

Basi di dati, primo modulo

28 giugno 2002

Tempo a disposizione: un'ora e trenta minuti. Libri chiusi.

Domanda 1 (25%)

Considerare i seguenti schedule:

1. $r_1(x)w_1(x)r_2(x)w_2(x)r_0(y)w_1(y)$
2. $w_0(z)r_1(z)r_2(z)w_2(z)w_2(y)$
3. $r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$
4. $r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$
5. $r_2(x)w_2(x)r_1(x)w_1(x)$

Specificare, con una breve giustificazione, a quali delle seguenti classi ciascuno di essi appartiene: S (seriale), VSR (view-serializzabile), CSR (conflict-serializzabile), 2PL (generabile da uno scheduler basato sul lock a due fasi) e TS (generabile da uno scheduler che utilizzi il metodo dei timestamp; si assuma che l'ordinamento degli identificatori delle transazioni corrisponda a quello dei timestamp).

Domanda 2 (25%)

Spiegare perché, nel protocollo di commit a due fasi, l'intervallo che intercorre fra l'istante in cui un partecipante dichiara il proprio stato di "ready" e quello in cui riceve l'informazione sulla decisione globale viene chiamato "finestra di incertezza" e perché è importante che esso sia il più breve possibile.

Domanda 3 (25%) Considerare la seguente interrogazione in SQL:

```
SELECT DISTINCT C, L
FROM R1, R2, R3
WHERE R1.C = R2.D AND R2.F=R3.G AND R1.B=R3.L AND R3.H>10
```

Mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni; prestare attenzione anche alla DISTINCT, in quanto le realizzazioni degli operatori non producono necessariamente insiemi, ma liste di ennuple), giustificando brevemente le scelte più significative, con riferimento alle seguenti informazioni sulla base di dati:

- la relazione $R_1(\underline{ABC})$ ha 100.000 ennuple, una struttura heap e un indice secondario su C ;
- la relazione $R_2(\underline{DEF})$ ha 30.000 ennuple, una struttura heap e un indice secondario sulla chiave D ;
- la relazione $R_3(\underline{GHL})$ ha 10.000 ennuple, una struttura heap e un indice secondario sulla chiave G .

Domanda 4 (25%)

Molti DBMS hanno introdotto recentemente indici basati su una tecnica innovativa, detta "bitmap." Un indice bitmap, su un attributo A con K valori diversi, per una relazione con N ennuple, è costituito da K vettori (uno per ciascun valore dell'attributo A) di N bit ciascuno (un bit per ogni ennupla della relazione): l' i -esimo bit del vettore associato al valore a_j è 1 se il valore della i -esima ennupla sull'attributo A è a_j e 0 se il valore è diverso. L'accesso ai vettori è organizzato per mezzo di un albero. In sostanza, rispetto ad un B^+ -tree, abbiamo una struttura di accesso simile, ma con foglie diverse: nel B^+ -tree, per ogni valore dell'attributo abbiamo una lista di indirizzi, mentre nell'indice bitmap abbiamo un vettore di bit (deve poi esistere una tabella che associ gli indirizzi delle ennuple ai numeri da 1 a N). Sulla base degli elementi forniti:

- valutare lo spazio necessario per le foglie di un indice bitmap, confrontandolo con quello necessario per le foglie di un B^+ -tree; in particolare, commentare come varia tale confronto in funzione del valore di K ;
- spiegare come possono essere realizzate selezioni congiuntive su più attributi su ciascuno dei quali sia definito un indice bitmap;
- spiegare per quali ragioni un indice bitmap può risultare vantaggioso nel contesto dei data warehouse.