

Basi di dati II, primo modulo

Primo homework — 9 marzo 2010

Note

- Gli homework sono facoltativi ma è estremamente importante svolgerli (e anche discuterli ad esempio sul forum), perché le domande proposte nei compiti di esame possono essere molto simili.

Domanda 1 Alcuni DBMS prevedono la possibilità di includere in un indice i valori di altri attributi delle ennuple, oltre a quelli degli attributi su cui l'indice è realizzato. Ad esempio, una istruzione del tipo

```
CREATE INDEX contoCorrenteIX ON contoCorrente (numero) INCLUDE (saldo)
```

crea un indice sulla relazione `contoCorrente` includendo nelle foglie dell'indice, oltre ai valori di `numero` (su cui l'indice è realizzato), anche quelli di `saldo`. Confrontare (con riferimento ad esempio a specifiche operazioni di ricerca e di aggiornamento che potrebbero essere avvantaggiate o penalizzate) questa soluzione con le due seguenti

```
CREATE INDEX contoCorrenteIX ON contoCorrente (numero)
CREATE INDEX contoCorrenteIX ON contoCorrente (numero, saldo)
```

Domanda 2 Si considerino un sistema con blocchi di dimensione $B = 1000$ byte e puntatori ai blocchi di $p = 5$ byte e una base di dati sulle seguenti relazioni, ognuna delle quali (i) ha una struttura heap (ii) ha un indice secondario sulla chiave (iii) non ammette valori nulli

- $R_1(\underline{ABC})$, contenente $S_1 = 1.000.000$ ennuple di $l_1 = 40$ byte, di cui $l_A = 5$ per il campo chiave A
- $R_2(\underline{DEF})$, contenente $S_2 = 400.000$ ennuple di $l_2 = 100$ byte, di cui $l_D = 4$ per il campo chiave D
- $R_3(\underline{GHL})$, contenente $S_3 = 500.000$ ennuple di $l_3 = 25$ byte, di cui $l_G = 5$ per il campo chiave G

e con una vista definita come segue:

- `CREATE VIEW V AS SELECT * FROM (R1 LEFT JOIN R2 ON B=D) JOIN R3 ON C=G`

In tale contesto,

- mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni) per ciascuna delle seguenti interrogazioni
 1. `SELECT A, L FROM V`
 2. `SELECT A FROM V`
 3. `SELECT A, E FROM V`
- stimare il costo, in termini di numero di accessi a memoria secondaria (ignorando la presenza di eventuali buffer) per l'operazione 1.

Domanda 3 Siano r_1 ed r_2 due relazioni contenenti rispettivamente N_1 e N_2 ennuple, con fattore di blocco rispettivamente F_1 e F_2 . Si supponga che il sistema abbia a disposizione un buffer di dimensione pari a $M = 101$ blocchi. Calcolare il numero di accessi a memoria secondaria necessario per eseguire un join $r_1 \bowtie_{A_1=A_2} r_2$ (con A_1 attributo di r_1 e A_2 attributo di r_2), nei seguenti casi, da considerare separatamente l'uno dall'altro e assumendo che il DBMS sia in grado di eseguire il join solo con il metodo *nested-loop* (eventualmente utilizzando l'accesso diretto tramite un indice invece della scansione interna) e che utilizzi solo strutture primarie disordinate. Si supponga infine che il blocco abbia dimensioni $B = 1$ Kbyte, che i puntatori occupino $p = 4$ byte e i valori dei due attributi in questione occupino $k = 20$ byte ciascuno.

1. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_2 è la chiave di r_2 mentre i valori di A_1 in r_1 si ripetono mediamente in $e = 6$ ennuple; non vi sono indici
2. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_1 è la chiave di r_1 mentre i valori di A_2 in r_2 si ripetono mediamente in $e = 6$ ennuple; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$
3. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; gli attributi coinvolti non sono chiavi e hanno, ciascuno, valori che si ripetono mediamente in $e = 6$ ennuple; è definito un indice su $r_1(A_1)$
4. $N_1 = 5.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 20$; A_1 è la chiave di r_1 e A_2 è la chiave di r_2 ; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$