

COMPITO DI SISTEMI INFORMATIVI

10 dicembre 2004 (Tot. 16) Tempo: 2h

Esercizio 1 (punti 4)

Si consideri il seguente log:

- | | |
|-------------------|---------------------|
| 1. B(T1) | 13. C(T1) |
| 2. U(T1,O1,B1,A1) | 14. U(T4,O5,B6,A4) |
| 3. D(T1,O2,B2) | 15. B(T5) |
| 4. B(T2) | 16. D(T3,O5,B7,A5) |
| 5. I(T2,O3,A2) | 17. U(T5,O3,B8,A6) |
| 6. D(T2,O1,B3) | 18. C(T3) |
| 7. B(T3) | 19. U(T5,O3,B9,A7) |
| 8. U(T3,O4,B4,A3) | 20. D(T4,O3,B10) |
| 9. C(T2) | 21. I(T5,O6,A8) |
| 10. B(T4) | 22. U(T4,O7,B11,A9) |
| 11. CK(T1,T3,T4) | |
| 12. D(T1,O4,B5) | |

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

Esercizio 2 (punti 4)

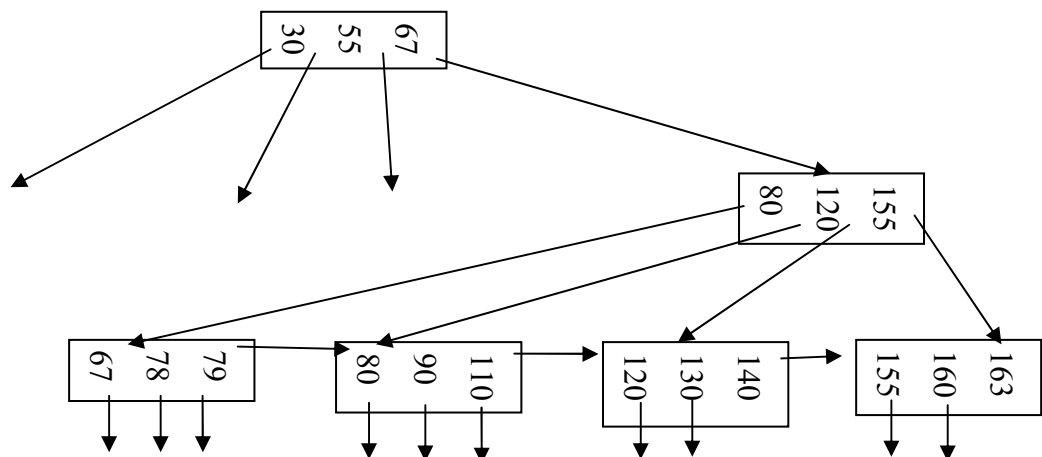
Dato il seguente schedule:

$r_1(x)$ $r_1(z)$ $r_2(z)$ $w_3(y)$ $w_4(z)$ $w_2(x)$ $r_3(z)$ $r_4(x)$ $r_1(y)$

si costruisca il grafo dei conflitti e si dica se e' conflict serializzabile.

Esercizio 3 (punti 4)

Sia dato il seguente B+ tree di ordine 3.



Si mostri come si modifica l'albero nel caso di inserimento della chiave 125

Esercizio 4 (punti 4)

Si scriva in ODL la definizione di un database per una biblioteca. Il database deve memorizzare informazioni sui lettori registrati nella biblioteca, sui libri, sugli autori dei libri e sui prestiti. Per i lettori il database deve memorizzare nome, cognome, indirizzo e anno di nascita. Per i libri il database deve memorizzare titolo, editore, anno di pubblicazione, tipo (a dorso rigido o morbido). Per gli autori, il database deve memorizzare il nome, il cognome, l'anno di nascita e la nazione di nascita. Per i prestiti il database deve memorizzare la data di inizio prestito e la data di fine prestito (modellare le date usando le stringhe). Il database deve memorizzare le relazioni che mettono in corrispondenza gli autori con i libri da loro scritti (anche piu' di un libro per autore), i libri con gli autori (anche piu' di un autore per libro), i lettori con i prestiti (anche piu' di un prestito per lettore), i prestiti con i lettori (un solo lettore per prestito), i prestiti con i libri (un libro per prestito) e i libri con i prestiti (un prestito per libro).

SOLUZIONE

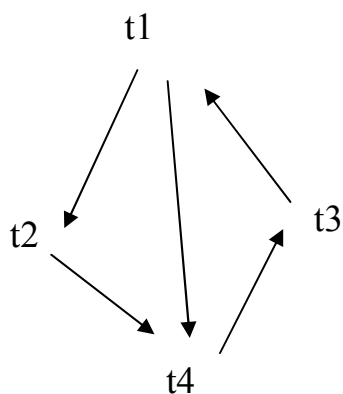
Esercizio 1

1. B(T1)
2. U(T1,O1,B1,A1)
3. D(T1,O2,B2)
4. B(T2)
5. I(T2,O3,A2)
6. D(T2,O1,B3)
7. B(T3)
8. U(T3,O4,B4,A3)
9. C(T2)
10. B(T4)
11. CK(T1,T3,T4)
12. D(T1,O4,B5)
13. C(T1)
14. U(T4,O5,B6,A4)
15. B(T5)
16. D(T3,O5,B7,A5)
17. U(T5,O3,B8,A6)
18. C(T3)
19. U(T5,O3,B9,A7)
20. D(T4,O3,B10)
21. I(T5,O6,A8)
22. U(T4,O7,B11,A9)

11 UNDO={T1,T3,T4} REDO={}
13 UNDO={T3,T4} REDO={T1}
15 UNDO={T3,T4,T5} REDO={T1}
18 UNDO={T4,T5} REDO={T1,T3}
UNDO
22 O7=B11
21 D(O6)
20 I(O3,B10)
19 O3=B9
17 O3=B8
14 O5=B6
REDO
2 O1=A1
3 D(O2)
8 O4=A3
12 D(O4)
16 D(O5)

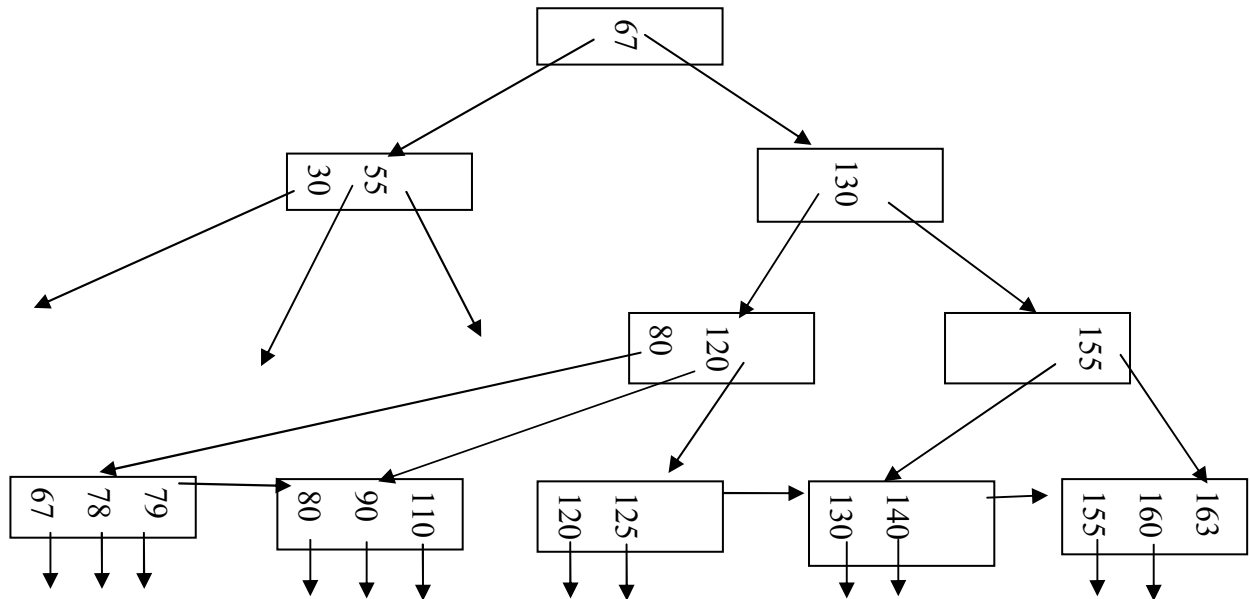
Esercizio 2

r1(x) r1(z) r2(z) w3(y) w4(z) w2(x) r3(z) r4(x) r1(y)



Esercizio 3

Inserimento di 125



Esercizio 4

class lettore (extent lettori)

```
{
    attribute string nome;
    attribute string cognome;
    attribute string indirizzo;
    attribute integer annoDiNascita;
    relationship Set<prestito> PrestitiInCorso inverse prestito::Lettore;
}
```

class libro (extent libri)

```
{
    attribute string titolo;
    attribute string editore;
    attribute integer annoDiPubblicazione;
    attribute enum tipoLibro {dorsoRigido, dorsoMorbido} tipo;
    relationship Set<autore> Autori inverse autore::Libri;
    relationship prestito Prestito inverse prestito::Libro;
}
```

class autore (extent autori)

```
{
    attribute string nome;
    attribute string cognome;
    attribute integer annoDiNascita;
    attribute string nazioneDiNascita;
    relationship Set<libro> Libri inverse libro::Autori;
}
```

```
class prestito (extent prestiti)
{
    attribute string dataDiInizio;
    attribute string dataDiFine;
    relationship lettore Lettore inverse lettore::PrestitiInCorso;
    relationship libro Libro inverse libro::Prestito;
}
```