Tecnologia delle basi di dati (ex Basi di dati, primo modulo)
20 giugno 2007 — Compito A

Tempo a disposizione: 2 ore e 15 minuti. **Nota**: è richiesta una "bella copia" comprensibile e ordinata.

Domanda 1 (25%) Si consideri la seguente base di dati relazionale, relativa alle ricette acquisite da un insieme di farmacie:

- Ricette(Numero, CodFarmacia, CFPaziente, Data)
- Farmacie(CodFarmacia, Nome, Via, NumeroCivico, Città)
- ElementiRicetta(NumeroRicetta, CodFarmaco, Quantità)
- Farmaci(Codice, Descrizione, CodMolecola, CodCasa, Prezzo, Fascia)
- Molecole(CodMolecola, Descrizione)
- Pazienti(CF, Cognome, Nome, DataNascita, Via, NumeroCivico, Città)
- CaseFarmaceutiche(CodCasa, Nome)
- ASL(Codice, Nome)
- Territorio(Via, Città, NumeroCivico, ASL)

Si noti che ci sono dati che cambiano nel tempo fra cui prezzi e fasce ('A', 'B' o 'C') dei farmaci e indirizzi dei pazienti.
Costruire, in tale contesto, uno schema a stella che permetta di analizzare le prescrizioni (quantità e prezzi complessivi) rispetto a

- data (dimensione standard i cui dettagli possono essere omessi);
- farmaci, con le loro proprietà (molecola e casa farmaceutica);
- prescrizione di farmaci nella stessa ricetta
- ASL di residenza e fascia d'età (ad esempio, 0-3,4-17, 18-30, ...; ma potrebbero variare) dei pazienti;
- ASL della farmacia

Supporre che, per ovvie ragioni di privacy, non possano essere riportati dati che permettano di risalire alle identità dei pazienti (CF, cognome, nome, data di nascita e indirizzo)
Specificare la grana dei fatti e indicare sinteticamente come si ottiene la relazione dei fatti da quelle della base di dati (supponendo disponibili, ove necessario, opportune tabelle per la conversione delle chiavi).

Possibile soluzione
Schema dimensionale

- FattiPrescrizioni(KData, KFarmaco, KRicetta, KASLfarmacia, KASLpaziente, KFasciaEtà, Quantità, Importo)
- Farmaci(KFramaco, Codice, Descrizione, CodMolecola, DescrizioneMolecola, CodCasa, NomeCasa, Fascia)
- ASL(KASL, CodiceASL, Nome)
- FasciaEtà(KFasciaEtà, ...)
- Data(KData, ...)

Commenti:

- sono indicate chiavi ad hoc per le dimensioni
- *Ricetta* è una dimensione "degenere," cioè senza attributi
- per la privatezza, si eliminano tutte le informazioni personali, indicando solo ASL del paziente e fascia d'età; quindi la grana scelta è "singole prescrizioni;" in effetti, vista la presenza di *KRicetta*, tutte le dimensioni, a parte *Ricetta* e *Farmaci*, sono secondarie
- la tabella dei fatti si calcola con join delle varie tabelle e nessuna aggregazione.

Domanda 2 (20%) Illustrare, brevemente ma con ordine, vantaggi e svantaggi (relativamente a vari tipi di operazioni) delle seguenti strutture fisiche, definite con riferimento ad uno stesso attributo A non chiave:

1. indice primario (sparso) multilivello statico;
2. indice secondario multilivello statico;
3. B+-tree secondario;
4. hash.

Inoltre, indicare, in ciascuno dei quattro casi (sia in forma simbolica sia in forma numerica), il costo di operazioni di ricerca basate su A ("trovare i record con un certo valore per l'attributo A"), con riferimento ad una relazione R contenente $L = 100.000.000$ ennuple di $R = 25$ byte ciascuna, di cui $a = 12$ per l'attributo A; si abbiano mediamente $m = 3$ record con lo stesso valore su A; supporre che i blocchi abbiano dimensione $B = 2$KB, approssimabile come $B = 2.000$ e che i puntatori ai blocchi abbiano lunghezza $p = 4$.

Possibile soluzione

1. indice primario (sparso) multilivello statico: va bene per ricerche puntuali e su intervallo, ma soffre in caso di aggiornamenti
2. indice secondario multilivello statico: va bene per ricerche puntuali e su intervallo (un po' meno del precedente, perché l'indice è più grande e soprattutto perché il file non è ordinato, il che è rilevante perché il campo non è chiave e comunque per ricerche su intervalli), ma soffre in caso di aggiornamenti
3. B+-tree secondario: come i precedenti, un po' meno efficiente ma senza svantaggi particolari in presenza di aggiornamenti
4. hash: molto efficiente per accessi puntuali, ma non per intervalli; degenera se le dimensioni variano significativamente.

Costi

1. $\text{LIV}_1 + 1 = 4$, dove LIV_1 è la profondità dell'indice, pari a $\lceil \log_F L/(B/R) \rceil = 3$, dove F è il fattore di blocco dell'indice, pari a $B/(a+p)$;
2. $\text{LIV}_2 + m = 7$, dove LIV_2 è la profondità dell'indice, pari a $\lceil \log_F L \rceil = 4$; si noti che il termine m è necessario qui perché il file è disordinato, mentre non è necessario nel caso precedente, perché i record sono tutti nello stesso blocco (salvo poche eccezioni)
3. $\text{LIV}_3 + m = 8$, dove LIV_3 è la profondità dell'indice, pari a $\lceil \log_{F'} L \rceil = 5$, dove $F' = 2/3F$ (supponendo un riempimento medio del 66%);
4. poco più di 1

Domanda 3 (20%) Si consideri una forma di equivalenza fra schedule denominata *final-state*-equivalenza, secondo la quale due schedule S_1 e S_2 sono equivalenti se, per ogni istanza i della base di dati, essi la trasformano nello stesso modo (e quindi $S_1(i) = S_2(i)$, se con $S(i)$ indichiamo l'istanza della base di dati ottenuta applicando a i le operazioni dello schedule S).

1. Formulare una definizione per questa proprietà, variante della definizione di view-equivalenza.
2. Spiegare, almeno intuitivamente, il rapporto che esiste fra final-state-equivalenza e view-equivalenza (sono equivalenti, incomparabili, oppure una implica l'altra?).

Possibile soluzione
Due schedule sono view-equivalenti se trasformano la base di dati nello stesso modo e offrono all'esterno gli stessi valori. La final-state-equivalenza non tiene conto di questa seconda proprietà. Gli schedule $w_1(x), r_2(x), w_3(x)$ e $w_1(x), w_3(x), r_2(x)$ sono final-state-equivalenti ma non view-equivalenti (perché hanno diversa relazione "legge-da"). Una definizione formale di final-state-equivalenza potrebbe richiedere (i) stesse scritture finali; (ii) stessa relazione "legge-da" per le transazioni che scrivono dopo la lettura. Di conseguenza, la final-state-equivalenza è condizione necessaria, ma non sufficiente per la view-equivalenza.

Domanda 4 (15%) Illustrare il protocollo del commit a due fasi utilizzando un esempio della vita quotidiana e in particolare l'organizzazione di una partita a tennis la quale è prevista la partecipazione di quattro specifiche persone insieme alla disponibilità di un campo presso un circolo.

Possibile soluzione
Deve esistere un coordinatore, che contatta ciascuno degli altri, chiedendo la disponibilità (in forma impegnativa) se tutti confermano e è disponibile il campo, allora il coordinatore conferma la partita. Altrimenti comunica a tutti l'annullamento.

Domanda 5 (20%) Specificare in quali casi (e perché) un indice può risultare utile per eseguire una proiezione con eliminazione di duplicati.

Possibile soluzione
L'eliminazione dei duplicati richiede l'ordinamento, che può essere favorito dalla presenza di un indice. In particolare, se l'indice è definito per un insieme di attributi da cui gli altri coinvolti nell'operazione dipendono funzionalmente, allora è sufficiente una scansione.
Osservazioni analoghe valgono per l'eliminazione dei duplicati nell'unione (domanda analoga del compito B).