Corso di Basi di dati — Prova scritta — 2 maggio 2000
Cenni sulle soluzioni

Tempo a disposizione: un'ora e quarantacinque minuti. Libri chiusi.

Domanda 1 (25%)
Come noto, alcuni DBMS permettono una tecnica di memorizzazione chiamata "co-clustering" o "clustering eterogeneo," in cui un file contiene record di due o più relazioni e tali record sono allocati (ad esempio ordinati) secondo i valori di opportuni campi dell'una e dell'altra relazione. Ad esempio, date due relazioni

- *Ordini(CodiceOrdine, Cliente, Data, Totale)*
- *LineeOrdine(CodiceOrdine, Linea, Prodotto, Importo)*

questa tecnica (con riferimento agli attributi *CodiceOrdine* delle due relazioni) permetterebbe una memorizzazione contigua di ciascun ordine con le rispettive "linee d'ordine," cioè dei prodotti ordinati (ciascun ordine fa riferimento a più prodotti, ognuno su una "linea").
Con riferimento all'esempio, indicare quali delle seguenti operazioni possono trarre vantaggio dall'uso di questa opportunità e quali ne possono essere penalizzate (spiegare la risposta possibilmente anche in termini quantitativi, attraverso l'uso di esempi):

1. stampa dei dettagli (cioè delle linee d'ordine) di tutti gli ordini (ordinati per codice)
2. stampa dei dettagli di un ordine
3. stampa delle informazioni sintetiche (codice, cliente, data, totale) di tutti gli ordini

Soluzione (cenni) L'aspetto cruciale da sottolineare è l'organizzazione del file in blocchi (in quanto, come noto, il numero di accessi in memoria secondaria è pari al numero di blocchi cui si deve accedere). La presenza di specifiche strutture fisiche (indici, hash, ordinamento) è marginale ai fini della risposta a questa domanda, perché esse possono avere sostanzialmente la stessa efficacia sia in presenza sia in assenza di "co-clustering." Le tre operazioni:

1. può ottenere un lieve vantaggio, in quanto il join si trova già preparato, ma la cosa è poco rilevante, perché è comunque necessario un ordinamento, che ha sostanzialmente lo stesso costo nei due casi;
2. qui si potrebbe avere un vantaggio, in quanto si troverebbero in uno stesso blocco sia le informazioni sintetiche sia i dettagli (ovviamente, nel caso singolo il vantaggio è minimo, ma se l'operazione è molto frequente, esso diventa significativo);
3. in questo caso si ha sicuramente un peggioramento, perché si deve accedere a tutti blocchi del file che contengono entrambe le relazioni, mentre senza "co-clustering" si dovrebbe accedere solo ai blocchi relativi alla relazione *Ordini*; quantitativamente: supponiamo che i blocchi siano di 1000byte e che *Ordini* abbia 10.000 ennuple di 100 byte ciascuna, mentre *LineeOrdine* abbia 200.000 ennuple di 50 byte ciascuna; senza "co-clustering" *Ordini* occupa 1000 blocchi (e quindi questa operazione richiede 1000 accessi) mentre con il "co-clustering" le due relazioni (non più scandibili separatamente) occupano 11000 blocchi (e l'operazione richiede quindi 11mila accessi).

Domanda 2 (25%)
Le seguenti situazioni corrispondono ad alcune delle note anomalie delle transazioni concorrenti. Commentare brevemente ciascuna di esse e spiegare come il 2PL (con le sue estensioni) riesce ad evitarle.

1. Un cliente consulta un calendario di concerti e ne individua uno che gli interessa. Quando chiede il biglietto gli viene detto che il calendario non era definitivo e quel concerto non esiste.
2. Un signore ha dieci milioni sul proprio conto corrente e firma due assegni da tre milioni ciascuno. I due beneficiari si presentano quasi contemporaneamente a due impiegati diversi della stessa banca, ciascuno dei quali verifica che i soldi sono disponibili (ci sono dieci milioni) e, pagato l'assegno, registra il nuovo saldo di sette milioni.

3. Un appassionato lettore di gialli chiede quali libri siano disponibili di Agatha Christie. In risposta, riceve un elenco di tre libri. Chiede di ordinarli tutti. Quando li riceve, sono quattro.

Soluzione (cenni)

1. Questo potrebbe essere assimilato ad un caso di lettura sporca (il calendario non era definitivo, non c'era stato un commit). Da un altro punto di vista, lo si poteva vedere come un caso di lettura inconsistente. Il 2PL stretto risolve comunque entrambi.
2. Un classico caso di perdita di aggiornamento, superabile con il 2PL.
3. In questo caso, il problema non è nel numero di libri, ma nel fatto che viene inserito un nuovo record, relativamente al quale i lock sui tre record preesistenti non hanno effetto; qui solo un "lock sul predicato" può risultare efficace: si deve impedire che vengano inseriti ulteriori libri di Agatha Christie.

Domanda 3 (15%)
In quale dei seguenti casi le transazioni vengono rilanciate con lo stesso timestamp e in quale con un nuovo timestamp? Spiegare perché non sussiste conflitto fra i due tipi di generazione.

- prevenzione (o rimozione) dello stallo
- controllo di concorrenza basato su timestamp

Soluzione (cenni)

- con lo stesso timestamp (per evitare la starvation)
- con nuovo timestamp (perché in questo protocollo vengono uccise le transazioni "meno giovani" che svolgano azioni in ritardo)

Non sussiste conflitto perché le due tecniche vengono usate in contesti diversi, che non si presentano mai assieme (la prima nel controllo di concorrenza basato su 2PL e la seconda in quello basato su timestamp).

Domanda 4 (15%)
Indicare (con un breve commento) quali delle seguenti affermazioni sono vere e quali false.

1. un data warehouse non è una base di dati
2. un data warehouse ha requisiti di concorrenza diversi da quelli per la basi di dati OLTP
3. un data warehouse è una risorsa aziendale integrata, spesso più delle basi di dati OLTP
4. un data warehouse è soggetto a molte, brevi operazioni di aggiornamento, al fine di garantire l'attualità dei dati

Soluzione (cenni)

FALSO un data warehouse è una base di dati, con specifiche finalità, ma comunque una base di dati
VERO le operazioni sono prevalentemente di interrogazione e per giunta fatte da pochi utenti
VERO succede spesso che un data warehouse contenga dati provenienti da varie basi di dati OLTP non altrimenti integrate
FALSO le politiche di aggiornamento possono essere diverse, a seconda delle esigenze, ma in generale l'attualità dei dati non è prioritaria e gli aggiornamenti sono fatti periodicamente

Domanda 5 (20%)
Descrivere brevemente alcune delle funzionalità più significative dei sistemi di basi di dati "object-relational."
Soluzione (cenni) Gli argomenti principali cui far riferimento sono l'approccio evoluzionario (e la compatibilità con i sistemi relazionali), la possibilità di utilizzare gli identificatori di oggetto per costruire riferimenti espliciti e la possibilità di definire e riutilizzare tipi, definendo anche operazioni su di essi.