

Gestione dei dati – appello del 31/1/2012

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 $S = w_1(A) r_1(X) r_2(Z) r_2(Y) r_4(B) w_3(X) w_3(B) r_5(A) w_2(B) r_4(Z) r_1(Y)$

1. dire se S è conflict-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 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;
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 recuperabile e, in caso positivo, scrivere tale schedule.

Esercizio 2 Si consideri una base di dati contenente la relazione **Residente** con attributi **codiceFiscale**, **cognome**, **nome**, **annoDiNascita**, **comuneDiResidenza**, **annoDiInizioResidenza**. La relazione **Residente** contiene 10^5 record, e ogni record di tale relazione ha dimensione $N/50$, dove N è la dimensione di una pagina di memoria. Inoltre **codiceFiscale** è chiave per la relazione **Residente**. La relazione **Residente** è soggetta ad aggiornamenti.

1. Si considerino query del tipo: “selezionare nome e cognome di tutte le persone residenti in un certo comune c ed aventi anno di nascita maggiore o uguale ad un certo anno a ”. Scegliere l’organizzazione fisica per la relazione **Residente** 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 **Residente** scelta al punto 1;
3. si considerino ora query del tipo: “selezionare il codice fiscale di tutte le persone nate dopo un certo anno a , e aventi anno di inizio residenza minore di un certo anno r ”. Scegliere l’organizzazione fisica per la relazione **Residente** 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 **Residente** scelta al punto 3.

Esercizio 3 Data la seguente DTD:

```
<!DOCTYPE r [  
  <!ELEMENT r (b, a*, c, d*, (e|f)+) >  
  <!ELEMENT a (a*, ((e,c)*|d)) >  
  <!ELEMENT b (f|d)*>  
  <!ELEMENT c (f+, (a|b), c*)>  
  <!ELEMENT d (e,a*)>  
  <!ELEMENT e (#PCDATA)>  
  <!ELEMENT f EMPTY>  
  <!ATTLIST a x CDATA #REQUIRED>  
  <!ATTLIST b y CDATA #REQUIRED>  
  <!ATTLIST c z 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 **studente**, **corso**, **voto** e la relazione **Residenza** con attributi **studente**, **indirizzo**. La relazione **Esame** contiene 100.000 record, e ogni record di tale relazione ha dimensione $N/25$, dove N la dimensione di una pagina di memoria. Tale relazione è memorizzata in un heap file. La relazione **Residenza** contiene 5.000 record, e ogni record di tale relazione ha dimensione $N/5$. Inoltre, **studente** è chiave per la relazione **Residenza**, e su tale relazione è dichiarato un indice B⁺-tree clusterizzato (che usa l’alternativa 1) con chiave di ricerca **studente**.

Si consideri la seguente query Q:

```
select E.studente, E.voto, R.indirizzo from Esame E, Residenza R  
where E.studente = R.studente and E.corso = "GD"
```

Si assuma che: (i) il sistema possa eseguire i join sia mediante l’algoritmo Block Nested Loop che mediante l’algoritmo Sort Merge Join; (ii) il buffer abbia a disposizione 7 slot liberi per eseguire la query; (iii) la relazione **Esame** abbia 200 record con valore dell’attributo **corso** pari a “GD”. 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.