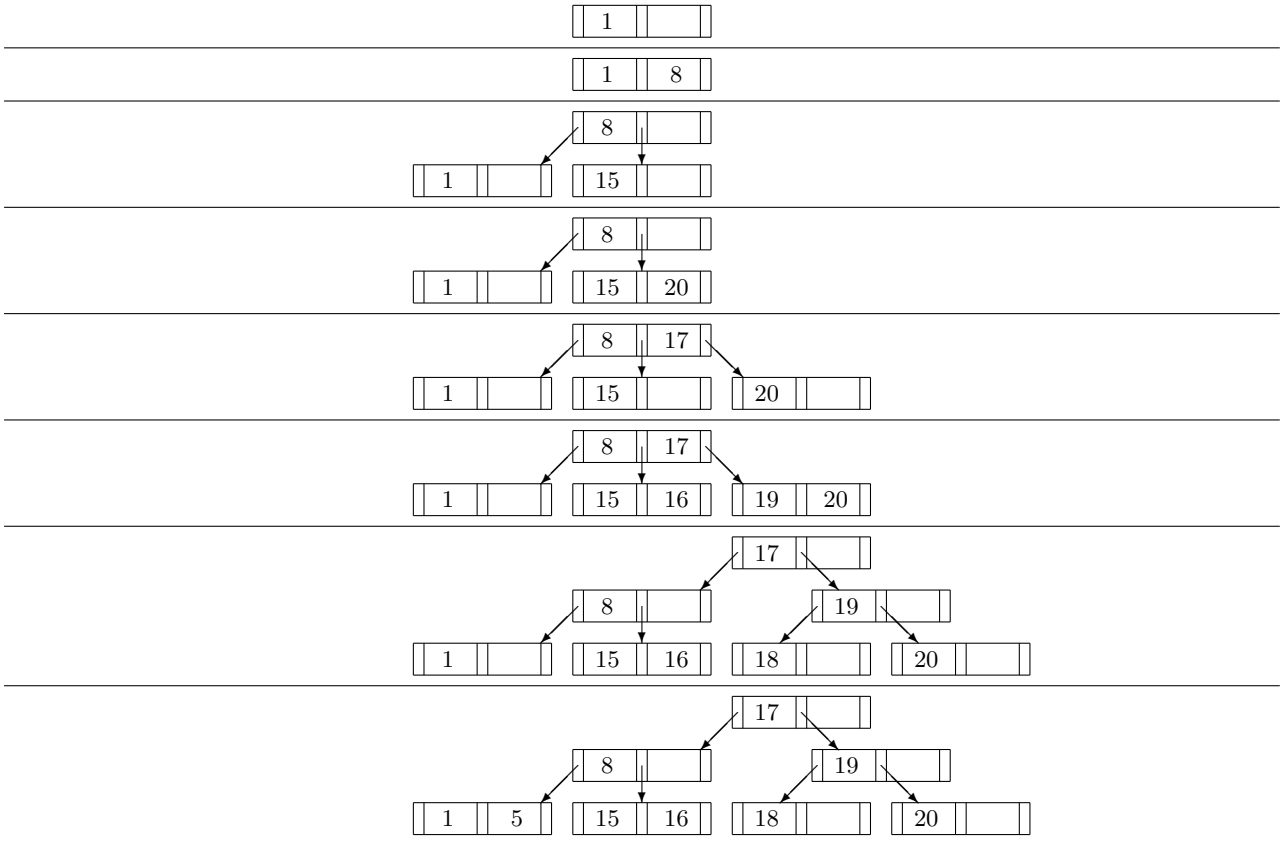
Tecnologia delle basi di dati (ex Basi di dati, primo modulo)

17 luglio 2006 — Cenni sulle soluzioni

Si fa riferimento al compito A, ma le osservazioni valgono anche per l'altro.

Domanda 1 (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): 1, 8, 15, 20, 17, 19, 16, 18, 5. Mostrare anche i passi salienti che portano a tale contenuto.

Possibile soluzione:



Domanda 2 (20%) Considerare le seguenti richieste di lettura e scrittura ricevute da un gestore del controllo di concorrenza basato su timestamp (assumendo che si tratti delle prime richieste ricevute dopo l'avvio del sistema):

$$r_3(x), r_2(x), r_4(y), w_2(x), r_6(y), r_1(x), w_3(x), w_4(y), w_7(x), w_6(y), r_5(x)$$

Indicare quali vengono accettate e quali rifiutate e, di conseguenza, quali transazioni vengono uccise.

Possibile soluzione:
Vengono rifiutate: $w_2(x), w_4(y), r_5(x)$

Domanda 3 (30%) Alcuni DBMS prevedono la possibilità di definire indici parziali, cioè indici definiti solo su parte di una relazione. Ad esempio, data una relazione $R(\underline{A}, B, C)$, si può definire un indice

```
CREATE INDEX RIX ON R (B) WHERE B>10
```

che supponiamo organizzato come un B+-tree sui valori di B, limitatamente a quelli che soddisfano la condizione $B > 10$ (e quindi solo le ennuple che soddisfano la condizione sono accessibili attraverso l'indice).
Discutere, con un breve commento, in quali dei seguenti casi si ritiene che questo indice sia più conveniente rispetto ad un indice standard definito con `CREATE INDEX RIX ON R (B)` (supporre che la relazione venga aggiornata di frequente, in particolare con molti inserimenti):

1. le ennuple con valore di B maggiore di 10 sono una piccola minoranza; le interrogazioni più frequenti fanno accessi puntuali sui valori di B maggiori di 10
2. i valori di B minori di 10 si ripetono moltissime volte, mentre quelli maggiori pochissime; si eseguono accessi puntuali su tutti i valori di B
3. le ennuple con valore di B maggiore di 10 sono la maggioranza e i valori di B maggiori di 10 si ripetono di frequente; le interrogazioni più frequenti fanno accessi puntuali su tutti i valori di B

Inoltre, illustrare sinteticamente come possono essere eseguite le seguenti interrogazioni e quale può essere il loro costo se la relazione ha struttura heap, $N = 500.000$ ennuple, di $l = 12$ byte ciascuna ($l_A = 4$ per ciascun campo), i blocchi hanno dimensione $B = 1.000$ e i puntatori ai blocchi hanno lunghezza $p = 3$; inoltre, i valori di B sono tutti positivi; quelli minori di 10 si ripetono ciascuno molte volte (il 95% delle ennuple contiene valori minori di 10), mentre quelli maggiori di 10 si ripetono pochissime volte ciascuno

1. `SELECT * FROM R WHERE B=5 AND C>20`
2. `SELECT * FROM R WHERE B=25`
3. `SELECT * FROM R WHERE B>100 AND B<200 AND C=20`

Possibile soluzione:
Prima serie di quesiti:

1. l'indice parziale porta grandi benefici, perché le ennuple in questione sono poche ma sono accedute molto di frequente
2. l'indice parziale porta grandi benefici, perché i valori minori di 10 sono poco selettivi e per essi la scansione sequenziale è preferibile; quelli maggiori di 10 traggono molto vantaggio dall'indice (che è più piccolo e ha meno aggiornamenti)
3. l'indice parziale porta svantaggi, perché rende inefficienti gli accessi alle ennuple con valori minori di 10, senza migliorare significativamente quelli alle ennuple con valori maggiori di 10

Seconda serie di quesiti:

1. è necessaria una scansione sequenziale, costo $N/(B/l)$
2. si può usare l'indice, costo pari alla profondità dell'albero p_I più il numero m medio di record per ciascun valore del campo B; per calcolare la profondità, consideriamo che
   - il fattore di blocco dell'indice è $B/(l_A + p) = 1.000/7 =$ ca. 130 e, considerando un riempimento parziale, in concreto possiamo approssimarlo a 100;
   - al livello delle foglie, il numero di blocchi dell'indice è pari al numero di record coinvolti (e cioè il 5% di N e quindi 25.000) diviso per il fattore di blocco quindi pari a circa 250;
   - la profondità è quindi approssimabile a uno più il logaritmo in base 100 di 250, arrotondato all'intero superiore, e quindi a 3 (in altri termini, 250 foglie, 3 nodi al livello immediatamente superiore e poi 1)
3. si può usare l'indice, costo pari alla profondità dell'albero più la scansione della porzione di indice coinvolta (nel caso peggiore, tutta)

Domanda 4 (30%) Si consideri la seguente base di dati relazionale, relativa alle prescrizioni di farmaci acquisite da un insieme di farmacie:

- Ricette(Numero, CodFarmacia, CFPaziente, Data)
- Farmacia(CodFarmacia, Nome)
- ElementiRicetta(NumeroRicetta, NumeroLinea, CodFarmaco)
- 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);
- ASL di residenza e fascia d'età (ad esempio, 0-3,4-17, 18-30, ...) dei pazienti;
- farmacia.

Supporre che, per ovvie ragioni di privacy, non possano essere riportati dati che permettano di risalire alle identità dei pazienti.
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, KFarmacia, KFarmaco, KFarmaco, KASL, KFasciaEtà, Quantità, Importo)
- Farmacia(KFarmacia, CodFarmacia, Nome)
- 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
- per la privatezza, si eliminano tutte le informazioni personali, raggruppando sulla base di quelle espressamente richieste (per l'età si potrebbe pensare ad una granularità più fine, per prepararsi a future aggregazioni diverse; quindi la grana scelta è "prescrizioni giornaliere per fascia d'età, farmaco, farmacia e ASL di appartenenza"
- la tabella dei fatti si calcola con una aggregazione (con conteggio delle linea e somma dei prezzi) del join di Ricette e ElementiRicetta, sulla base della grana scelta e previe opportune trasformazioni per ottenere la fascia di età.