

Basi di dati, primo modulo — 18 novembre 2003

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

Domanda 1 (35%) Illustrare i principali algoritmi per l'esecuzione dell'operatore di join, indicando anche le strutture fisiche necessarie e le prestazioni, facendo anche riferimento, se possibile, al ruolo dei buffer.

Domanda 2 (25%)

Considerare i seguenti schedule:

1. $r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$
2. $r_1(y)r_4(z)r_4(y)w_4(y)w_4(z)r_1(z)r_2(z)r_3(z)w_3(z)$
3. $r_4(z)r_4(y)w_4(y)w_4(z)r_2(x)w_2(x)r_1(x)w_1(x)$
4. $w_1(z)r_2(z)r_3(z)w_3(z)w_3(y)$
5. $r_2(x)w_2(x)r_3(x)w_3(x)r_1(y)w_2(y)$

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 3 (20%)

Il controllo della concorrenza deve tener conto anche delle pagine degli indici.

1. Spiegare perché il 2PL, certamente corretto, risulta estremamente inefficiente se applicato ai B-tree.
2. Considerare il seguente protocollo, chiamato *tree protocol* (per semplicità riferirsi solo a lock esclusivi):
 - su ciascun nodo, a parte la radice, può ottenere un lock solo una transazione che abbia il lock sul nodo genitore
 - non è possibile per una transazione ottenere due volte un lock sullo stesso nodo

Si noti che è possibile rilasciare il lock sul genitore dopo aver ottenuto quello sul figlio e che non è imposta la condizione di 2PL.

Spiegare, intuitivamente, perché il tree protocol garantisce la serializzabilità e perché esso risulta più efficiente di un protocollo 2PL.

Domanda 4 (20%) Illustrare brevemente il concetto di middleware e un argomento correlato che risulti di interesse, ad esempio fra quelli trattati nel corso integrativo.