

Sistemi informativi II
Prof. Stefano Paraboschi
II prova - 20/12/2007

A. Si ha una base di dati con il seguente schema:

PRODOTTO(Codice,Nome,Prezzo,QtaOrdinata)
ORDINE(CodOrd,CodProd,Qta,PrezzoTotale)

Realizzare tramite trigger *statement-level* un meccanismo che garantisca che il valore di *QtaOrdinata* sia aggiornato rispetto a inserimenti di tuple in *ORDINE*. Realizzare inoltre tramite trigger *row-level* un meccanismo che aggiorna il valore di *PrezzoTotale* in seguito ad update dell'attributo *Prezzo*. Discutere quindi la terminazione dei trigger.

B. Si hanno 3 tabelle: *STUDENTE* con 10.000 tuple in 1.000 blocchi, organizzata con un indice primario sulla matricola con fanout pari a 150 e un indice indiretto con fanout pari a 20 sul cognome; *ESAME* con 100.000 tuple in 1.000 blocchi, organizzata in modo sequenziale entry-based, con un indice secondario con fanout pari a 100 sulla matricola e un indice secondario con fanout pari a 50 sul codice del corso; *CORSO* con 1.000 tuple in 100 blocchi, organizzata con un indice primario con fanout pari a 50 su *CodCorso* e un indice secondario con fanout pari a 40 su *Titolo*.

Si ha quindi la seguente query

```
select *
from Studente S join Esame E on (S.Matr=E.Matr)
      join Corso C on (E.CodCorso=C.CodCorso)
where Studente like "Pa\%" and C.Titolo like "Sistemi\%"
```

Valutare il numero di operazioni di lettura di blocco richieste per il calcolo della query, confrontando le due strategie in cui si parte dall'applicazione della selezione rispettivamente prima sugli studenti o prima sui corsi, tenendo conto che la selettività della condizione sul cognome è pari all'1%, mentre la selettività sul titolo è pari allo 0.2 %.

C. Si ha la seguente tabella che descrive i dati di vendita di un prodotto:

VENDITE

Mese	Prodotto	Totale
Nov	Pane	2
Nov	Salame	3
Nov	Latte	3
Dic	Pane	2
Dic	Salame	2
Dic	Latte	2

Costruire il risultato della seguente query:

```
select sum(Totale)
from Vendite
group by Mese, Prodotto
with cube
```

Mostrare quindi come si potrebbe realizzare a partire da questo risultato la seguente query aggregata:

```
select sum(Totale)
from Vendite
where Prodotto = 'Pane'
```

D. Il seguente DTD rappresenta l'organizzazione di una scuola elementare.

```
<!ELEMENT SCUOLA ( MAESTRI, CLASSI )>
<!ELEMENT MAESTRI ( MAESTRO+ )>
<!ELEMENT CLASSI ( CLASSE+ )>
<!ELEMENT MAESTRO ( CODICEFISCALE, NOME, COGNOME )>
<!ELEMENT CLASSE ( NOME, INSEGNAMENTO+, STUDENTE+ )>
<!ELEMENT INSEGNAMENTO ( CODICEFISCALEMAESTRO, RUOLO )>
<!ELEMENT STUDENTE ( CODICEFISCALE, NOME, COGNOME )>
```

dove CODICEFISCALE, CODICEFISCALEMAESTRO, NOME, COGNOME e RUOLO sono elementi contenenti solo PCDATA.

1. Scrivere in XQuery l'interrogazione che estrae i cognomi dei maestri che hanno complessivamente più di 50 allievi.
2. Scrivere in XQuery l'interrogazione che estrae, per ogni bambino, l'elenco di tutti i suoi maestri in ordine alfabetico.
3. Si consideri lo schema a oggetti seguente che rappresenta le stesse informazioni:

```
create class Persona(char(16) CodFiscale, string Nome, string Cognome);

create class Maestro inherits Persona;

create class Studente inherits Persona;

create class Classe
    (string Nome,
     setof(struct(*Maestro Docente, string Materia)) Insegnamenti
     setof(*Studente) Studenti);
```

Esprimere in OQL la query che estrae i nomi e cognomi degli studenti che hanno "Rossi" come docente di matematica.