

# Basi di dati, primo modulo

20 settembre 2002

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

## Domanda 1 (25%)

Considerare i seguenti schedule:

1.  $r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$
2.  $r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$
3.  $r_2(x)w_2(x)r_3(x)w_3(x)r_1(y)w_2(y)$
4.  $w_1(z)r_2(z)r_3(z)w_3(z)w_3(y)$
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 (35%)

Si consideri una relazione IMPIEGATO(Matricola,Cognome,Nome,DataNascita) con un numero di ennuple pari a  $N$  abbastanza stabile nel tempo e una dimensione di ciascuna ennupla (a lunghezza fissa) pari a  $L$  byte, di cui  $K$  per la chiave.

Supporre di avere a disposizione un DBMS che permetta strutture fisiche disordinate (heap) e hash e che preveda la possibilità di definire indici secondari e operi su un sistema operativo che utilizza blocchi di dimensione  $B$  e con puntatori ai blocchi di  $P$  caratteri.

Siano le seguenti le operazioni principali:

1. ricerca esatta sul numero di matricola con frequenza  $f_1$
2. ricerca sul cognome (anche su sottostringa iniziale) con frequenza  $f_2$ ; mediamente una richiesta restituisce 4 record.

Individuare alcune (almeno una) possibili organizzazione fisica per tale relazione e calcolare (approssimativamente) il numero di accessi a memoria secondaria (nell'unità di tempo) supponendo  $N = 5.000.000$ ,  $L = 125$ ,  $K = 5$ ,  $B = 1.000$ ,  $P = 4$ ,  $f_1 = 100$ ,  $f_2 = 1.000$ .

## Domanda 3 (25%)

Si consideri una base di dati relativa ad un sistema di prenotazione (per viaggi aerei), con le relazioni:

- *Prenotazioni*(Numero, Volo, Data, NumeroPosti)
- *Disponibilità*(Volo, Data, PostiTotali, PostiDisponibili)

in cui il valore dell'attributo *PostiDisponibili* è, per ciascun volo (e giorno), pari alla differenza fra *PostiTotali* e la somma dei posti prenotati su tale volo. Sulla relazione *Prenotazioni* sono effettuati inserimenti ed eliminazioni, ma non modifiche.

Descrivere (senza preoccuparsi dei dettagli sintattici) una o più regole attive che permettano di mantenere corretto il valore di *PostiDisponibili* a fronte di inserimenti ed eliminazioni nella relazione *Prenotazioni*.

## Domanda 4 (15%)

Illustrare brevemente la tecnica chiamata "predicate lock" e spiegare perché è importante e come può essere realizzata.