

Baze podatakaIspit traje **180** minuta

Ime prezime i broj indeksa studenta	Potpis dežurnog	Broj poena

Napomena: Nije dozvoljena upotreba literature.

1.(4) Pokazati šta je rezultat sledećeg upita za dati sadržaj posmatranih tabela:

PTable	Sif	Dt	Qt	Pr	Weight
	10	1	20	1	800
	20	1	10	2	700
	30	2	10	1	900
	40	2	20	3	1000
	50	2	30	1	700
	60	5	10	1	700
	70	4	20	4	900

```

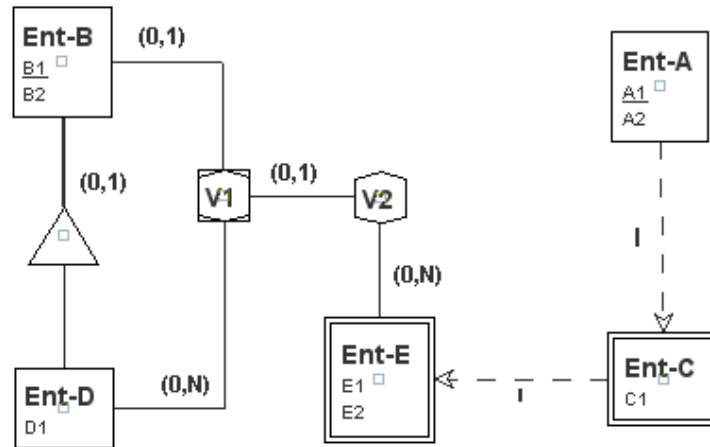
SELECT P1.Sif, P1.Weight
FROM PTable AS P1, PTable AS P2
WHERE P1.Weight >= P2.Weight
GROUP BY P1.Sif, P1.Weight
HAVING (SELECT COUNT(*) FROM PTable) - COUNT(*) + 1 <= 3;

```

Odgovor:

30	900						
40	1000						
70	900						

2. (4) Model entiteta i odnosa, prikazan na slici, prevesti u šemu relacione baze podataka, uz naznaku svih stranih ključeva zaokruživanjem.



Odgovor:

Ent-A: (A1, A2)
 Ent-B: (B1, B2)
 Ent-C: (A1, C1)
 Ent-D: (B1, D1)
 Ent-E: (A1, E1, E2)
 V1: (B1B, B1D)
 V2: (B1B, A1)

3.(8) Dat je deo šeme relacione baze podataka lanca prodavnica. Potrebno je napisati iskaz relacione algebre koji vraća isti rezultat kao i dati SQL upit.

PRODAVAC (SifP, Ime); RACUN (SifR, SifP, Stanje)

```
SELECT DISTINCT P.Ime
FROM Prodavac P
WHERE NOT EXISTS ( SELECT R.SifR
                   FROM Racun R
                   WHERE R.SifP = P.SifP AND R.Stanje <= 0)
```

Odgovor:

$$\pi_{Ime} \left(Prodavac \cap \left(\pi_{SifP}(Prodavac) - \pi_{SifP}(\sigma_{Stanje \leq 0}(Racun)) \right) \right) \rightarrow Rez(Ime)$$

4.(9) Dati su šema relacije $D(R, S, T, U, X)$ i skup funkcijskih zavisnosti $F = \{U \rightarrow RS, XUS \rightarrow R, T \rightarrow SU, R \rightarrow UT\}$. Potrebno je:

a) Odrediti skup kandidat ključeva KK date šeme.

Odgovor:

$KK = \{XU, XT, XR\}$

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.

	$U \rightarrow RS$	$XUS \rightarrow R$	$T \rightarrow SU$	$R \rightarrow UT$
BCNF	x	√	x	x
3NF	x	√	x	√
2NF	x	√	x	√

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

Odgovor:

$D_1(\underline{U}, R)$ $D_2(\underline{I}, S, U)$ $D_3(\underline{R}, T)$ $D_4(\underline{X}, \underline{R})$

5.(9) Dati su šema relacije $R(A, C, D, E, F)$ i skup funkcijskih zavisnosti $F = \{A \rightarrow D, C \rightarrow E, F \rightarrow A, E \rightarrow F\}$. Navodeći konkretnu relaciju nad šemom R pokazati da dekompozicija šeme R na $R_1(A, D)$, $R_2(C, E)$ i $R_3(F, A)$ nije bez gubitaka.

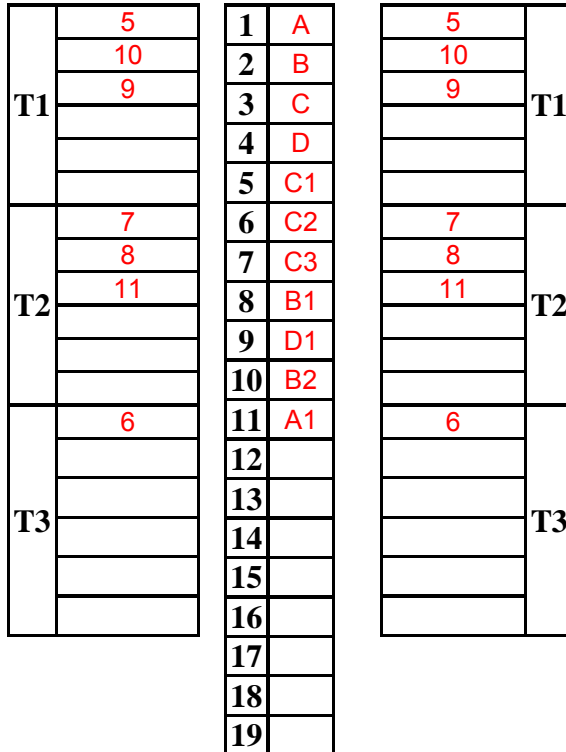
Odgovor:

A	C	D	E	F
a0	c1	d0	e1	f0
a0	c0	d0	e0	f1
a0	c2	d0	e0	f1

6.(36)

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_{24} ? Koje operacije mehanizam Prateće Stranice vrši posle popravke sistema u cilju vraćanja baze u konzistentno stanje?

Odgovor:



Vreme	T_1	T_2	T_3
t_1	Read (C)		
t_2	Read (B)		
t_3	$C := C + B$		
t_4	Write (C)		
t_5	$B := B - 5$		
t_6			Read (C)
t_7			$C := 50$
t_8			Write (C)
t_9			Commit
t_{10}		Read (C)	
t_{11}		$C := C + 1$	
t_{12}		Write (C)	
t_{13}		Read (B)	
t_{14}		$B := B - 1$	
t_{15}		Write (B)	
t_{16}	Read (D)		
t_{17}	$D := D + C$		
t_{18}	Write (D)		
t_{19}	Write (B)		
t_{20}	Commit		
t_{21}		Read (A)	
t_{22}		$A := A * 2$	
t_{23}		Write (A)	
t_{24}		Commit	

Oporavak:

1. Commit izvršen (prikazano na slici):
Oporavak: nije potreban
2. Commit nije izvršen:
Oporavak: Restart { T_2 }

b) Da li (u slučaju opisanom u prethodnoj tački) vrednosti u bazi podataka nakon uspešnog izvršavanja datog redosleda odgovaraju nekom serijskom redosledu? Navesti sve takve redoslede.

Odgovor:

Ne odgovaraju nijednom serijskom redosledu.

c) Uvesti mehanizam striktnog vremenskog markiranja i proveriti da li se dobija isti redosled. Vrednosti vremenskih marki su $TS(T_1)=300$, $TS(T_2)=200$, $TS(T_3)=100$. Pri eventualnom restartovanju transakcija odabrati najpovoljniji slučaj (sa najmanjim brojem restartovanja).

Odgovor: **Redosled se promenio.**

T	Op	S	RA	WA	CA	RB	WB	CB	RC	WC	CC	RD	WD	CD	TS(T ₁)	TS(T ₂)	TS(T ₃)
			0	0	1	0	0	1	0	0	1	0	0	1	300	200	100
T ₁	Rd(C)	Ok							300								
T ₁	Rd(B)	Ok				300											
T ₁	Wr(C)	Ok								300	0						
T ₃	Rd(C)	Rb															400
T ₂	Rd(C)	Rb														500	
T ₁	Rd(D)	Ok										300					
T ₁	Wr(D)	Ok											300	0			
T ₁	Wr(B)	Ok					300	0									
T ₁	Commit	Ok						1			1			1			
T ₃	Rd(C)	Ok							400								
T ₃	Wr(C)	Ok								400	0						
T ₃	Commit	Ok									1						
T ₂	Rd(C)	Ok							500								
T ₂	Wr(C)	Ok								500	0						
T ₂	Rd(B)	Ok				500											
T ₂	Wr(B)	Ok					500	0									
T ₂	Rd(A)	Ok	500														
T ₂	Wr(A)	Ok		500	0												
T ₂	Commit	Ok			1			1			1						

d) Neka se za redosled iz tačke a) za oporavak od kvara koristi mehanizam Sistemskog Dnevnika sa Neodloženim 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_{21} ?

Odgovor:

< T1, starts >
 < T1, C, C₀, C₁ >
 < T3, starts >
 < T3, C, C₁, C₂ >
 < T3, commits >
 < T2, starts >
 < T2, C, C₂, C₃ >
 < T2, B, B₀, B₁ > Undo{ T₂} Redo{ T₁, T₃} Restart{ T₂}
 < T1, D, D₀, D₁ >
 < T1, B, B₁, B₂ >
 < T1, commits >

e) Koje su moguće vrednosti svakog od podataka u bazi neposredno pre trenutka kvara opisanog u tački pod d) ukoliko je poznato da pre početka izvršavanja transakcija podaci A, B, C i D imaju vrednosti 10, 20, 30 i 40 respektivno?

Odgovor:

Vreme	Podatak	Moguće vrednosti
t_{21}	A	10
t_{21}	B	20, 15, 19
t_{21}	C	30, 50, 51
t_{21}	D	40, 90

f) 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(C) operacije transakcije T₂ urađena kontrolna tačka koja je završena u trenutku neposredno posle Read(D) operacije transakcije T₁? Kako će izgledati sistemski dnevnika u tom slučaju?

Odgovor:

< T1, starts >
 < T1, C, C₀, C₁ >
 < T3, starts >
 < T3, C, C₁, C₂ >
 < T3, commits >
 < T2, starts >
 < T2, C, C₂, C₃ >
 <CHK (T1, T2)>
 < T2, B, B₀, B₁ > Undo{ T₂} Redo{ T₁} Restart{ T₂}
 <CHK END>
 < T1, D, D₀, D₁ >
 < T1, B, B₁, B₂ >
 < T1, commits >