

# Gestione dei dati – appello del 17/9/2010

COGNOME: .....
NOME: .....
MATRICOLA: .....

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

**Esercizio 1** Si consideri una base di dati contenente la relazione **Residente** con attributi **codiceFiscale**, **cognome**, **nome**, **comuneDiNascita**, **comuneDiResidenza**, **annoDiNascita**. La relazione **Residente** contiene  $10^7$  record, e ogni record di tale relazione ha dimensione  $N/500$ , dove  $N$  è la dimensione di una pagina di memoria. Inoltre **codiceFiscale** è chiave per la relazione **Residente**. La relazione **Residente** non è soggetta ad aggiornamenti.

1. Si considerino query del tipo: “selezionare cognome, nome e codice fiscale di tutti i residenti in un certo comune  $n$  nati in un certo anno  $a$ ”. Scegliere l’organizzazione fisica per la relazione **Residente** che minimizza il costo dell’esecuzione di tali query. Si assuma che per ogni nome di comune  $n$  e ogni anno  $a$  siano presenti in media 50 record il cui valore di **comuneDiResidenza** è uguale a  $n$  e il cui valore di **annoDiNascita** è uguale a  $a$ ;
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 secondi) 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 cognome, nome e codice fiscale di tutti i residenti in un certo comune  $n$  nati dopo il 1999”. Scegliere l’organizzazione fisica per la relazione **Residente** che minimizza il costo dell’esecuzione di tali query. Si assuma che per ogni nome di comune  $n$  siano presenti in media 500 record il cui valore di **comuneDiResidenza** è uguale a  $n$  e il cui valore di **annoDiNascita** è maggiore di 1999;
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 necessario all’esecuzione delle query di cui al punto 3 in base all’organizzazione fisica per la relazione **Residente** scelta al punto 3.

**Esercizio 2** Si consideri una base di dati contenente la relazione **Studente** con attributi **cognome**, **nome**, **matricola**, **dataNascita**, **nomeComune**, **nomeProvincia** e la relazione **Comune** con attributi **nomeComune**, **numeroAbitanti**, **nomeProvincia**, **superficie**.

La relazione **Studente** contiene 400000 record, e ogni record di tale relazione ha dimensione  $N/200$ , dove  $N$  la dimensione di una pagina di memoria. Si assuma inoltre che la relazione **Comune** contenga 10000 record, e che ogni record di tale relazione abbia dimensione  $N/100$ . Si supponga inoltre che **matricola** sia chiave per la relazione **Comune** e che (**nomeComune**, **nomeProvincia**) sia chiave per la relazione **Comune**. Le suddette relazioni sono soggette ad aggiornamenti. Inoltre sulla relazione **Comune** è dichiarato un indice  $B^+$ -tree con chiave di ricerca **nomeComune** e fan-out 100.

Si consideri la seguente query  $Q$ :

```
select S.cognome, S.nome, S.matricola, C.numeroAbitanti from Studente S, Comune C
where S.nomeComune = C.nomeComune
```

Assumendo che il buffer abbia a disposizione 12 slot liberi per eseguire  $Q$ , dire quale algoritmo per l’esecuzione della precedente query è preferibile tra Block Nested Loop e Index Nested Loop, calcolando i rispettivi costi di esecuzione della query.

**Esercizio 3** Dato il seguente schedule  $S = w_0(D) r_2(C) w_3(B) w_3(E) c_3 w_1(E) w_0(C) r_0(A) c_0 r_1(B) r_2(D) w_2(A) r_2(B) c_2 w_1(D) c_1$

1. dire se  $S$  è recuperabile, motivando la risposta;
2. dire se  $S$  è ACR, motivando la risposta;
3. dire se  $S$  è view-serializzabile, motivando la risposta;
4. 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.

**Esercizio 4** Data la seguente DTD:

```
<!DOCTYPE esami [
  <!ELEMENT esami (studente+) >
  <!ELEMENT studente (cognome, nome+, matricola, corso, voto) >
  <!ELEMENT cognome (#PCDATA)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT matricola (#PCDATA)>
  <!ELEMENT corso (nomeCorso, docente+, aula)>
  <!ELEMENT nomeCorso (#PCDATA)>
  <!ELEMENT docente (cognome, nome) >
  <!ELEMENT aula (#PCDATA)>
  <!ELEMENT voto (#PCDATA)>
  <!ATTLIST studente annoNascita CDATA #REQUIRED>
  <!ATTLIST docente dipartimento 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.