

COMPITO DI SISTEMI INFORMATIVI/BASI DI DATI II

13 dicembre 2006 (Tot. 16) Tempo: 2h

Esercizio 1 (punti 3)

Si consideri il seguente log:

1. B(T2)
2. U(T1,O1,B2,A2)
3. I(T2,O2,A3)
4. B(T3)
5. D(T3,O1,B5)
6. U(T2,O2,B6,A6)
7. U(T1,O2,B7,A7)
8. C(T1)
9. B(T4)
10. I(T4,O3,A10)
11. U(T2,O3,B11,A11)
12. D(T3,O2,B12)
13. I(T4,O4,A13)
14. U(T2,O4,B14,A14)
15. CK(T2,T3,T4)
16. U(T4,O3,B16,A16)
17. I(T3,O5,A17)
18. U(T2,O4,B18,A18)
19. C(T2)
20. B(T5)
21. U(T5,O3,B21,A21)
22. I(T3,O6,A22)
23. U(T4,O6,B23,A23)
24. B(T6)
25. D(T6,O3,B25)
26. U(T5,O6,B26,A26)
27. D(T4,O6,B27)
28. C(T4)
29. U(T6,O5,B29,A29)

si mostrino le operazioni di recovery da effettuare supponendo che il guasto avvenga subito dopo l'ultimo record del log.

Esercizio 2 (punti 4)

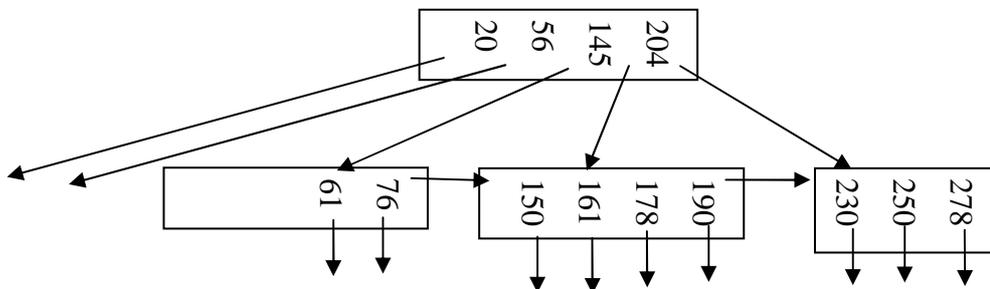
Dato il seguente schedule:

r1(x) r2(x) r3(y) r1(y) r2(z) r3(z) w1(w) w3(y) r1(x) w2(x) w1(w) w2(z) r2(y)

si indichi se rispetta il two-phase locking. Nel caso lo rispetti, si indichi una sequenza di acquisizione e rilascio di lock compatibile con lo schedule e che rispetti il two-phase locking. Nel caso non lo rispetti, si indichi almeno uno dei conflitti.

Esercizio 3 (punti 3)

Sia dato il seguente B+ tree di ordine 4.



Si mostri come si modifica l'albero nel caso di inserimento della chiave 165.

Esercizio 4 (punti 6)

Siano date le relazioni

Vendita(CodVendita,Data,Negoziio, CodArticolo, Quantità)

Articolo(CodArticolo,Descrizione,Prezzo)

la query

```
SELECT V.*, A.*
```

```
FROM Vendita AS V, Articolo AS A
```

```
WHERE V.CodArticolo=A.CodArticolo AND A.Prezzo>=10
```

e i parametri:

buffer di memoria centrale disponibili: $M=1.000$

dimensione del buffer: $B=2.000$ bytes

numero di tuple: $T(\text{Vendita})=50.000$, $T(\text{Articolo})=3.000$,

dimensione delle tuple: $S(\text{Vendita})=500$ bytes, $S(\text{Articolo})=1000$,

$S(\text{CodArticolo})=40$

numero di valori: $V(\text{Vendita},\text{CodArticolo})=2.000$, $V(\text{Articolo},\text{CodArticolo})=3.000$,

Si indichi qual'è la sequenza migliore per effettuare l'operazione e si indichi il costo di tale sequenza.

In particolare, occorrerà considerare le due sequenze:

$(\sigma_{\text{Prezzo} \geq 10} \text{Articolo}) \bowtie \text{Vendita}$

$\sigma_{\text{Prezzo} \geq 10}(\text{Articolo} \bowtie \text{Vendita})$

indicare quale delle due è la più economica e indicarne il costo.

Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record. Si supponga inoltre di non avere indici su nessun attributo e di non conoscere il dominio di prezzo.

SOLUZIONE

Esercizio 1

1. B(T2)
2. U(T1,O1,B2,A2)
3. I(T2,O2,A3)
4. B(T3)
5. D(T3,O1,B5)
6. U(T2,O2,B6,A6)
7. U(T1,O2,B7,A7)
8. C(T1)
9. B(T4)
10. I(T4,O3,A10)
11. U(T2,O3,B11,A11)
12. D(T3,O2,B12)
13. I(T4,O4,A13)
14. U(T2,O4,B14,A14)
15. CK(T2,T3,T4)
16. U(T4,O3,B16,A16)
17. I(T3,O5,A17)
18. U(T2,O4,B18,A18)
19. C(T2)
20. B(T5)
21. U(T5,O3,B21,A21)
22. I(T3,O6,A22)
23. U(T4,O6,B23,A23)
24. B(T6)
25. D(T6,O3,B25)
26. U(T5,O6,B26,A26)
27. D(T4,O6,B27)
28. C(T4)
29. U(T6,O5,B29,A29)

- 15 UNDO={T2,T3,T4} REDO={}
 - 19 UNDO={T3,T4} REDO={T2}
 - 20 UNDO={T3,T4,T5} REDO={T2}
 - 24 UNDO={T3,T4,T5,T6} REDO={T2}
 - 28 UNDO={T3,T5,T6} REDO={T2,T4}
- UNDO
- 29 O5=B29
 - 26 O6=B26
 - 25 I(O3,B25)
 - 22 D(O6)
 - 21 O3=B21
 - 17 D(O5)
 - 12 I(O2,B12)
 - 5 I(O1,B5)
- REDO
- 3 I(O2,A3)
 - 6 O2=A6
 - 10 I(O3,A10)
 - 11 O3=A11
 - 13 I(O4,A13)
 - 14 O4=A14
 - 16 O3=A16
 - 18 O4=A18
 - 23 O6=A23
 - 27 D(O6)

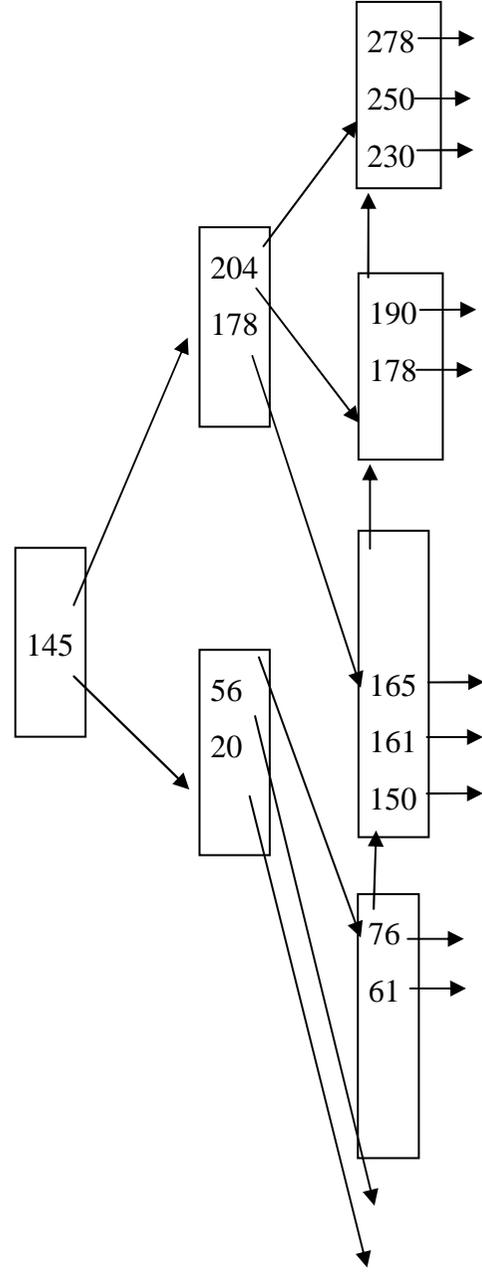
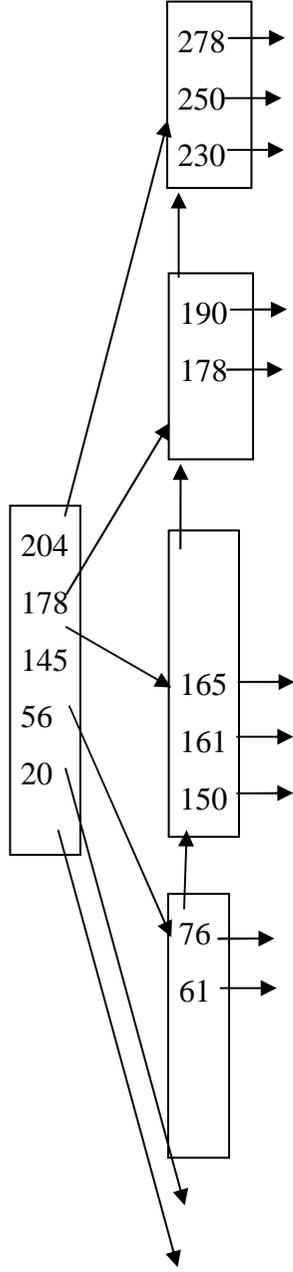
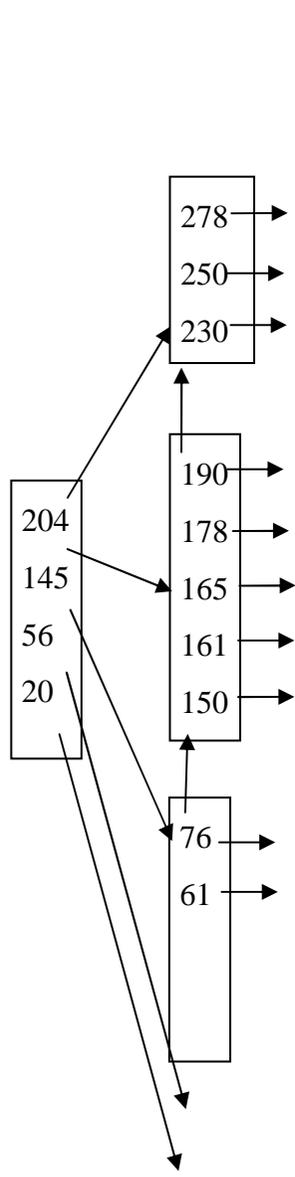
Esercizio 2

r1(x) r2(x) r3(y) r1(y) r2(z) r3(z) w1(w) w3(y) r1(x) w2(x) w1(w) w2(z) r2(y)

Lo schedule rispetta il two-phase locking. Una sequenze di acquisizione e rilascio dei lock compatibile con lo schedule e che rispetti il two-phase locking è la seguente:

Transazione 1	Transazione 2	Transazione 3
r_lock(x)		
	r_lock(x)	
		r_lock(y)
r_lock(y)		
	r_lock(z)	
		r_lock(z)
w_lock(w)		
unlock(y)		
		w_lock(y)
unlock(x)		
	w_lock(x)	
		unlock(z)
	w_lock(z)	
		unlock(y)
	r_lock(y)	
	unlock(x)	
unlock(w)		
	unlock(z)	
	unlock(y)	

Esercizio 3



Esercizio 4

buffer di memoria centrale disponibili: $M=1.000$
dimensione del buffer: $B=2.000$ bytes
numero di tuple: $T(\text{Vendita})=50.000$, $T(\text{Articolo})=3.000$,
dimensione delle tuple: $S(\text{Vendita})=500$ bytes, $S(\text{Articolo})=1000$,
 $S(\text{CodArticolo})=40$

Calcoliamo innanzitutto il numero di blocchi occupati da ciascuna relazione:
 $B(\text{Vendita})=T(\text{Vendita}) * S(\text{Vendita}) / B = 50000 * 500 / 2000 = 12500$
 $B(\text{Articolo}) = 3000 * 1000 / 2000 = 1500$

Prima sequenza: $(\sigma_{\text{Prezzo} \geq 10} \text{Articolo}) \triangleright \triangleleft \text{Vendita}$

Sia $X = \sigma_{\text{Prezzo} \geq 10} \text{Articolo}$

Dato che non conosco il dominio di prezzo, utilizzo la formula

$$\text{Costo}(X) = B(\text{Articolo}) = 1500$$

$$T(X) = T(\text{Articolo}) / 3 = 3000 / 3 = 1000$$

$$B(X) = 1000 * 1000 / 2000 = 500$$

X sta interamente in memoria centrale quindi posso usare il join a un passo

$$\text{Costo}(X \triangleright \triangleleft \text{Vendita}) = B(X) + B(\text{Vendita}) = 500 + 12500 = 13000$$

$$\text{Costo}((\sigma_{\text{Prezzo} \geq 10} \text{Articolo}) \triangleright \triangleleft \text{Vendita}) = 13000 + 1500 = 14500$$

Seconda sequenza: $\sigma_{\text{Prezzo} \geq 10}(\text{Articolo} \triangleright \triangleleft \text{Vendita})$

Sia $Y = \text{Articolo} \triangleright \triangleleft \text{Vendita}$

Né Articolo né vendita stanno in memoria centrale, quindi uso l'hash join ibrido

$$\text{Costo}(Y) = (3 - 2 * 1000 / 1500) * (1500 + 12500) = 23333$$

Dato che il costo del solo join supera quello della prima sequenza, la migliore sequenza per effettuare l'operazione è la prima e il suo costo è 14500