Basi di dati II, primo modulo
Esercizi di autovalutazione — 20 marzo 2012

Domanda 1 Il check-point (quiescente) prevede, fra le altre, le seguenti attività (durante le quali non sono accettate richieste di nuove transazioni):

1. scrittura su disco (flush) di tutte le pagine sporche del buffer
2. scrittura in memoria stabile di tutti i record del log incluso il record di checkpoint

Spiegare perché il checkpoint sarebbe inutilizzabile se le operazioni avvenissero in ordine inverso (prima la scrittura del record di checkpoint e poi la flush del buffer).

Domanda 2 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 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").
Supponendo che, nell'uno come nell'altro caso, vi sia un indice secondario sulla chiave primaria in ciascuna delle relazioni (sulla relazione LineeOridne, l'indice sia su *CodiceOrdine* e *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 di tutte le informazioni (incluse i dettagli delle linee d'ordine) di tutti gli ordini (ordinati per codice)
2. stampa di tutte le informazioni di un ordine
3. stampa delle informazioni sintetiche (codice, cliente, data, totale) di tutti gli ordini

Domanda 3 Si supponga di avere un recovery manager che utilizzi un checkpoint non quiescente e che scriva record di update ("SetStringRecord" secondo la terminologia di SimpleDB) aventi la forma seguente:

<SETSTRING, TxID, TableName, BlkNo, Offset, BeforeValue, AfterValue >

Si noti che, rispetto alla notazione usata sul libro, l'oggetto dell'operazione viene identificato da TableName (nome della relazione o meglio del file che la memorizza), BlkNo (numero del blocco nel file), Offset (posizione del valore di interesse nel blocco).

In tale contesto, supporre che il recovery manager, al riavvio dopo un crash, trovi i seguenti record nel log:

```
<START, 1>
<START, 2>
<SETSTRING, 2, Impiegati, 33, 0, xxxx, Rossi>
<SETSTRING, 1, Impiegati, 44, 0, xxxx, Neri>
<START, 3>
<COMMIT, 2>
<SETSTRING, 3, Impiegati, 33, 0, Rossi, Verdi>
<START, 4>
<SETSTRING, 4, Impiegati, 55, 0, xxxx, Bruni>
<NQCKPT, 1, 3, 4>
<SETSTRING, 4, Impiegati, 55, 0, Bruni, Neri>
<SETSTRING, 4, Impiegati, 66, 0, xxxx, Bianchi>
<START, 5>
<COMMIT, 4>
```

1. Indicare quali modifiche sulla base di dati vengono eseguite durante un recovery di tipo undo-redo.
2. Indicare quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo undo-only (cioè no-redo)
3. Indicare quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo redo-only (cioè no-undo)
4. è possibile che la transazione T_1 sia andata in commit?
5. è possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 20 della relazione Impiegati?
6. è possibile che la transazione T_1 abbia modificato su disco il blocco 20 della relazione Impiegati?
7. è possibile che la transazione T_1 non abbia modificato su disco il blocco 44 della relazione Impiegati?