

# Tecnologia delle basi di dati — 15 luglio 2008

Tempo a disposizione: due ore.

**NB:** le risposte saranno valutate anche sulla base della chiarezza espositiva; sarà apprezzata la sinteticità.

## Domanda 1 (15%)

Commentare brevemente la seguente affermazione: “le tecniche per il controllo di concorrenza basate su 2PL e su timestamp si basano su condizioni sufficienti ma non necessarie per la view-serializzabilità.” In particolare, chiarire se essa è vera o falsa, spiegare perché e motivare le ragioni pratiche per le quali si utilizzano tali tecniche anziché la view-serializzabilità stessa.

## Domanda 2 (15%)

Come noto, esistono protocolli di scrittura che permettono di realizzare tecniche di recovery che evitano le operazioni di undo oppure le operazioni di redo. Spiegare perché i sistemi di basi di dati di solito utilizzano invece protocolli che richiedono sia operazioni di redo sia operazioni di undo. In particolare, spiegare se questa scelta si basa sull’assunzione che i guasti sono rari oppure sull’assunzione che i guasti sono frequenti.

## Domanda 3 (10%)

Illustrare le caratteristiche principali della memorizzazione “column-based,” cioè quella su cui si basa il sistema SADAS, illustrato in un seminario (e anche altri sistemi sperimentali in corso di sviluppo nel mondo). Accennare ai contesti in cui essa può risultare vantaggiosa, spiegando brevemente quali possono essere i vantaggi.

## Domanda 4 (30%)

Si consideri una relazione IMPIEGATO(Matricola, Cognome, Nome, DataNascita) con un numero di ennuple pari a  $N$  abbastanza stabile nel tempo (pur con molti inserimenti ed eliminazioni) e una dimensione di ciascuna ennupla (a lunghezza fissa) pari a  $L$  byte, di cui  $K$  per la chiave Matricola e  $C$  per il campo Cognome.

Supporre di avere a disposizione un DBMS che permetta strutture fisiche disordinate (heap), ordinate (con indice primario sparso) e hash e che preveda la possibilità di definire indici secondari e operi su un sistema operativo che utilizza blocchi di dimensione  $B$  e con puntatori ai blocchi di  $P$  caratteri.

Indicare quale possa essere l’organizzazione fisica preferita nel caso in cui le operazioni principali siano le seguenti:

1. ricerca sul cognome (o una sua sottostringa iniziale, abbastanza selettiva, in media una sottostringa identifica  $S = 10$  ennuple) con frequenza  $f_1$
2. ricerca sul numero di matricola, con frequenza  $f_2$
3. ricerca sulla base di un intervallo della data di nascita (poco selettivo, restituisce circa il 5% delle ennuple), con frequenza  $f_3$  molto minore di  $f_1$  e  $f_2$

assumendo  $N = 10.000.000$ ,  $L = 100$ ,  $K = 5$ ,  $C = 20$ ,  $B = 1.000$ ,  $P = 4$ ,  $f_1 = 100$ ,  $f_2 = 2.000$ ,  $f_3 = 1$ .

Individuare le alternative più sensate sulla base di ragionamenti qualitativi e poi valutarle quantitativamente, ignorando i benefici derivanti dai buffer e dalla contiguità di memorizzazione.

## Domanda 5 (15%)

Si supponga di dover realizzare un data mart relativo ad esami universitari. Fra i vari dati di interesse vi sono i voti riportati negli esami. Spiegare quali differenze possono sussistere nel data mart fra il caso in cui interessa ragionare sul voto medio e quello in cui interessa ragionare sulle distribuzioni per fasce di voto (cioè, ad esempio, quanti voti fra 18 e 20, quanti fra i 21 e 23 e così via). Spiegare anche come procedere se interessano entrambi gli aspetti.

## Domanda 6 (10%)

Spiegare perché, scrivendo un programma che accede a due basi di dati diverse utilizzando JDBC, non è possibile garantire il commit a due fasi. Indicare quali funzionalità aggiuntive sono necessarie per i server locali che vogliono realizzare tale servizio e come si può procedere con un sistema che non disponga di tali funzionalità.

## Domanda 7 (10%)

Le seguenti affermazioni sono tutte vere. Provare a dare una breve spiegazione per ciascuna di esse

1. i primi DBMS a oggetti non prevedevano linguaggi di interrogazione
2. successivamente, ci si è resi conto che un DBMS a oggetti non può non avere un linguaggio di interrogazione
3. OQL, il più noto linguaggio per basi di dati a oggetti, non prevede funzionalità per l’aggiornamento.