

Gestione dei dati – appello del 20/6/2012

COGNOME:
NOME:
MATRICOLA:

Autorizzo la pubblicazione del mio voto di questo esame sul sito web http://www.dis.uniroma1.it/~rosati/gd , secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede,

Esercizio 1 Dato il seguente schedule $S = r_1(A) r_2(B) r_3(C) r_4(D) r_5(E) w_4(E) w_3(D) w_2(C) w_1(B) w_5(A)$

1. dire se S è view-serializzabile, motivando la risposta;
2. dire se S è eseguibile da uno scheduler che segue il protocollo 2PL con lock esclusivi e condivisi. In caso positivo, completare lo schedule S con le opportune istruzioni di lock e unlock. In caso negativo, motivare la risposta;
3. dire se S è conflict-serializzabile, motivando la risposta. In caso positivo, scrivere uno schedule seriale equivalente a S ;
4. dire se è possibile inserire in S le istruzioni di commit delle transazioni T_1, \dots, T_5 in modo tale che lo schedule risultante sia ACR e, in caso positivo, scrivere tale schedule.

Esercizio 2 Si consideri una base di dati contenente la relazione **Impiegato** con attributi **matricola**, **cognome**, **nome**, **citta**, **annoDiAssunzione**, **stipendio**, **dataDiNascita**. La relazione **Impiegato** contiene 10^5 record, e ogni record di tale relazione ha dimensione $N/100$, dove N è la dimensione di una pagina di memoria. Inoltre **matricola** è chiave per la relazione **Impiegato**. La relazione **Impiegato** non è soggetta ad aggiornamenti.

1. Si considerino query del tipo: “selezionare la data di nascita di tutti gli impiegati assunti in un certo anno a , abitanti nella città c e la cui matricola è pari a m ”. Scegliere l’organizzazione fisica per la relazione **Impiegato** che minimizza il costo dell’esecuzione di tali query;
2. assumendo un tempo medio di accesso ad una pagina in memoria di massa di 5 millisecondi, e trascurando i tempi di elaborazione in memoria centrale, calcolare il tempo (espresso in millisecondi) necessario all’esecuzione delle query di cui al punto 1 in base all’organizzazione fisica per la relazione **Impiegato** scelta al punto 1;
3. si considerino ora query del tipo: “selezionare la città e la data di nascita di tutti gli impiegati aventi cognome c , nome n e matricola compresa tra m_1 e m_2 ”. Scegliere l’organizzazione fisica per la relazione **Impiegato** che minimizza il costo dell’esecuzione di tali query;
4. assumendo un tempo medio di accesso ad una pagina in memoria di massa di 5 millisecondi, e trascurando i tempi di elaborazione in memoria centrale, calcolare il tempo (espresso in millisecondi) necessario all’esecuzione delle query di cui al punto 3 in base all’organizzazione fisica per la relazione **Impiegato** scelta al punto 3.

Esercizio 3 Data la seguente DTD:

```
<!DOCTYPE x [  
  <!ELEMENT x (s+)>  
  <!ELEMENT s (r,p,t,v)>  
  <!ELEMENT r (y*,v,(r|s)*)>  
  <!ELEMENT p (z)>  
  <!ELEMENT t (y|s)*>  
  <!ELEMENT v (#PCDATA)>  
  <!ELEMENT y (y|s)*>  
  <!ELEMENT z (#PCDATA)>  
  <!ATTLIST p val CDATA #REQUIRED>  
  <!ATTLIST s attr CDATA #REQUIRED>  
>
```

1. Scrivere lo schema relazionale ottenuto applicando l’algoritmo schema-driven di XML shredding alla DTD precedente.
2. Scrivere un documento XML valido rispetto alla DTD precedente e scrivere l’istanza della base di dati ottenuta traducendo tale documento nello schema relazionale ottenuto al punto 1.

Esercizio 4 Si consideri una base di dati contenente la relazione **Esame** con attributi **codStudente**, **codCorso**, **voto**, **data**, **numCrediti** e la relazione **Studente** con attributi **codStudente**, **nome**, **cognome**, **annoImmatricolazione**, **indirizzo**. La relazione **Esame** contiene 200.000 record, e ogni record di tale relazione ha dimensione $N/40$, dove N la dimensione di una pagina di memoria. Su tale relazione è dichiarato un indice B⁺-tree clusterizzato (che usa l’alternativa 2) con chiave di ricerca **data**. La relazione **Studente** contiene 20.000 record, e ogni record di tale relazione ha dimensione $N/100$. Inoltre, **codStudente** è chiave per la relazione **Studente**, e su tale relazione è dichiarato un indice hash statico con chiave di ricerca **annoImmatricolazione**.

Si consideri la seguente query Q:

```
SELECT S.cognome, S.nome, E.voto FROM Esame E, Studente S  
WHERE E.codStudente = S.codStudente AND S.annoImmatricolazione = "2007" AND E.data = "20/06/2012"
```

Si assuma che: (i) il sistema possa eseguire i join mediante l’algoritmo Block Nested Loop; (ii) il buffer abbia a disposizione 22 slot liberi per eseguire la query; (iii) la relazione **Esame** abbia 2000 record con valore dell’attributo **data** pari a "20/06/2012"; (iv) la relazione **Studente** abbia 4000 record con valore dell’attributo **annoImmatricolazione** pari a "2007". Si richiede di scrivere il query plan per tale query che minimizza il numero di accessi alla memoria di massa, e di calcolare il costo dell’esecuzione di tale query plan, espresso in numero di accessi alla memoria di massa. Si ricorda che i query plan da considerare devono necessariamente prevedere relazioni primitive come relazioni interne delle operazioni di join.