COMPITO DI APPLICAZIONI DI SISTEMI INFORMATIVI

15 Giugno 2004 (Tot. 16) Tempo: 2h

Esercizio 1 (punti 4)

Si consideri il seguente log:

1. B(T1)
2. D(T1,O1,B1)
3. B(T2)
4. U(T2,O2,B2,A1)
5. I(T1,O3,A2)
6. B(T3)
7. D(T3,O2,B3)
8. I(T2,O4,A3)
9. C(T2)
10. CK(T1,T3)
11. U(T1,O5,B4,A4)
12. U(T3,O6,B5,A5)
13. B(T4)
14. C(T1)
15. I(T4,O7,A6)
16. U(T3,O8,B6,A7)
17. D(T4,O9,B7)
18. I(T3,O10,A8)

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(x) r1(y) w3(y) r2(x) r3(z) w4(z) r4(y)
si costruisca il grafo dei conflitti e si dica se e' conflict serializzabile.

Esercizio 3 (punti 4)

•Date le relazioni
Giocatore(CodGio,CodSqua,Nome,Cognome,Data_di_nascita,Stipendio)
Squadra(CodSqua,Nome,Citta', N_scudetti)
•Si calcoli il costo di esecuzione della query

```
SELECT G.Nome, G.Cognome, S.Nome
FROM Giocatore AS G, Squadra AS S
WHERE G.CodSqua=S.CodSqua
```

•supponendo che vengano utilizzati gli algoritmi di Block-Based Nested-Loops Join, Sort Join, Hash Join Ibrido e Join con Indice.
•Si supponga di avere M=150 buffer di memoria centrale disponibili, che B(Giocatore)=5000, T(Giocatore)=15.000, B(Squadra)=500, T(Squadra)=2000 e si supponga di avere un indice primario su Squadra.CodSqua.

Esercizio 4 (punti 4)

Si scriva in ODL la definizione di un database per un circolo velico. Il database deve memorizzare informazioni sui membri del circolo, sulle barche e sulle regate. Per i membri 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). Per le regate, il database deve memorizzare il nome, la data (usare una stringa) e il numero di barche partecipanti. Il database deve memorizzare le relazioni che mettono in corrispondenza le barche con i relativi proprietari (un socio per barca), i soci con le barche possedute (anche piu' di una per un socio), le barche con le regate da esse vinte e le regate con la barca vincitrice.

SOLUZIONE

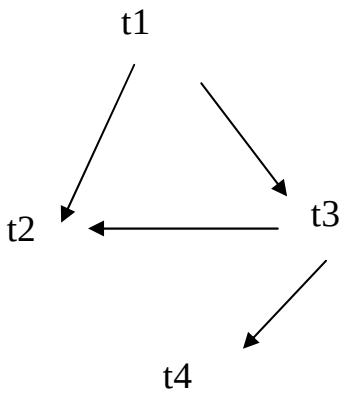
Esercizio 1

1. B(T1)
2. D(T1,O1,B1)
3. B(T2)
4. U(T2,O2,B2,A1)
5. I(T1,O3,A2)
6. B(T3)
7. D(T3,O2,B3)
8. I(T2,O4,A3)
9. C(T2)
10. CK(T1,T3)
11. U(T1,O5,B4,A4)
12. U(T3,O6,B5,A5)
13. B(T4)
14. C(T1)
15. I(T4,O7,A6)
16. U(T3,O8,B6,A7)
17. D(T4,O9,B7)
18. I(T3,O10,A8)

10 UNDO={T1,T3} REDO={}
13 UNDO={T1,T3,T4} REDO={}
14 UNDO={T3,T4} REDO={T1}
UNDO
18 D(O10)
17 I(O9,B7)
16 O8=B6
15 D(O7)
12 O6=B5
7 I(O2,B3)
REDO
2 D(O1)
5 I(O3,A2)
11 O5=A4

Esercizio 2

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



Esercizio 3

M=150
B(Giocatore)=5000, T(Giocatore)=15.000, B(Squadra)=500, T(Squadra)=2000
Indice primario su Squadra.CodSqua.

•Join Block-Based Nested-Loops:
–Costo= B(Squadra)+B(Squadra)B(Giocatore)/(M-1) = 500 +500 * 5000/150=17.166
•Sort Join:
–puo' essere fatto se B(Squadra)+B(Giocatore)<=M^2 B(Squadra)+B(Giocatore)<=22.500
5500<=22.500 ok
–Costo=3(B(Squadra)+B(Giocatore))=3(5000+500)=16.500
•Hash Join Ibrido:
–puo' essere fatto se B(Squadra)<=M^2
500<=22.500 ok
–Costo=(3-2M/B(Squadra))(B(Giocatore)+B(Squadra))=
(3 –2 * 150/500)(5000+500)=(3-0.6)(5500)=13.200
•Index Join:
–Costo=B(Giocatore)+T(Giocatore)*1=5000+15.000*1=20.000
•

Esercizio 4

```
class socio (extent soci)
{
        attribute string nome;
        attribute string cognome;
        attribute string indirizzo;
        attribute integer annoDiNascita;
        relationship Set<barca> possiede inverse barca::proprietario;
}
class barca (extent barche)
{
        attribute string nome;
        attribute integer stazza;
        attribute integer lunghezza;
        attribute enum tipoBarca {aVela, aMotore} tipo;
        relationship socio proprietario inverse socio::possiede;
        relationship Set<regata> regateVinte inverse regata::vincitore;
}
class regata (extent regate}
{
        attribute string nome;
        attribute string data;
        attribute integer numPart;
        relationship barca vincitore inverse barca::regateVinte;
}
```