

COMPITO DI SISTEMI INFORMATIVI
2 aprile 2009 – Compito (Tot. 16) Tempo: 2h

Esercizio 1 (punti 3)

Si consideri il seguente log:

- | | |
|----------------------|----------------------|
| 1. I U(T2,O1,B1,A1) | 16. U(T5,O3,B16,A16) |
| 2. I(T1,O2,A2) | 17. I(T1,O6,A17) |
| 3. U(T1,O3,B3,A3) | 18. C(T1) |
| 4. U(T2,O2,B4,A4) | 19. B(T6) |
| 5. C(T2) | 20. U(T6,O6,B20,A20) |
| 6. B(T3) | 21. I(T5,O7,A21) |
| 7. U(T3,O1,B7,A7) | 22. U(T3,O7,B22,A22) |
| 8. I(T3,O4,A8) | 23. U(T6,O3,B23,A23) |
| 9. B(T4) | 24. C(T5) |
| 10. D(T4,O1,B10) | 25. U(T6,O4,B25,A25) |
| 11. U(T1,O2,B11,A11) | 26. D(T6,O5,B26) |
| 12. I(T4,O5,A12) | 27. C(T6) |
| 13. D(T4,O2,B13) | 28. U(T4,O6,B23,A28) |
| 14. B(T5) | |
| 15. CK(T1,T3,T4,T5) | |

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

Esercizio 2 (punti 4)

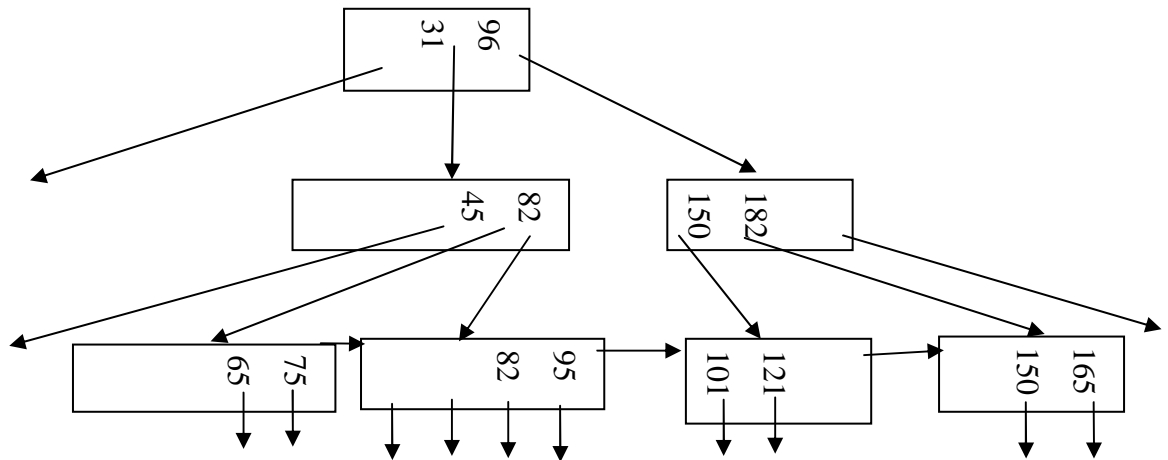
Dato il seguente schedule:

w1(x) r2(y) r2(x) w3(y) r4(y) w2(x) w5(y) w3(z) r5(y) r3(z)

si indichi se è conflict serializzabile.

Esercizio 3 (punti 3)

Sia dato il seguente B+ tree di ordine 4.



Si mostri come si modifica l'albero nel caso di cancellazione della chiave 95.

Esercizio 4 (punti 6)

Sia data la seguente relazione

Impiegato(CodiceImp, Nome, Cognome, Indirizzo, Salario, Capo),

la query

SELECT I1.*, I2.*

FROM Impiegato AS I1, Impiegato AS I2

WHERE I1.Capo = I2.CodiceImp AND I2.Salario >2000

e i parametri:

buffer di memoria centrale disponibili: $M=120.000$

dimensione del buffer: $B=400$ bytes

numero di tuple: $T(\text{Impiegato})= 200000$,

dimensione delle tuple: $S(\text{Impiegato})=1000$ $S(\text{CodiceImp})=100$

numero di valori $V(\text{Impiegato}, \text{Capo})=100$

Si calcoli il costo minimo di ciascuna delle seguenti sequenze:

$\text{Impiegato} \triangleright \triangleleft_{\text{Capo}=\text{CodiceImp}} (\sigma_{\text{Salario} > 2000} \text{ Impiegato})$

$\sigma_{\text{Salario} > 2000} (\text{Impiegato} \triangleright \triangleleft_{\text{Capo}=\text{CodiceImp}} \text{ Impiegato})$

Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record.

Si supponga che CodiceImp sia chiave primaria di Impiegato e di avere i seguenti indici:

- primario su Impiegato.Salario,
- secondario su Impiegato.CodiceImp, e
- secondario su Impiegato.Capo.

SOLUZIONE

Esercizio 1

1. U(T2,O1,B1,A1)
2. I(T1,O2,A2)
3. U(T1,O3,B3,A3)
4. U(T2,O2,B4,A4)
5. C(T2)
6. B(T3)
7. U(T3,O1,B7,A7)
8. I(T3,O4,A8)
9. B(T4)
10. D(T4,O1,B10)
11. U(T1,O2,B11,A11)
12. I(T4,O5,A12)
13. D(T4,O2,B13)
14. B(T5)
15. CK(T1,T3,T4,T5)
16. U(T5,O3,B16,A16)
17. I(T1,O6,A17)
18. C(T1)
19. B(T6)
20. U(T6,O6,B20,A20)
21. I(T5,O7,A21)
22. U(T3,O7,B22,A22)
23. U(T6,O3,B23,A23)
24. C(T5)
25. U(T6,O4,B25,A25)
26. D(T6,O5,B26)
27. C(T6)
28. U(T4,O6,B23,A28)

- 15 UNDO={T1,T3,T4,T5} REDO={ }
18 UNDO={T3,T4,T5 } REDO={T1}
19 UNDO={T3,T4,T5,T6} REDO={T1}
24 UNDO={T3,T4, T6} REDO={T1 T5}
27 UNDO={T3,T4 } REDO={T1, T5, T6}

UNDO

- 28 O6=B28
22 O7=B22
13 I(O2,B13)
12 D(O5)
10 I(O1,B10)
8 D(O4)
7 O1=B7

REDO

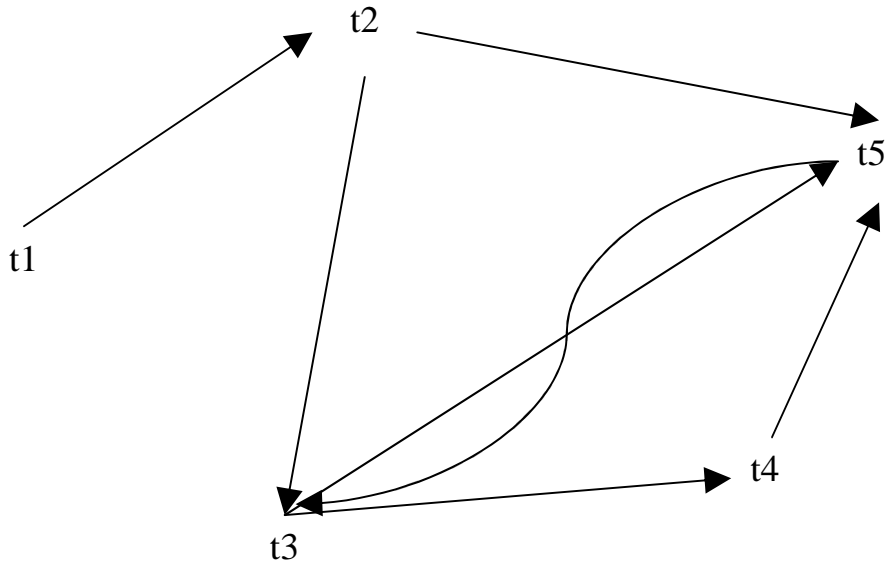
- 2 I(O2,A3)
3 O3=A3
11 O2=A11
16 O2=A16
17 I(O6,A17)
20 O6=A20
21 I(O7,A21)
23 O3=A23
25 O4=A25
26 D(O5)

Esercizio 2

Dato il seguente schedule:

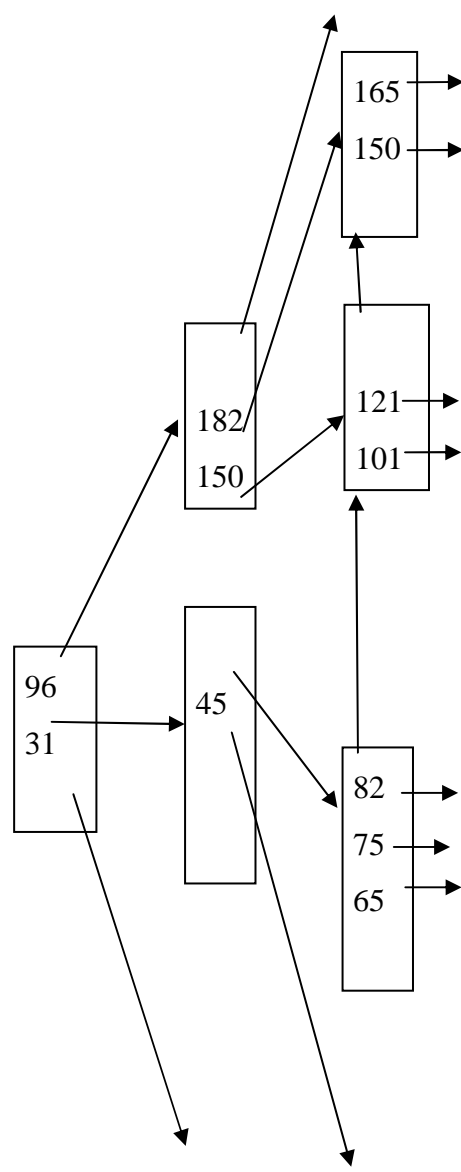
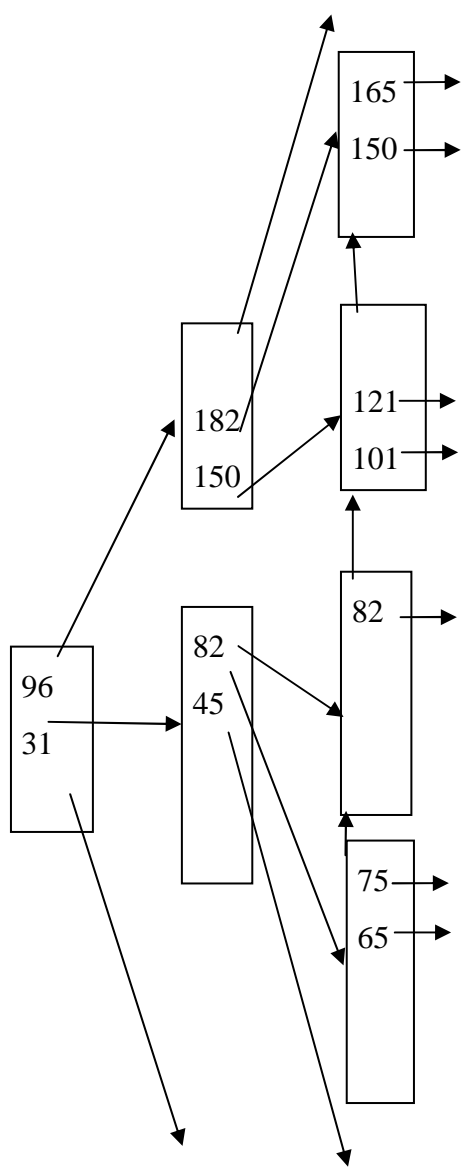
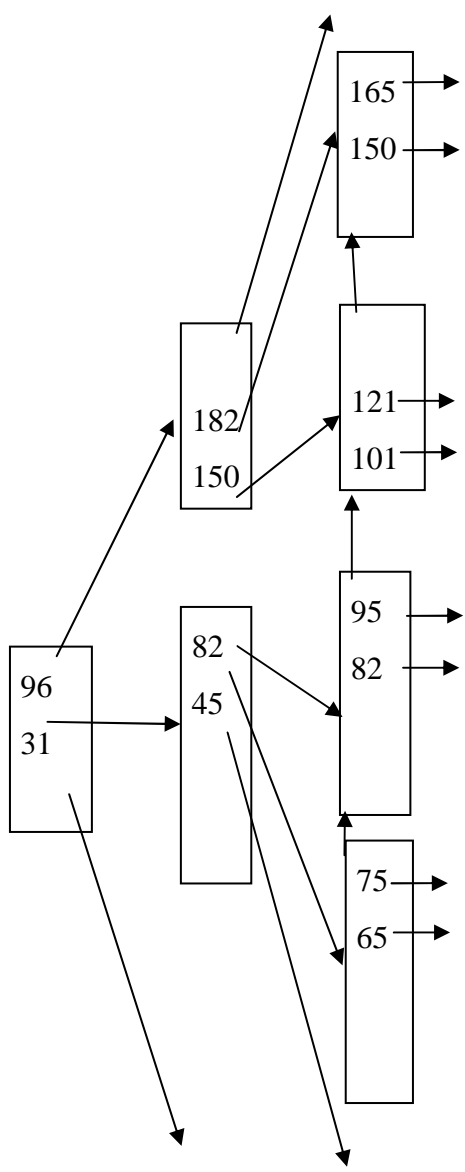
w1(x) r2(y) r2(x) w3(y) r4(y) w2(x) w5(y) w3(z) r3(y) r5(z)

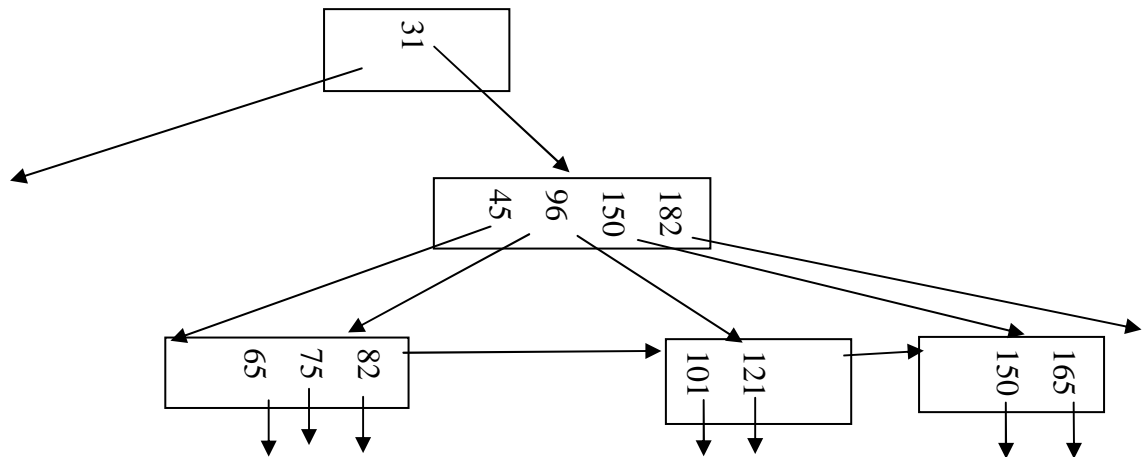
si indichi se è conflict serializzabile.



Lo schedule non è conflict serializzabile.

Esercizio 3





Esercizio 4

Parametri:

buffer di memoria centrale disponibili: $M=120.000$

dimensione del buffer: $B=400$ bytes

numero di tuple: $T(\text{Impiegato})=200000$,

dimensione delle tuple: $S(\text{Impiegato})=1000$ $S(\text{CodiceImp})=100$

numero di valori $V(\text{Impiegato,Capo})=100$

Sequenze

$(\sigma_{\text{Salario} > 2000} \text{ Impiegato}) \triangleright \triangleleft_{\text{Capo}=\text{CodiceImp}} \text{ Impiegato}$

$\sigma_{\text{Salario} > 2000} (\text{Impiegato} \triangleright \triangleleft_{\text{Capo}=\text{CodiceImp}} \text{ Impiegato})$

Calcoliamo innanzitutto il numero di blocchi occupati dalla relazione:

$B(\text{Impiegato})=T(\text{Impiegato}) \cdot S(\text{Impiegato})/B = 200000 \cdot 1000/400=500000$

$\text{Impiegato} \triangleright \triangleleft_{\text{Capo}=\text{CodiceImp}} (\sigma_{\text{Salario} > 2000} \text{ Impiegato})$

$\sigma_{\text{Salario} > 2000} (\text{Impiegato} \triangleright \triangleleft_{\text{Capo}=\text{CodiceImp}} \text{ Impiegato})$

Prima sequenza: $\text{Impiegato} \triangleright \triangleleft_{\text{Capo}=\text{CodiceImp}} (\sigma_{\text{Salario} > 2000} \text{ Impiegato})$

Sia $X = (\sigma_{\text{Salario} > 2000} \text{ Impiegato})$

C'è un indice primario su Salario

$F=1/3$

$\text{Costo}(X) = 3 + \lceil f \cdot B(\text{Impiegato}) \rceil = 3 + \lceil 0.333 \cdot 500000 \rceil = 166503$

$T(X) = 200000 \cdot 0.333 = 66600$

$B(X) = 66600 \cdot 1000/400 = 166500$

Né X né Impiegato stanno in memoria centrale, quindi si prova con il join nested-loop basato sui blocchi, con l'hash join ibrido e con il join con indice visto che c'è un indice secondario su Impiegato.CodiceImp.

$$\text{CostoJNL}(X \triangleright \triangleleft \text{Impiegato}) = B(X) + B(X) * B(\text{Impiegato}) / M \\ = 166500 + 166500 * 500000 / 120000 = 860250$$

Hash join ibrido: si può fare perché $M > \sqrt{B(X)} \Rightarrow 120000 > \sqrt{166500} \Rightarrow 120000 > 408$

$$\text{CostoHJI}(X \triangleright \triangleleft \text{Impiegato}) = (3 - 2M/B(X))(B(X) + B(\text{Impiegato})) = (3 - 2 * 120000 / 166500) * (166500 + 500000) = 1038779$$

$$\text{CostoIJ}(X \triangleright \triangleleft \text{Impiegato}) = B(X) + T(X) \lceil B(\text{Impiegato}) / V(\text{Impiegato}, \text{CodiceImp}) \rceil = \\ = 166500 + 66600 * 1 = 233100$$

$$\text{Costo totale} = 166503 + 233100 = 399603$$

Seconda sequenza: $\sigma_{\text{Salario} > 2000} (\text{Impiegato} \triangleright \triangleleft_{\text{Capo=CodiceImp}} \text{Impiegato})$

Sia $Z = \text{Impiegato} \triangleright \triangleleft_{\text{Capo=CodiceImp}} \text{Impiegato}$

Impiegato non sta in memoria centrale, quindi si prova con il join nested-loop basato sui blocchi, con l'hash join ibrido, con il join con indice usando l'indice secondario su Impiegato.CodiceImp e con join con indice usando l'indice secondario su Impiegato.Capo

$$\text{CostoJNL}(Z) = B(\text{Impiegato}) + B(\text{Impiegato}) * B(\text{Impiegato}) / M = 500000 + 500000 * 500000 / 120000 = 2583333$$

Hash join ibrido: si può fare perché $M > \sqrt{B(\text{Impiegato})} \Rightarrow 120000 > \sqrt{500000} \Rightarrow 120000 > 707$

$$\text{CostoHJI}(Z) = (3 - 2M/B(\text{Impiegato}))(B(\text{Impiegato}) + B(\text{Impiegato})) = (3 - 2 * 120000 / 500000) * (500000 + 500000) = 2520000$$

$$\text{CostoIJ1}(Z) = B(\text{Impiegato}) + T(\text{Impiegato})T(\text{Impiegato}) / V(\text{Impiegato}, \text{CodiceImp}) = \\ = 500000 + 200000 * 1 = 700000$$

$$\text{CostoIJ2}(Z) = B(\text{Impiegato}) + T(\text{Impiegato})T(\text{Impiegato}) / V(\text{Impiegato}, \text{Capo}) = \\ = 500000 + 200000 * 200000 / 100 = 400500000$$

$$T(Z) = T(\text{Impiegato}) * T(\text{Impiegato}) / \max\{V(\text{Impiegato}, \text{CodiceImp}), V(\text{Impiegato}, \text{Capo})\} = \\ = 200000 * 200000 / 200000 = 200000$$

$$S(Z) = S(\text{Impiegato}) + S(\text{Impiegato}) - S(\text{CodiceImp}) = 1000 + 1000 - 100 = 1900$$

$$B(Z) = 200000 * 1900 / 400 = 950000$$

Dato che non ci sono indici su Z:

$$\text{Costo: } (\sigma_{\text{Salario} > 2000} Z) = B(Z) = 950000$$

$$\text{Costo totale} = 700000 + 950000 = 1650000$$

