR(TODAY)  
GROUP BY projekt.sifProjekt, nazivProjekt  
ORDER BY 2 DESC, 1

b) SELECT osoba.\* FROM osoba, dnevni\_plan  
WHERE dnevni\_plan.sifOsoba = osoba.sifOsoba  
AND MONTH(datumZadatak) = 3  
GROUP BY 1,2,3,4  
HAVING SUM(brojSati) > ALL (SELECT SUM(brojSati)  
FROM dnevni\_plan  
WHERE dnevni\_plan.sifOsoba <> osoba.sifOsoba  
AND WEEKDAY(dnevni\_plan.datumZadatak) NOT IN (0,6)  
GROUP BY dnevni\_plan.sifOsoba)

c) UPDATE projekt SET datumZavrsetak = (SELECT MAX (datumZadatak)  
FROM dnevni\_plan , zadatak  
WHERE dnevni\_plan.sifZadatak = zadatak.sifZadatak  
AND zadatak.sifProjekt = projekt.sifProjekt)  
WHERE datumZavrsetak < TODAY

d) SELECT sifProjekt, nazivProjekt, opisProjekt  
FROM projekt  
WHERE (SELECT COUNT(\*) FROM zadatak  
WHERE zadatak.sifProjekt = projekt.sifProjekt) > 10  
OR (projekt.datumZavrsetak - projekt.datumPocetak < 100 AND  
(SELECT COUNT (DISTINCT sifOsoba) FROM dnevni\_plan, zadatak  
WHERE dnevni\_plan.sifZadatak = zadatak.sifZadatak  
AND projekt.sifProjekt = zadatak.sifProjekt) > 15)

-- zadatak se može riješiti i pomoću UNION, te korištenjem grupiranja