

## COMPITO DI SISTEMI INFORMATIVI

19 luglio 2004 (Tot. 16) Tempo: 2h

### Esercizio 1 (punti 4)

Si consideri il seguente log:

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

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:

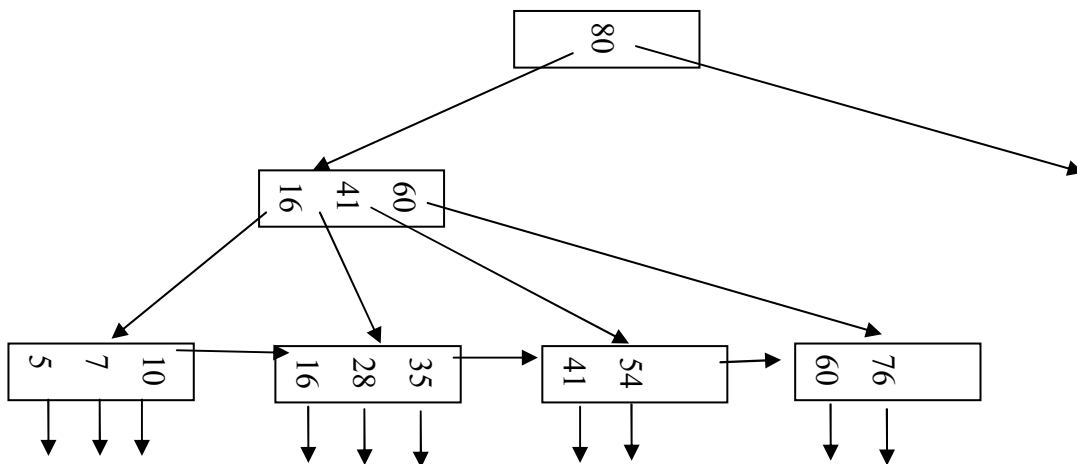
w1(x) w2(z) r1(y) w3(y) r2(x) r3(z) w4(z) r4(y)

si indichi se è view-equivalente allo schedule seriale

w1(x) r1(y) w2(z) r2(x) w3(y) r3(z) w4(z) r4(y)

### 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 30

**Esercizio 4 (punti 4)**

Si scriva in SQL-99 (modello object-relational) la definizione di un database per un circolo velico. Il database deve memorizzare informazioni sui soci del circolo, sulle barche e sulle regate. Per i soci del circolo il database deve memorizzare nome, cognome, indirizzo e anno di nascita. Per le barche il database deve memorizzare nome, stazza, lunghezza, tipo (a vela o a motore) e il socio del circolo che le possiede. Per le regate, il database deve memorizzare il nome, la data, il numero di barche partecipanti e la barca vincitrice.

## 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. I(T1,O4,A3)
7. CK(T1,T2)
8. B(T3)
9. D(T3,O5,B3)
10. U(T2,O3,B4,A4)
11. C(T3)
12. U(T1,O4,B5,A5)
13. C(T1)
14. B(T4)
15. I(T4,O6,A6)
16. D(T2,O6,B6)
17. D(T4,O4,B7)
18. U(T2,O7,B8,A7)

- 7 UNDO={T1,T2} REDO={}
  - 8 UNDO={T1,T2,T3} REDO={}
  - 11 UNDO={T1,T2} REDO={T3}
  - 13 UNDO={T2} REDO={T1,T3}
  - 14 UNDO={T2,T4} REDO={T1,T3}
- UNDO
- 18 O7=B8
  - 17 I(O4,B7)
  - 16 I(O6,B6)
  - 15 D(O6)
  - 10 O3=B4
  - 5 D(O3)
- REDO
- 2 O1=A1
  - 3 D(O2)
  - 6 I(O4,A3)
  - 9 D(O5)
  - 12 O4=A5

### Esercizio 2

w1(x) w2(z) r1(y) w3(y) r2(x) r3(z) w4(z) r4(y)

"legge-da"={(t2,t1),(t3,t2),(t4,t3)}  
scritture finali={(t1,x),(t3,y),(t4,z)}

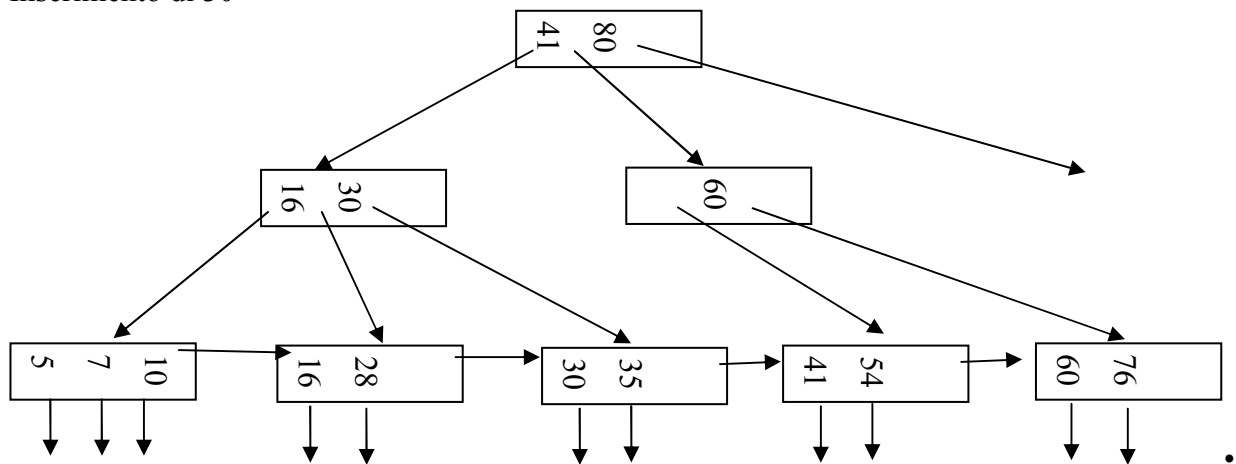
w1(x) r1(y) w2(z) r2(x) w3(y) r3(z) w4(z) r4(y)

"legge-da"={(t2,t1),(t3,t2),(t4,t3)}  
scritture finali={(t1,x),(t3,y),(t4,z)}

I due schedule sono view-equivalenti e quindi il primo schedule è view-serializzabile

### Esercizio 3

Inserimento di 30



### Esercizio 4

```
create type tipoSocio as (  
    nome CHAR(20),  
    cognome CHAR(20),  
    indirizzo CHAR(20),  
    annoDiNascita INT);
```

```
create type tipoBarca as (  
    nome CHAR(20),  
    stazza INT,  
    lunghezza INT,  
    tipoBarca CHAR(20),  
    proprietario REF tipoSocio);
```

```
create type tipoRegata as (  
    nome CHAR(20),  
    data DATE,  
    numPart INT,  
    vincitore REF tipoBarca);
```

```
create table socio as tipoSocio;  
create table barca as tipoBarca;  
create table regata as tipoRegata;
```