Tecnologia delle basi di dati (ex Basi di dati, primo modulo)
17 giugno 2008

Tempo a disposizione: due ore e quindici minuti.
Cenni sulle soluzioni (per alcune domande)

Domanda 1 (30%) Si consideri la base di dati seguente, relativa alle telefonate di una singola giornata effettuate da clienti di telefonia fissa di una azienda telefonica:

- TELEFONATE(Orario, Chiamante, Chiamato, Durata) dove Orario indica l'istante preciso (ora, minuti, secondi) di inizio della telefonata, Chiamante è un'utenza (cioè un numero, comprensivo di prefisso) dell'azienda e Chiamato è un'utenza, dell'azienda stessa oppure di un'altra azienda
- UTENZE(Numero, SubDistretto, PianoTariffario) (le utenze dell'azienda, tutte nazionali), con vincolo di riferimento da SubDistretto verso SUBDISTRETTI
- UTENZEESTERNE(Numero, Tipo, Azienda) (le utenze di altre aziende) dove Tipo assume uno dei tre valori: nazionale fissa, nazionale mobile, estera
- SUBDISTRETTI(Codice, Descrizione, Distretto) con vincolo di riferimento da Distretto verso DISTRETTI
- DISTRETTI(Codice, Descrizione, Provincia)

Con riferimento a tale base di dati, progettare uno o più schemi dimensionali (specificando per ciascuno grana, misure e dimensioni e illustrando brevemente come le tabelle possono essere ottenute partendo dai dati disponibili) che permettano di rispondere facilmente ad interrogazioni quali ad esempio (la lista non ha pretesa di essere esaustiva):

1. calcolare il numero e la durata complessiva delle telefonate fatte da una specifica utenza (o da ciascuna utenza di un certo insieme, specificato attraverso il distretto, oppure il subdistretto oppure il piano tariffario), in ciascuna fascia oraria (0-6,6-12,...,18-24), in ciascun mese dell'anno, e in ciascun giorno della settimana, verso utenze dei vari tipi (nazionali fisse, nazionali mobili, estere)
2. calcolare il numero e la durata complessiva delle telefonate fatte da utenze di un certo distretto con un certo piano tariffario, verso utenze di ciascuno dei distretti (o verso utenze di altre aziende e di altri tipi), in ciascuna ora della giornata (0, 1, 2, ..., 23)
3. calcolare il numero delle telefonate fatte da utenze di un certo distretto con un certo piano tariffario, verso utenze di ciascuno dei distretti (o verso utenze di altre aziende e di altri tipi), in ciascuna ora della giornata (0, 1, 2, ..., 23), distinguendo fra classi di lunghezza (ad esempio, telefonate "brevi," "medie," "lunghe" e "lunghissime")

Si noti che:

- per ragioni di privacy non è possibile scendere oltre con i dettagli delle dimensioni quando è identificabile la singola utenza (interrogazione 1), mentre ciò è possibile quando sono aggregate le informazioni su tutte le utenze di un distretto (per questo motivo risulteranno probabilmente necessari almeno due schemi dimensionali)
- il piano tariffario di una utenza può cambiare nel tempo; spiegare come si tiene conto di questo aspetto
- i subdistretti sono fissi, mentre i distretti sono aggregazioni di subdistretti variabili nel tempo; spiegare come si tiene conto di questo aspetto

Possibile soluzione:

Per rispondere al requisito sulla "privacy," è opportuno prevedere due schemi dimensionali, uno con maggiore dettaglio sul chiamante e un dettaglio molto grossolano su chiamati, date e fasce orarie e l'altro con minore dettaglio sul chiamante e maggiore sulle altre dimensioni.

Il requisito della conformità delle dimensioni può essere perseguito attraverso la aggregazione delle dimensioni (ad esempio, fra utenza e distretto) oppure attraverso la separazione delle dimensioni (ad esempio, mese e giorno della settimana, senza data).

Dal punto di vista della creazione, si potrebbe costruire, nell'area di staging, un unico schema dimensionale con il massimo della granularità, da cui poi i due di interesse potrebbero essere ottenuti per aggregazione.

Primo schema dimensionale

- FattiTelefonateUtente(KUtente, KFasciaOrariaG, KMese, KGiornoSettimana, KTipoUtenzaDest, Numero, Durata)
- Utente(KUtente, Numero, KSubDistretto, Subdistretto, ..., KDistretto, Distretto, ..., PianoTariffario, ...) attenzione al piano tariffario ...
- FasciaOrariaG(KFasciaOrariaG, FasciaDi6Ore, ...)
- Mese(KMese, Mese, ..., Trimestre, ...)
- GiornoSettimana(KGiornoSettimana, Giorno, ...)
- TipoUtenza(KTipoUtenza, ...)

Commenti:

- sono indicate chiavi ad hoc per le dimensioni
- per la privatezza, si include solo ciò che è espressamente richiesto e niente di più; probabilmente è meglio non avere la data
- la tabella dei fatti si calcola con una aggregazione ...

Secondo schema dimensionale

- FattiTelefonateDistretti(KSubDistrettoChiamante, KSubDistrettoDestinazione, KPianoTariffario, KFasciaOrariaF, KData, KClasseLunghezza, Numero, Durata)
- SubDistretto(KSubDistretto, Subdistretto, ..., KDistretto, Distretto, ...,) si potrebbe aggregare a livello di distretto; in ogni caso, è una "slowly changing dimension"
- FasciaOrariaF(KFasciaOrariaF, FasciaDi1Ora, FasciaDi6Ore, ...)
- Data ...
- PianoTariffario
- ClasseLunghezza

Domanda 2 (20%) Ai fini dello studio congiunto di affidabilità e concorrenza, è necessario considerare negli schedule anche le operazioni di commit e abort (c_i e a_i indicano rispettivamente il commit e l'abort della transazione T_i) e considerare come schedule anche i relativi prefissi (cioè porzioni di schedule eseguite fino ad un certo momento, costituite anche da transazioni non concluse). Allo scopo, si considerino le seguenti definizioni, in cui per "schedule" si intende più in generale un "prefisso di schedule" e per "coppia di transazioni T_i, T_j in s," si intende "coppia di transazioni T_i, T_j distinte (cioè con $i \neq j$) che compaiano, almeno con un prefisso non vuoto, in s"; inoltre, si assuma che se T_i legge x da T_j, l'abort a_j non compare fra la scrittura $w_j(x)$ e la lettura $r_i(x)$:

- uno schedule s è *recoverable (RC)* se, per ogni coppia di transazioni T_i, T_j in s, vale la proprietà seguente: se T_i legge un qualche x da T_j in s e il commit c_i di T_i compare in s, allora anche il commit c_j di T_j compare in s e lo precede (in simboli $c_j <_s c_i$)
- uno schedule *evita gli abort in cascata (EAC)* se, per ogni coppia di transazioni T_i, T_j, vale la proprietà seguente: se T_i legge x da T_j in s allora $c_j <_s r_i(x)$ (il commit c_j di T_j compare nello schedule e precede la lettura di x da parte di T_i)
- uno schedule *è stretto (ST)* se, per ogni coppia di transazioni T_i, T_j, vale la proprietà seguente: se ci sono un'operazione $o_i(x)$ (lettura o scrittura) e una $w_j(x)$ con $w_j(x) <_s o_i(x)$, allora s deve includere anche c_j con $c_j <_s o_i(x)$ oppure a_j con $a_j <_s o_i(x)$

Dimostrare che

1. ogni schedule ST è anche EAC (ma non necessariamente viceversa)
2. ogni schedule EAC è anche RC (ma non necessariamente viceversa)
3. il 2PL stretto permette solo schedule ST

Domanda 3 (25%) Si considerino un sistema con blocchi di dimensione $B = 1000$ byte e puntatori ai blocchi di $P = 2$ byte e una relazione *T(A,B,C,D)* di cardinalità pari circa a $R = 6.000.000$, con ennuple di $L = 50$ byte e campo chiave A di $L_A = 5$ byte e campo B di $L_B = 3$ byte. Valutare i pro e i contro relativamente alla presenza di un indice secondario sulla chiave A e di un altro, pure secondario, su B, in presenza del seguente carico applicativo:

1. inserimento di una nuova ennupla (con verifica del soddisfacimento del vincolo di chiave), con frequenza $f_1 = 300$
2. ricerca di una ennupla sulla base del valore della chiave A, con frequenza $f_2 = 100$
3. ricerca di ennuple sulla base del valore di B, con frequenza $f_3 = 500$
4. elenco di tutte le ennuple, ordinato secondo il valore di B, con frequenza $f_4 = 10$

Ragionare in termini di costo degli accessi a memoria secondaria, assumendo disponibilità di buffer che permettano di mantenere stabilmente in memoria due livelli per ciascun indice e considerando che la relazione possa essere memorizzata in forma contigua (assumendo un rapporto 100:1 fra tempo di posizionamento della testina e tempo di lettura). Trascurare le problematiche relative alla concorrenza.

Possibile soluzione:

È necessario calcolare il costo delle quattro operazioni, in presenza e assenza degli indici. Notazioni:

N_T numero di blocchi della relazione T; è pari circa a $R/(B/L) = 300.000$

CONT riduzione di costo dovuta alla contiguità; è pari a 100

SEQ costo della scansione sequenziale $1 + (N_T - 1)/\text{CONT}$; è pari a circa 3.000

PROF$_A$ il fattore di blocco dell'indice è 1.000/7, circa 160 e quindi, con un riempimento di circa il 60-70%, il fan-out è circa 100 e quindi i livelli necessari sono 4

PROF$_B$ il fattore di blocco dell'indice è 1.000/5, circa 200 e quindi, con un riempimento di circa il 60-70%, il fan-out è circa 130 e quindi i livelli necessari sono ancora 4

Op1 È influenzata da entrambi gli indici anche se in modo diverso

nessun indice è necessaria la scansione sequenziale , per verificare il vincolo di chiave; costo pari a SEQ = 3.000; nessun costo per la manutenzione degli indici

indice su A accesso tramite l'indice; costo pari alla profondità dell'indice su A, più uno (per accedere al blocco del file) meno due (grazie ai buffer); costo pari a 3

indice su B scansione sequenziale per la verifica della chiave e accesso all'indice per l'aggiornamento

indici sia su A sia su B accesso ai due indici

Op2 È influenzata dal solo indice su A accessi diretti o sequenziali come sopra

Op3 È influenzata dal solo indice su B simile; aggiungere il fattore e reolativo alla molteplicità

Op4 Può essere influenzata dal solo indice su B:

senza indice su B è necessario ordinare; essendo noto che ci sono almeno 200 blocchi di buffer, si può pensare ad un merge-sort ad almeno 100 vie, che quindi può completare l'ordinamento in tre passate; se i blocchi di buffer fossero 1.000 (a anche un po' meno) si potrebbe procedere in due passate; ogni passata richiede una scansione sequenziale contigua per leggere e una per scrivere e quindi il costo potrebbe essere $4 \times$ SEQ o $6 \times$ SEQ, pari a circa 12.000 o 16.000

con indice su B sfruttando l'indice, si deve effettuare un accesso per ogni record, senza poter sfruttare la contiguità e quindi il costo è pari circa a R e quindi è comunque maggiore di quello che si avrebbe non usando l'indice: quindi, l'indice non viene usato

Questa operazione può quindi essere ignorata

Domanda 4 (20%) 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): 2, 17, 7, 15, 20, 19, 16, 18, 5, 1. Mostrare anche i passi salienti che portano a tale contenuto.

Domanda 5 (5%) Illustrare brevemente (meno di mezza pagina) le caratteristiche fondamentali dei sistemi ERP, eventualmente con riferimento al sistema SAP/R3.