

## **Basi di Dati II**

### **Corso di Laurea in Informatica**

#### **Esercizio**

Si consideri una relazione AUTOMOBILI(NumeroTelaio, Modello, AnnoImmatricolazione, Valore-Commerciale) con un numero di tuple pari a  $N = 90000$  stabile nel tempo, ed una dimensione di ciascuna tupla (a lunghezza fissa) pari a  $L = 100$  byte di cui  $K = 25$  byte per la chiave.

Si supponga che la relazione sia memorizzata nel database in un file di tipo heap costituito da un numero di blocchi pari a  $B$ . Sia  $D = 2400$  byte la dimensione di ciascuno blocco ed  $f = .75$  il fattore di riempimento.

- (a) determinare il numero di blocchi  $B$  necessari a memorizzare l'intera relazione;
- (b) sarebbe possibile realizzare un file di indice ad un livello sulla chiave della relazione? Di che tipo dovrebbe essere (primario, secondario, di cluster / denso, sparso)? Assumendo che il puntatore al record abbia dimensione  $P = 5$  byte, quanti blocchi sono necessari per memorizzare l'indice?
- (c) si calcoli la profondità di un indice multilivello costruito sulla chiave della relazione e il numero totale dei blocchi che compongono tale indice multilivello, considerando che i puntatori ai blocchi e ai record abbiano dimensione  $P = 5$  byte.

#### **Esercizio**

Si consideri la relazione  $R(\text{codfisc}, \text{nome}, \text{cognome}, \text{datanasc})$  dove  $\text{codfisc}$  ha dimensione 16Byte,  $\text{nome}$  24Byte,  $\text{cognome}$  40Byte e  $\text{datanasc}$  10Byte. La relazione viene memorizzata in un file ordinato sulla chiave della relazione ( $\text{codfisc}$ ) che ha blocchi di dimensione pari a 512Byte e fattore di riempimento pari al 100%. Il numero di tuple che compongono la relazione è pari a  $N = 100000$ .

- (a) Quanti blocchi sono necessari per memorizzare l'intera relazione?

Si consideri il problema di creare un indice multilivello sull'attributo  $\text{cognome}$  e si assuma che la dimensione del puntatore ai blocchi sia di 10Byte.

- (b) Che tipo di indice è necessario utilizzare (primario/secondario, denso/sparso)? Giustificare la risposta
- (c) Quanti livelli e quanti blocchi in totale sono necessari per memorizzare l'indice?

### Esercizio

Si consideri lo schema relazionale composto dalle seguenti relazioni:

R (A, B, C, D)

S (D, E, F)

in cui PK(R) = A, PK(S) = D e FK(R → S) = D e si consideri l'interrogazione:

```
SELECT R.A, R.D, S.F
FROM R, S
WHERE R.D = S.D AND R.C = 1 AND S.E = 2;
```

- Si costruisca l'albero canonico (o iniziale) per l'esecuzione dell'interrogazione;
- si ottimizzi l'albero canonico per minimizzare i costi di esecuzione;
- quale algoritmo di esecuzione dell'operazione di JOIN è più conveniente utilizzare per questa interrogazione supponendo che le tuple di R siano  $nR = 1000$ , le tuple di S siano  $nS = 500$  e la selettività dell'operazione di selezione su C sia  $sC = 5$  e quella su E sia  $sE = 10$ ? Giustificare la risposta.

### Esercizio

Sia dato il seguente *log di sistema*:

DUMP, B(T<sub>1</sub>), B(T<sub>2</sub>), B(T<sub>3</sub>), I(T<sub>1</sub>, O<sub>1</sub>, A<sub>1</sub>), D(T<sub>2</sub>, O<sub>2</sub>, B<sub>2</sub>), B(T<sub>4</sub>), U(T<sub>4</sub>, O<sub>3</sub>, B<sub>3</sub>, A<sub>3</sub>), U(T<sub>1</sub>, O<sub>4</sub>, B<sub>4</sub>, A<sub>4</sub>), C(T<sub>2</sub>), CK(T<sub>1</sub>, T<sub>3</sub>, T<sub>4</sub>), B(T<sub>5</sub>), B(T<sub>6</sub>), U(T<sub>5</sub>, O<sub>5</sub>, B<sub>5</sub>, A<sub>5</sub>), A(T<sub>3</sub>), CK(T<sub>1</sub>, T<sub>4</sub>, T<sub>5</sub>, T<sub>6</sub>), B(T<sub>7</sub>), A(T<sub>4</sub>), U(T<sub>7</sub>, O<sub>6</sub>, B<sub>6</sub>, A<sub>6</sub>), U(T<sub>6</sub>, O<sub>3</sub>, B<sub>7</sub>, A<sub>7</sub>), B(T<sub>8</sub>), A(T<sub>7</sub>), guasto

- descrivere la successione di operazioni da eseguire per effettuare la ripresa a caldo;
- indicare la realizzazione progressiva degli insiemi di UNDO e REDO;
- esplicitare le azioni di recovery da eseguire in termini degli stati *before* ed *after*.

### Esercizio

Siano dati i seguenti *schedule*:

S1: r1(x) r2(x) w1(x) w2(x) r2(y) w2(y)

S2: r3(x) w3(x) w3(y) r1(y) w1(x) w1(y) r2(x) w2(x) w2(y)

S3: r1(x) w1(x) r3(x) r2(x) w3(x)

- Verificare esplicitamente se S1 sia conflict-serializzabile (CSR) e/o view-serializzabile (VSR). In caso affermativo indicare tutti gli schedule seriali ad esso equivalenti.
- Verificare esplicitamente se S2 sia conflict-serializzabile (CSR) e/o view-serializzabile (VSR). In caso affermativo indicare tutti gli schedule seriali ad esso equivalenti.
- Verificare esplicitamente se S3 sia conflict-serializzabile (CSR) e/o view-serializzabile (VSR). In caso affermativo indicare tutti gli schedule seriali ad esso equivalenti.
- Assumendo di utilizzare il protocollo di lock a 2 fasi strict, verificare se negli schedule S1, S2 e S3 si presentano delle situazioni di stallo (*dead-lock*), specificando le situazioni di attesa.

### Esercizio

Si considerino le relazioni *LIBRI*(*ISBN*, *Titolo*, *Autore*, *PrezzoCopertina*, ...), *LIBRERIE*(*IDlibreria*, *NomeLibreria*, *Indirizzo*, ...), *MAGAZZINO*(*IDlibreria*, *ISBN*, *Quantità*, *PrezzoVendita*...) utilizzate per la gestione delle copie dei libri in carico a varie filiali della libreria *LeggoTanto*.

Si supponga che la catena di librerie abbia 3 filiali distribuite sul territorio (identificabili con il valore dell'attributo *IDlibreria*) e di utilizzare un DBMS distribuito per fare in modo che ciascuna filiale abbia in locale i dati relativi al suo magazzino ed ai suoi libri.

- Esprimere in algebra relazionale una frammentazione opportuna (per tutte le relazioni necessarie – per il solo frammento relativo a *IDlibreria* = 3);
- esprimere in SQL la frammentazione di cui al punto precedente;
- esprimere a livello di trasparenza di allocazione l'interrogazione per ricavare per ciascuna filiale (*NomeLibreria*) il numero totale di libri in carico.

### Esercizio

Dato il seguente DTD:

```
<!ELEMENT library (book+)>
<!ELEMENT book (title,author,ISBN,year)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT year (#PCDATA)>

<!ATTLIST book
  librarycode ID #REQUIRED
  location CDATA #IMPLIED
  loantype (short|long) "long">
```

- scrivere un documento XML che sia valido rispetto al DTD;
- esprimere un metodo per indicare più di un autore di un libro, giustificando la risposta;
- commentare la definizione degli attributi dell'elemento book, specificando le possibilità ed i vincoli forniti dalla stessa.

### Esercizio

Dato il seguente documento XML

```
<?xml version="1.0"?>
<!DOCTYPE ...>
<PartsRelation>
  <NOTE>Tabella delle parti prodotto</NOTE>
  <PartTuple CITY="London">
    <PNUM>P1</PNUM>
    <PNAME>Nut</PNAME>
    <WEIGHT>12.0</WEIGHT>
    <NOTE>Part Color is Red by default</NOTE>
  </PartTuple>
  <PartTuple COLOR="Green" CITY="Paris">
    <PNUM>P2</PNUM>
    <PNAME>Bolt</PNAME>
```

```
        <WEIGHT>17.0</WEIGHT>
    </PartTuple>
    <PartTuple COLOR="Blue" CITY="Oslo">
        <PNUM>P3</PNUM>
        <PNAME>Screw</PNAME>
        <WEIGHT>17.0</WEIGHT>
    </PartTuple>
</PartsRelation>
```

- (a) Indicare quale elemento costituisce la radice del documento XML e specificare quanti e quali attributi sono presenti nel documento XML;
- (b) indicare se il documento è ben formato (well-formed), giustificando la risposta;
- (c) scrivere un opportuno DTD per il quale il documento XML sia valido, assumendo che i possibili valori per CITY siano London, Oslo o Paris, e che uno di questi debba essere presente, e che i possibili valori per COLOR siano Red, Green e Blue con Red di default.