

## Tecnologia delle basi di dati — 18 settembre 2008

Tempo a disposizione: due ore.

**Domanda 1** (15%) Un fenomeno di interesse per la tecnologia delle basi di dati è il recente sviluppo delle memorie flash, che al momento hanno prestazioni e costi che le pongono in posizione intermedia fra la RAM e i dischi. La tabella seguente mostra alcuni dati interessanti (un po' semplificati) riguardo alle prestazioni di memorie flash e dischi:

	memorie flash	dischi
velocità di trasferimento	66MB/s	300MB/s
tempo di latenza	0.1 ms	12ms
tempo di lettura di un "blocco" da 4KB	0.16ms	12.01ms
tempo di lettura di un "blocco" da 256KB	3.98ms	12.85ms

Sulla base di tali dati:

- discutere le possibili implicazioni in termini di dimensioni dei blocchi e di tecniche di allocazione; ad esempio, quando è importante la contiguità? in quale caso possono convenire blocchi grandi o blocchi piccoli? (se possibile, fare riferimento sia ai blocchi per i record veri e propri sia a quelli per gli indici);
- osservando che si può in questo modo costruire una strategia a tre livelli (RAM, flash, dischi) al posto di quella attuale a due (RAM, dischi), ragionare sulla seguente alternativa: conviene considerare la memoria flash come estensione della RAM (e quindi dedicarla al buffer, che può così diventare enorme) oppure alla memoria persistente (che può così diventare più veloce)? Quali potrebbero essere le implicazioni di ciascuna alternativa sulla gestione dell'affidabilità?

**Domanda 2** (20%) Illustrare brevemente le motivazioni che portano, nei datawarehouse, a preferire l'utilizzo di identificatori ad-hoc, generati nella fase di caricamento, rispetto all'uso di chiavi provenienti dalle applicazioni.

**Domanda 3** (20%) 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 appelliIX ON appelli (codiceCorso) INCLUDE (numeroPrenotati)
```

crea un indice sulla relazione **appelli** (che supponiamo relativa a tutti gli appelli dell'anno; quindi **codiceCorso** non è chiave) includendo nelle foglie dell'indice, oltre ai valori di **codiceCorso** (su cui l'indice è realizzato), anche quelli di **numeroPrenotati**. Confrontare (con riferimento anche a specifiche operazioni di ricerca e di aggiornamento che potrebbero essere avvantaggiate o penalizzate) questa soluzione con le due seguenti

```
CREATE INDEX appelliIX ON appelli (codiceCorso)
CREATE INDEX appelliIX ON appelli (codiceCorso, numeroPrenotati)
```

(segue sul retro)

**Domanda 4** (30%) 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  $S_1 = 2.000.000$  ennuple di  $l_1 = 40$  byte, di cui  $l_A = 5$  per il campo chiave  $A$ ; su questa relazione è definito un vincolo di integrità referenziale fra l'attributo  $B$  e la chiave  $D$  della relazione  $R_2$ ;
- $R_2(\underline{DEF})$ , contenente  $S_2 = 800.000$  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 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 5** (15%) Si consideri un B-tree con nodi intermedi che contengono due chiavi e tre puntatori e foglie con due chiavi. Mostrare un possibile contenuto della struttura a seguito di inserimenti delle chiavi nel seguente ordine (a partire dall'albero vuoto): 16, 10, 8, 2, 1, 4, 3, 5, 21, 22, 23. Mostrare i passi salienti che portano a tale contenuto.