

Corso di Basi di dati
Prova scritta parziale
31 marzo 2000
Compito A

Tempo a disposizione: due ore. Libri e appunti aperti.

Domanda 1 (25%)

Si consideri una relazione IMPIEGATO(Matricola,Cognome,Nome,DataNascita) con un numero di ennuple pari a N abbastanza stabile nel tempo e una dimensione di ciascuna ennupla (a lunghezza fissa) pari a L byte.

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 .

Calcolare (approssimativamente) il numero di accessi a memoria secondaria (nell'unità di tempo) supponendo per la suddetta relazione varie organizzazioni (quelle che a prima vista si ritengono preferibili) nel caso in cui le operazioni principali siano le seguenti:

1. ricerca sul numero di matricola, con frequenza f_1
2. ricerca sul cognome (o una sua sottostringa iniziale, abbastanza selettiva, in media una sottostringa identifica $S = 4$ ennuple) con frequenza f_2
3. ricerca sulla base di un intervallo della data di nascita (poco selettivo), con frequenza f_3 molto minore di f_1 e f_2

In particolare, indicare la soluzione preferita (assumendo $N = 10000$, $L = 50$ e $B = 512$) nei casi

- (a) $f_1 = 100$, $f_2 = 500$, $f_3 = 1$
- (b) $f_1 = 500$, $f_2 = 100$, $f_3 = 1$

Commentare anche quanto possa essere opportuno fare scelte diverse se il numero di ennuple N può variare significativamente nel tempo.

Domanda 2 (20%)

Si consideri il seguente problema: uno stadio ha un certo numero di aree (identificate da lettere) ognuna con più settori (identificati da numeri, unici nell'ambito delle aree); ciascun settore ha più posti (identificati da numeri, unici all'interno del settore). Vengono accettate prenotazioni per (i) singoli posti; (ii) interi settori; (iii) intere aree. Illustrare una struttura di dati che permetta di rappresentare in modo compatto le prenotazioni e di rispondere in modo efficiente alle richieste di prenotazione. Suggerimento: utilizzare le idee che sono alla base del lock gerarchico; si noti però che l'albero non va rappresentato per intero (altrimenti la struttura non sarebbe compatta).

Domanda 3 (15%)

Descrivere (informalmente o in pseudocodice) un algoritmo efficiente per l'effettuazione del join di due relazioni in cui l'attributo di join ha un dominio costituito da pochi valori diversi (per esempio gli interi da 1 a 10). Nota: È possibile ottenere un algoritmo con complessità lineare (rispetto alle dimensioni del risultato), variante della tecnica "hash-based" per la realizzazione dei join.

Domanda 4 (20%)

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 non prevede funzionalità per l'aggiornamento.

Domanda 5 (20%)

Progettare uno schema multidimensionale a stella (star) o a fiocco di neve (snowflake) per un semplice data warehouse che voglia rappresentare informazioni relativamente agli incassi registrati da un insieme di film. I fatti di interesse sono le proiezioni, le misure gli incassi e il numero di spettatori, le dimensioni il luogo (cinema, con città, regione e area geografica [nord, centro, sud, isole]), il tempo (giorno, mese, trimestre), il film proiettato (con titolo, genere, nazione). Mostrare sia lo schema ER sia quello relazionale.

Corso di Basi di dati

Prova scritta parziale

31 marzo 2000

Compito B

Tempo a disposizione: due ore. Libri e appunti aperti.

Domanda 1 (25%)

Si consideri una relazione `STUDENTE(Matricola,Cognome,Nome,DataNascita)` con un numero di ennuple pari a N abbastanza stabile nel tempo e una dimensione di ciascuna ennupla (a lunghezza fissa) pari a L byte.

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 D .

Calcolare (approssimativamente) il numero di accessi a memoria secondaria (nell'unità di tempo) supponendo per la suddetta relazione varie organizzazioni (quelle che a prima vista si ritengono preferibili) 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 $S = 4$ 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), con frequenza f_3 molto minore di f_1 e f_2

In particolare, indicare la soluzione preferita (assumendo $N = 10000$, $L = 50$ e $D = 1024$) nei casi

- (a) $f_1 = 100$, $f_2 = 500$, $f_3 = 1$
- (b) $f_1 = 500$, $f_2 = 100$, $f_3 = 1$

Commentare anche quanto possa essere opportuno fare scelte diverse se il numero di ennuple N può variare significativamente nel tempo.

Domanda 2 (20%)

Si consideri il seguente problema: una sala concerti ha un certo numero di zone (identificate da lettere) ognuna con più palchi (identificati da numeri, unici nell'ambito delle zone); ciascun palco ha più posti (identificati da numeri, unici all'interno del palco). Vengono accettate prenotazioni per (i) singoli posti; (ii) interi palchi; (iii) intere zone. Illustrare una struttura di dati che permetta di rappresentare in modo compatto le prenotazioni e di rispondere in modo efficiente alle richieste di prenotazione. Suggerimento: utilizzare le idee che sono alla base del lock gerarchico; si noti però che l'albero non va rappresentato per intero (altrimenti la struttura non sarebbe compatta).

Domanda 3 (15%)

Descrivere (informalmente o in pseudocodice) un algoritmo efficiente per l'effettuazione del join di due relazioni in cui l'attributo di join ha un dominio costituito da pochi valori diversi (per esempio le lettere da A a L). Nota: È possibile ottenere un algoritmo con complessità lineare (rispetto alle dimensioni del risultato), variante della tecnica "hash-based" per la realizzazione dei join.

Domanda 4 (20%)

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 non prevede funzionalità per l'aggiornamento.

Domanda 5 (20%)

Progettare uno schema multidimensionale a stella (star) o a fiocco di neve (snowflake) per un semplice data warehouse che voglia rappresentare informazioni relativamente agli incassi registrati da un insieme di libri. I fatti di interesse sono le vendite, le misure gli incassi e il numero di copie, le dimensioni il luogo (libreria, con città, regione e area geografica [nord, centro, sud, isole]), il tempo (giorno, mese, trimestre), il libro (con titolo, genere, nazione). Mostrare sia lo schema ER sia quello relazionale.