

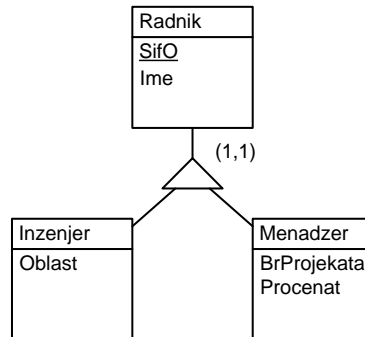
Baze podataka

Ispit traje **180** minuta

Ime prezime i broj indeksa studenta	Potpis dežurnog	Broj poena

Napomena: Nije dozvoljena upotreba literature.

1. (8) Za model entiteta i odnosa, prikazan na slici, kreirati SQL iskaze kojim se definišu sva ograničenja koja iz modela proističu.



Odgovor:

```

CREATE TABLE Radnik (SifO INT PRIMARY KEY, ...)
CREATE TABLE Inzenjer (SifO INT PRIMAR KEY REFERENCES Radnik, ...)
CREATE TABLE Menadzer (SifO INT PRIMAR KEY REFERENCES Radnik, ...)
  
```

```

CREATE ASSERTION Specijalizacija
CHECK ( NOT EXISTS (SELECT *
                    FROM Radnik
                    WHERE (SifO NOT IN (SELECT SifO FROM Inzenjer)
                          AND
                          SifO NOT IN (SELECT SifO FROM Menadzer)
                          )
                    OR
                    (SifO IN (SELECT SifO FROM Inzenjer)
                     AND
                     SifO IN (SELECT SifO FROM Menadzer)
                     )
                    )
  )
  
```

2.(5) Za šemu relacione baze podataka koja nastaje na osnovu modela prikaznog u zadatku 1, treba sastaviti iskaz relacione algebre koji vraća parove šifara Menadžera i Inženjera koji imaju isto ime.

Odgovor:

$$\pi_{SifO, Ime}(Radnik \bowtie Inzenjer) \rightarrow t_1(SifO, Ime)$$

$$\pi_{SifO, Ime}(Radnik \bowtie Menadzer) \rightarrow t_2(SifO, Ime)$$

$$\pi_{t_1.SifO, t_2.SifO}(\underset{t_1.Ime=t_2.Ime}{t_1 \times t_2}) \rightarrow Resenje(SifOI, SifOM)$$

3.(5) Za svaki od navedenih parova vrednosti atributa Katedra i Iskustvo navesti da li narušava navedeno ograničenje (staviti oznaku X pored onih koji narušavaju):

CONSTRAINT Test CHECK (CASE WHEN Katedra = 'RTI'
THEN CASE WHEN Iskustvo < 4
THEN 1 ELSE 0 END
ELSE 1 END = 1)

Katedra	Iskustvo	Ograničenje
RTI	NULL	x
RTI	6	x
EL	3	√
NULL	NULL	√
RTI	2	√
NULL	1	√

4.(18) Dati su šema relacije R(A, B, C, D, F, G) i skup funkcijskih zavisnosti $F=\{AB \rightarrow CD, BC \rightarrow FG, A \rightarrow G, G \rightarrow B, C \rightarrow G\}$. Potrebno je:

a) Odrediti skup kandidat ključeva KK date šeme.

Odgovor: $KK=\{A\}$

b) Ispitati redom da li je data šema u BC, 3. i 2. normalnoj formi i svaki put u tabeli naznačiti da li posmatrana zavisnost narušava posmatranu normalnu formu.

	$AB \rightarrow CD$	$BC \rightarrow FG$	$A \rightarrow G$	$G \rightarrow B$	$C \rightarrow G$
BCNF	√	x	√	x	x
3NF	√	x	√	x	x
2NF	√	√	√	√	√

c) Sprovesti normalizaciju date šeme u 3. normalnu formu algoritmom koji garantuje očuvanje funkcijskih zavisnosti.

Odgovor:

$R_1(\underline{A}, C, D) \quad R_2(\underline{C}, F, G) \quad R_3(\underline{G}, B)$

d) Da li je dekompozicija relacije R na relacije $R_1(A, B, C, D)$ i $R_2(A, C, F, G)$ bez gubitaka pri spajanju? Ako jeste, onda objasniti zašto, a potom odrediti u kojoj se normalnoj formi nalazi ova dekompozicija.

Odgovor:

$R_1 \cap R_2 \rightarrow \{A, C\}; \{A, C\} \rightarrow R_1; \{A, C\} \rightarrow R_2 \quad 2NF$

5.(34)

a) Ako se za oporavak od kvara koristi mehanizam Prateće stranice. Podaci A, B, C i D nalaze se u različitim stranicama na disku (podatak A u stranici 1,... , D u stranici 4). Prikazati izgled relevantnih delova sistema u trenutku kvara, neposredno posle trenutka t_{15} ? Koje operacije mehanizam Prateće Stranice vrši posle popravke sistema u cilju vraćanja baze u konzistentno stanje?

Odgovor:

T1	7	1	A
	10	2	B
		3	C
		4	D
		5	C_1
		6	C_2
T2	5	7	C_3
	9	8	A_1
		9	D_1
		10	A_2
		11	
		12	
T3	6	13	
	8	14	
	2	15	
		16	
		17	
		18	
	19		

T1	7
	10
T2	5
	9
T3	3
	1
	2

Vreme	T_1	T_2	T_3
t_1		Read (C)	
t_2		Write (C)	
t_3			Read (C)
t_4			Write (C)
t_5	Read (C)		
t_6	Write (C)		
t_7			Read (A)
t_8			Write (A)
t_9		Read (D)	
t_{10}		Write (D)	
t_{11}		Commit	
t_{12}	Read (A)		
t_{13}	Write (A)		
t_{14}	Commit		
t_{15}			Read (B)
t_{16}			Write (B)
t_{17}			Commit

Oporavak: **Restart** {T₃}

b) Koje su vrednosti svakog od podatka u bazi nakon uspešnog završetka prikazanog redosleda ukoliko je poznato da pre početka izvršavanja datog redosleda podaci A, B, C i D imaju vrednosti 15, 20, 50 i 60 respektivno, i ukoliko se pretpostavi da se sa svakim pročitanim podatkom unutar transakcije obavlja obrada $X=2*X$ (tj. kao da se za svaki podatak koji transakcija koristi, vrši obradu Read(X); $X=2*X$; Write (X))?

Odgovor:

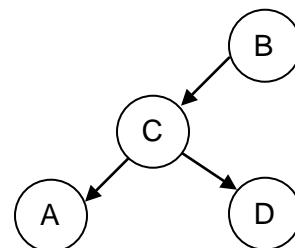
Podatak	Vrednosti
A	30
B	40
C	100
D	120

c) Da li je redosled prikazan na slici moguć ukoliko se uvede mehanizam zaključavanja po protokolu u obliku stabla? Ukoliko nije, navesti zašto nije, a ukoliko jeste onda treba popuniti datu tabelu za to izvršavanje (uz dodate instrukcije zaključavanja/otključavanja).

Odgovor: **Redosled je moguć.**

Vreme	T ₁	T ₂	T ₃
t ₁		L(C)	
t ₂		Rd(C)	
t ₃		Wr(C)	
t ₄		L(D)	
t ₅		Un(C)	
t ₆			L(B)
t ₇			L(C)
t ₈			Rd(C)
t ₉			Wr(C)
t ₁₀			L(A)
t ₁₁			Un(C)
t ₁₂	L(C)		
t ₁₃	Rd(C)		
t ₁₄	Wr(C)		
t ₁₅			Rd(A)
t ₁₆			Wr(A)
t ₁₇			Un(A)
t ₁₈		Rd(D)	
t ₁₉		Wr(D)	
t ₂₀		Un(D)	
t ₂₁		Commit	
t ₂₂	L(A)		
t ₂₃	Un(C)		
t ₂₄	Rd(A)		
t ₂₅	Wr(A)		
t ₂₆	Un(A)		
t ₂₇	Commit		
t ₂₈			Rd(B)
t ₂₉			Wr(B)
t ₃₀			Un(B)
t ₃₁			Commit
t ₃₂			
t ₃₃			
t ₃₄			
t ₃₅			

t ₃₆			
t ₃₇			
t ₃₈			
t ₃₉			
t ₄₀			
t ₄₁			
t ₄₂			
t ₄₃			
t ₄₄			
t ₄₅			
t ₄₆			
t ₄₇			
t ₄₈			
t ₄₉			
t ₅₀			
t ₅₁			
t ₅₂			
t ₅₃			
t ₅₄			
t ₅₅			
t ₅₆			
t ₅₇			
t ₅₈			
t ₅₉			
t ₆₀			
t ₆₁			



d) Neka se za redosled iz tačke a) za oporavak od kvara koristi mehanizam Sistemskog Dnevnika sa Neodložnim Upisom. Prikazati izgled sistemskog dnevnika i navesti operacije koje sistem zadaje posle popravke sistema, u cilju vraćanja baze u konzistentno stanje ukoliko je kvar nastao neposredno posle trenutka t_{17} ?

Odgovor:

<T2, starts>

<T2, C, C0, C1>

<T3, starts>

<T3, C, C1, C2>

<T1, starts>

<T1, C, C2, C3>

<T3, A, A0, A1>

<T2, D, D0, D1>

<T2, commits>

<T1, A, A1, A2>

<T1, commits >

<T3, B, B0, B1>

<T3, commits>

Oporavak: Redo {T1, T2, T3}

Odnosno, ako nije upisan T3 commit: Undo{T3}, Redo{T1, T2} Restart {T3}

e) Koje operacije zadaje mehanizam u cilju vraćanja baze u konzistentno stanje u slučaju opisanom u tački pod d) ukoliko je poznato da je u trenutku neposredno posle Write(A) operacije transakcije T_1 urađena kontrolna tačka koja je završena u trenutku neposredno posle Read(B) operacije transakcije T_3 ? Kako će izgledati sistemski dnevnik u tom slučaju? Obeležiti ulaze sistemskog dnevnika koji mogu biti oslobođeni nakon izvršene kontrolne tačke.

Odgovor:

<T2, starts> <----- može biti oslobođeno nakon kontrolne tačke

<T2, C, C0, C1> <----- može biti oslobođeno nakon kontrolne tačke

<T3, starts>

<T3, C, C1, C2>

<T1, starts>

<T1, C, C2, C3>

<T3, A, A0, A1>

<T2, D, D0, D1>

<T2, commits>

<T1, A, A1, A2>

<CHK (T1, T3)>

<T1, commits >

<CHK END>

<T3, B, B0, B1>

<T3, commits> Oporavak: Redo {T1, T3}

Odnosno, ako nije upisan T3 commit: Undo{T3}, Redo{T1} Restart {T3}