

Basi di dati II, primo modulo (ord.270) — Tecnologia delle basi di dati (ord.509)

14 settembre 2009

Tempo a disposizione: un'ora e trenta minuti.

Domanda 1 (30%) 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`. Considerare un sistema con blocchi di dimensione $B = 1000$ byte e puntatori ai blocchi di $p = 3$ byte e una relazione `contoCorrente` con $N = 100.000$ ennuple, campo `numero` di $n_1 = 2$ byte e campo `saldo` di $n_2 = 5$ byte. In tale contesto, supponendo che tutti i livelli intermedi degli indici siano contenuti nel buffer (quindi gli accessi a memoria secondaria siano necessari solo per le foglie dell'indice) e assumendo il costo di una scrittura pari a $k = 3$ volte di quello di una lettura, confrontare le prestazioni dell'indice sopra mostrato con quelle dell'indice tradizionale

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

facendo riferimento ad un carico applicativo così composto:

- o_1 lettura del `saldo` di un `contoCorrente` dato il `numero`, con frequenza $f_1 = 1000$
- o_2 lettura e modifica del `saldo` di un `contoCorrente` dato il `numero`, con frequenza $f_2 = 200$
- o_3 lettura del valore di `numero` di tutte le ennuple di `contoCorrente`, con frequenza $f_3 = 20$

Domanda 2 (30%) Una catena di negozi gestisce le attività utilizzando, in ciascun negozio, una base di dati con le seguenti relazioni:

- `Prodotti(CodiceProdotto,Descrizione,Prezzo,Categoria)`
- `Categorie(Codice,Descrizione,MacroCategoria)`
- `MacroCategorie(Codice,Descrizione)`
- `Vendite(NumeroScontrino,Ora)`
- `DettaglioVendite(NumeroScontrino,CodiceProdotto,Quantità)`

Si noti che

- Le informazioni relative alle vendite vengono mantenute solo nel corso della giornata.
- Il prezzo di un prodotto può variare da un giorno all'altro.

Utilizzando tali informazioni, la catena vuole realizzare un data mart relativo alle vendite dei prodotti, avente come misure le quantità vendute e gli incassi, che permetta di effettuare analisi di tipo temporale (incluse, oltre ai giorni, anche le fasce orarie della giornata, ad esempio 9-10, 10-11 e così via, oppure mattina e pomeriggio) e su prodotti (singoli e per categoria) e sui negozi. Allo scopo:

1. specificare un possibile dettaglio del data mart; al riguardo, si supponga che la quantità delle vendite sia tale che si è deciso di non utilizzare il massimo livello di dettaglio, ma solo quello strettamente indispensabile (in altri termini, la grana non deve essere il singolo dettaglio di vendita, ma una opportuna aggregazione; **specificare esplicitamente la grana scelta**)
2. specificare l'interrogazione SQL necessaria per produrre, giornalmente, le nuove ennuple da inserire nella tabella dei fatti (utilizzare eventualmente una o più viste per facilitare la scrittura dell'interrogazione)

Domanda 3 (20%) Specificare a quali delle seguenti classi ciascuno degli schedule sotto mostrati appartiene: S (seriale), VSR (view-serializzabile), CSR (conflict-serializzabile), 2PL (generabile da uno scheduler basato sul lock a due fasi) and TS (generabile da uno scheduler che utilizzi il metodo dei timestamp; si assuma che gli identificatori delle transazioni corrispondano ai timestamp).

1. $r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$
2. $r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$
3. $r_2(x)w_2(x)r_1(x)w_1(x)$
4. $r_1(x)w_1(x)r_2(x)w_2(x)r_0(y)w_1(y)$

Domanda 4 (20%) Si consideri un B-tree con nodi intermedi che contengono due chiavi e tre puntatori e foglie con due chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 1, 7, 22, 18, 15, 17, 19, 16, 5. Mostrare l'albero dopo l'inserimento di tre e sei chiavi e alla fine.