

Basi di dati II — Esame — 7 settembre 2015 — Compito A

Tempo a disposizione: due ore e trenta minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%) Si consideri una base di dati sulle seguenti relazioni, ognuna delle quali ha una struttura heap e un indice secondario sulla chiave (si suppongano le relazioni tutte molto grandi, con R_1 più piccola delle altre due)

- $R_1(\underline{A}BC)$
- $R_2(\underline{D}EF)$
- $R_3(\underline{G}HL)$

e con una vista definita come segue:

- `CREATE VIEW V AS SELECT * FROM (R1 JOIN R2 ON B=D) LEFT JOIN R3 ON C=G`

In tale contesto, supponendo che il sistema esegua join solo con nested loop (senza trarre profitto significativo dal buffer), utilizzando gli indici ove definiti, mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni, quindi indicando per il come viene eseguito) per ciascuna delle seguenti interrogazioni

```
SELECT A, E, F, H, L FROM V
```

```
SELECT A, B, C, E, F FROM V
```

```
SELECT A, E, F FROM V
```

Domanda 2 (15%) Una catena di supermercati ha una base di dati dei propri clienti che dispongono di una “tessera fedeltà,” con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l’uso della tessera e (b) il negozio della catena cui fa riferimento (quello presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l’interrogazione che calcola, per ciascun negozio, il numero dei clienti, la somma dei punti dei clienti e la relativa media per cliente. Indicare quale livello di isolamento (READ UNCOMMITTED, ... si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L’operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L’operazione è eseguita mentre vengono inseriti molti nuovi clienti, con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L’operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
4. L’operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L’operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.

1.	2.	3.	4.	5.

Domanda 3 (15%) Un *quad tree*, nella sua versione più semplice, è un indice definito con riferimento a due diversi attributi di una relazione. Siano A e B gli attributi in questione. La struttura è costituita da un albero in cui ogni nodo intermedio ha due etichette, una per ciascuno degli attributi, e quattro sottoalberi. Se i valori di un nodo sono (a, b) , allora

- il primo sottoalbero contiene riferimenti a record con valori di A minori di a e valori di B minori di b
- il secondo a record con valori di A minori di a e valori di B maggiori o uguali a b
- il terzo a record con valori di A maggiori o uguali ad a e valori di B minori di b
- il quarto a record con valori di A maggiori o uguali ad a e valori di B maggiori o uguali a b

Intuitivamente, ogni nodo divide lo spazio in quattro parti e quindi una foglia di un *quad tree* contiene riferimenti corrispondenti ad un “quadrato” dello spazio bidimensionale, con valori di A compresi in un intervallo $[a_1, a_2]$ e quelli di B in un intervallo $[b_1, b_2]$; in contrasto, una foglia di un $B+$ -tree definito su A, B (nell’ordine) contiene riferimenti “linearizzati,” cioè ordinati per A e, a parità di A , per B .

Consideriamo ora una relazione di $N = 4.000.000$ di ennuple, con fattore di blocco $F = 10$, in cui i due attributi A e B , insieme, formino una chiave e in cui vi siano circa $v = 2.000$ valori diversi per ciascun attributo (abbastanza densi, ad esempio valori numerici compresi fra 1 e 2.100 e distribuiti omogeneamente) e che ciascun valore di A sia associato a circa 2.000 valori diversi di B e viceversa, con una varianza abbastanza piccola. Supponiamo che le foglie contengano circa 100 riferimenti ciascuna (sia nel caso del *quad tree* sia in quello del $B+$ -tree) e di avere un sistema che permetta di caricare nel buffer tutti i livelli intermedi (ma non le foglie) sia per i *quad tree* sia per i $B+$ -tree. Si noti che in questo caso “quadrato” sopra citato contiene riferimenti con (circa) 10 valori di A e 10 di B : una foglia intermedia avrà ad esempio valori fra 50 e 60 per A e 120 e 130 per B .

Calcolare il costo (cioè il numero di foglie da visitare e, se serve, dei record del file) delle seguenti operazioni nel caso in cui sulla relazione (disordinata) è definito un *quad tree* e in quello in cui è definito un $B+$ -tree su A, B :

1. conteggio delle ennuple con un dato valore di B
2. accesso alle ennuple con un dato valore di A
3. conteggio delle ennuple con un dato valore di A
4. conteggio delle ennuple con valore di B compreso fra 100 e 115 (si tratta di circa 30.000 ennuple)
5. conteggio delle ennuple con valore di A compreso fra 100 e 115 (si tratta di circa 30.000 ennuple)
6. conteggio delle ennuple con valore di A compreso fra 100 e 115 e valore di B compreso fra 100 e 115 (si tratta di circa 200 ennuple)

Rispondere riempiendo la tabella alla pagina seguente, con qualche commento che spieghi la risposta:

Basi di dati II — 7 settembre 2015 — Compito A

	quad tree su A, B	B+-tree su A, B (nell'ordine)
1		
2		
3		
4		
5		
6		

Domanda 4 (15%)

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, 2>
<SETSTRING, 2, Impiegati, 33, 0, xxxx, Rossi>
<START, 1>
<SETSTRING, 1, Impiegati, 13, 0, xxxx, Neri>
<START, 3>
<COMMIT, 2>
<SETSTRING, 3, Impiegati, 33, 0, Rossi, Verdi>
<START, 4>
<SETSTRING, 4, Impiegati, 35, 0, xxxx, Bruni>
<NQCKPT, 1, 3, 4>
<SETSTRING, 4, Impiegati, 35, 0, Bruni, Neri>
<SETSTRING, 4, Impiegati, 66, 0, xxxx, Bianchi>
<START, 5>
<COMMIT, 4>
```

1. Fino a quale record del log arriva la scansione a ritroso?

2. Indicare le azioni di undo e redo che debbono essere eseguite (nell’ordine) durante un recovery di tipo undo-redo.

3. Indicare le azioni di redo che debbono essere eseguite (nell’ordine) durante un recovery di tipo redo-only.

4. È possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 28 della relazione Impiegati? Spiegare perché e in caso affermativo spiegarne le conseguenze.

5. In caso di strategia redo-only, è possibile che la transazione T_1 abbia modificato su disco il blocco 28 della relazione Impiegati? Spiegare.

6. È possibile che la transazione T_1 abbia modificato su disco il blocco 13 della relazione Impiegati? Spiegare.

Basi di dati II — 7 settembre 2015 — Compito A

Domanda 5 (15%)

Considerare un sistema distribuito su cui vengono eseguite due transazioni che coinvolgono tre nodi, un coordinatore M (lo stesso per entrambe le transazioni) e due partecipanti R1 e R2. Dopo la richiesta del coordinatore M di **prepare** (abbreviata con **prep**) per la prima transazione, i due partecipanti ricevono e rispondono correttamente, e uno dei due, R2, va in crash subito dopo aver risposto. Il coordinatore riceve le risposte, prende la decisione, invia i relativi messaggi (questi passi **non** sono stati indicati sotto e vanno quindi scritti) e subito dopo va anch'esso in crash (quindi senza fare in tempo a ricevere le conferme). Viceversa, per la seconda transazione, il coordinatore invia il messaggio di **prepare** ma non fa in tempo a ricevere le risposte. Indicare, nello schema sottostante, una possibile sequenza di scritture sui log e invio di messaggi (che includa anche i passi sopra illustrati), supponendo che entrambi i nodi siano ripristinati abbastanza presto (ma che il coordinatore perda alcuni messaggi di risposta inviati ad esso a seguito del commit). Per i messaggi si usi la notazione *tipo(transaz)→destinatari* (come nell'esempio: **prepare**(T1)→R1,R2). Supporre che nel log del coordinatore si scrivano solo i record di **prepare**, **commit** e **complete**, con i messaggi gestiti invece in memoria. Indicare ragionevoli istanti per i timeout, che permettano di concludere entrambe le transazioni.

Nodo M		Nodo R1		Nodo R2	
Log	Messaggi	Log	Messaggi	Log	Messaggi
prep(T1,R1,R2)	prep(T1)→R1,R2				<i>crash</i>
prep(T2,R1,R2)	prep(T2)→R1,R2				
	<i>crash</i>				
	<i>restart</i>				
					<i>restart</i>

Basi di dati II — 7 settembre 2015 — Compito A

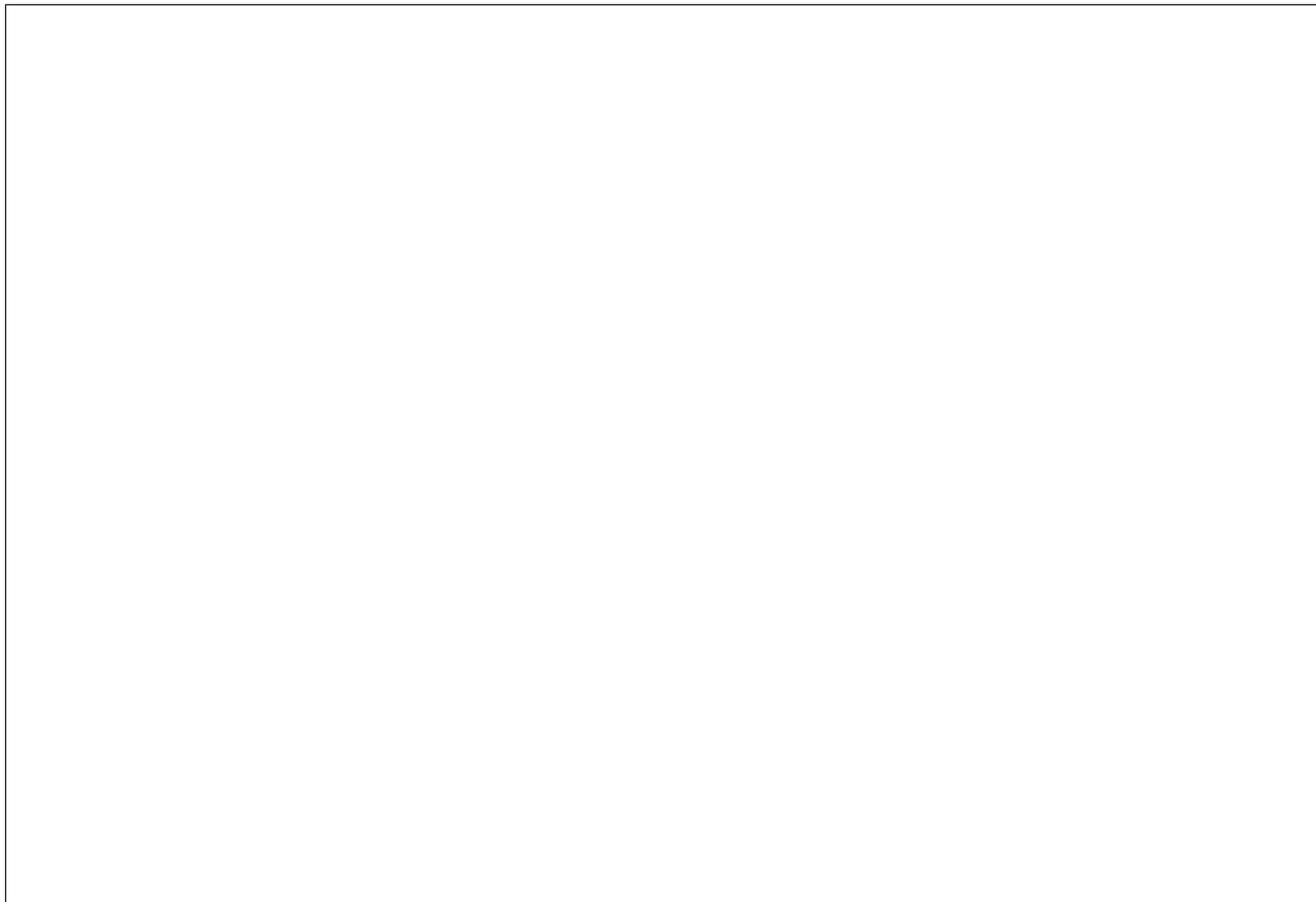
Domanda 6 (10%) Si consideri uno schema dimensionale utilizzato per analizzare le vendite in una catena di supermercati che, fra le dimensioni, ne preveda una sui negozi, come la seguente:

<u>KNegozi</u>	Nome	Città	Provincia	Regione
101	Pane buono	Arzachena	SS	Sardegna
102	Bontà	Arzachena	SS	Sardegna
103	Pane e vino	S. Teodoro	NU	Sardegna
104	Pane Sardo	Nuoro	NU	Sardegna
105	Pasta e pane	Palermo	PA	Sicilia
...

Si supponga ora che si presenti la seguente esigenza di modifica:

- sia pure molto raramente, le province cambiano; specificamente, si supponga che interessi gestire la modifica delle province della Sardegna avvenuta nel 2005 (ad esempio, da tale data il comune di Arzachena e quello di S. Teodoro appartengono alla provincia OT, Olbia-Tempio); in questo caso, si supponga che non interessi tanto la correlazione fra data della vendita e provincia nel momento della vendita, quanto la possibilità di fare analisi con riferimento alle due versioni del territorio, quella con le vecchie province e quella con le nuove (N.B. supporre che non interessino altre versioni oltre a queste due).

Modificare la dimensione (mostrando la nuova versione per la tabella in figura, con brevi commenti se necessario).



Basi di dati II — 7 settembre 2015 — Compito A

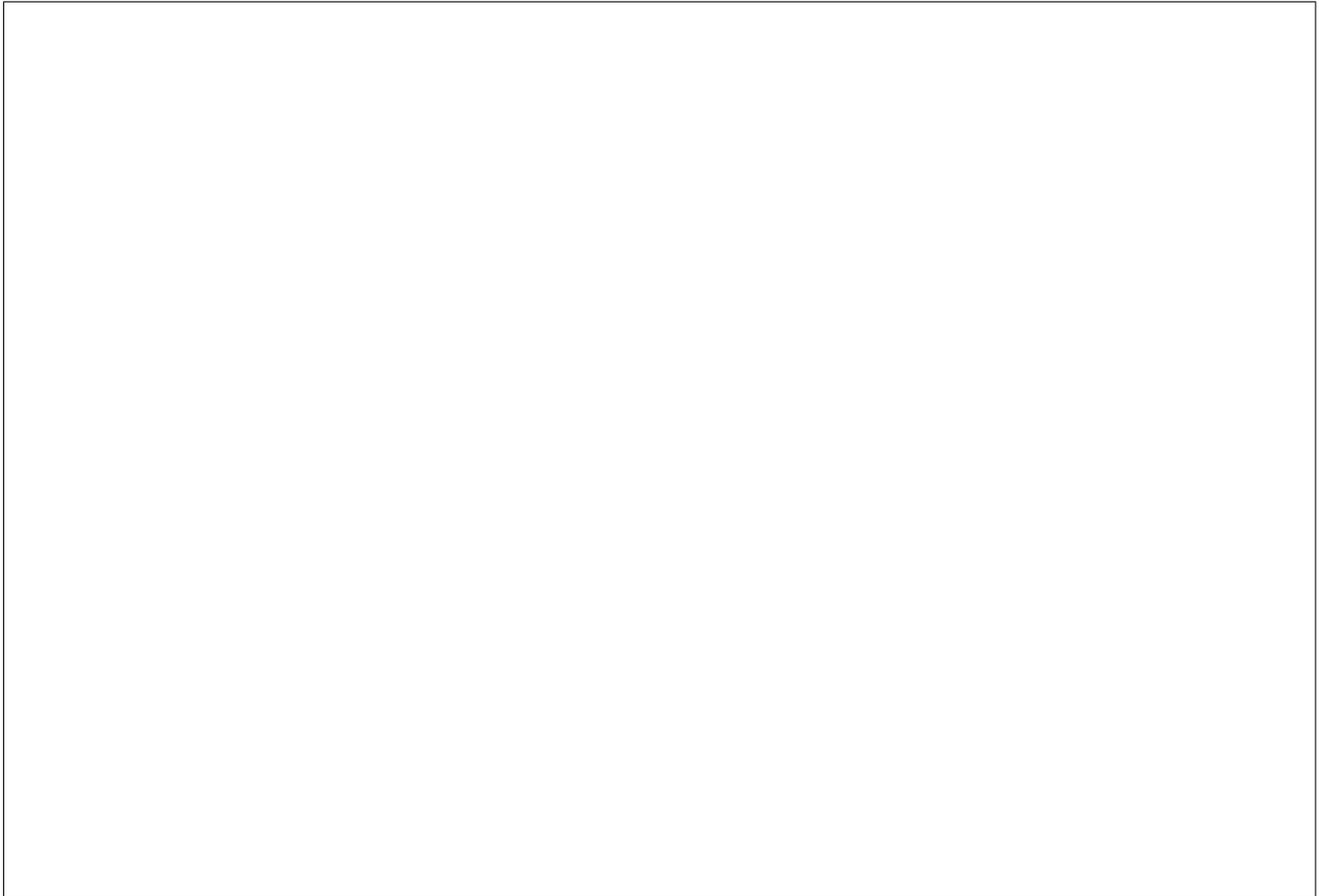
Domanda 7 (10%) Con riferimento allo schema dimensionale citato nella domanda precedente si supponga che la seguente sia la struttura della tabella dei fatti, con alcune delle ennuple:

<u>KNegozio</u>	<u>KProdotto</u>	<u>KData</u>	Quantità	Incasso
101	201	301	243	700,50
101	202	301	4	32,00
101	202	302	6	49,00
102	201	301	22	262,00
...

Si supponga ora che:

- per ogni negozio, interessa rappresentare anche il direttore, per svolgere analisi sulle vendite ascrivibili al direttore stesso; i direttori cambiano nel tempo e passano da un negozio all'altro (e possono anche dirigere due negozi nello stesso momento; ma ogni negozio ha, in un certo giorno, un solo direttore); è disponibile l'informazione relativa ai direttori dei negozi nel tempo (per tutto il periodo, anche passato, di interesse).

Modificare la tabella dei fatti (discutendo in particolare come si potrebbero aggiornare le sue ennuple e osservando se e di quanto varia la sua cardinalità; mostrare anche la nuova versione della tabella in figura).



Basi di dati II — Esame — 7 settembre 2015 — Compito B

Tempo a disposizione: due ore e trenta minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%) Si consideri una base di dati sulle seguenti relazioni, ognuna delle quali ha una struttura heap e un indice secondario sulla chiave (si suppongano le relazioni tutte molto grandi, con S_1 più piccola delle altre due)

- $S_1(\underline{A}BC)$
- $S_2(\underline{D}EF)$
- $S_3(\underline{G}HL)$

e con una vista definita come segue:

- `CREATE VIEW V AS SELECT * FROM (S1 JOIN S2 ON B=D) LEFT JOIN S3 ON C=G`

In tale contesto, supponendo che il sistema esegua join solo con nested loop (senza trarre profitto significativo dal buffer), utilizzando gli indici ove definiti, mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni, quindi indicando per il come viene eseguito) per ciascuna delle seguenti interrogazioni

```
SELECT A, E, F FROM V
```

```
SELECT A, B, C, E, F FROM V
```

```
SELECT A, E, F, H, L FROM V
```

Domanda 2 (15%) Una catena di supermercati ha una base di dati dei propri clienti che dispongono di una “tessera fedeltà,” con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l’uso della tessera e (b) il negozio della catena cui fa riferimento (quello presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l’interrogazione che calcola, per ciascun negozio, il numero dei clienti, la somma dei punti dei clienti e la relativa media per cliente. Indicare quale livello di isolamento (READ UNCOMMITTED, ... si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L’operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L’operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L’operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
4. L’operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L’operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.

1.	2.	3.	4.	5.

Domanda 3 (15%) Un *quad tree*, nella sua versione più semplice, è un indice definito con riferimento a due diversi attributi di una relazione. Siano A e B gli attributi in questione. La struttura è costituita da un albero in cui ogni nodo intermedio ha due etichette, una per ciascuno degli attributi, e quattro sottoalberi. Se i valori di un nodo sono (a, b) , allora

- il primo sottoalbero contiene riferimenti a record con valori di A minori di a e valori di B minori di b
- il secondo a record con valori di A minori di a e valori di B maggiori o uguali a b
- il terzo a record con valori di A maggiori o uguali ad a e valori di B minori di b
- il quarto a record con valori di A maggiori o uguali ad a e valori di B maggiori o uguali a b

Intuitivamente, ogni nodo divide lo spazio in quattro parti e quindi una foglia di un *quad tree* contiene riferimenti corrispondenti ad un “quadrato” dello spazio bidimensionale, con valori di A compresi in un intervallo $[a_1, a_2)$ e quelli di B in un intervallo $[b_1, b_2)$; in contrasto, una foglia di un $B+$ -tree definito su A, B (nell’ordine) contiene riferimenti “linearizzati,” cioè ordinati per A e, a parità di A , per B .

Consideriamo ora una relazione di $N = 4.000.000$ di ennuple, con fattore di blocco $F = 10$, in cui i due attributi A e B , insieme, formino una chiave e in cui vi siano circa $v = 2.000$ valori diversi per ciascun attributo (abbastanza densi, ad esempio valori numerici compresi fra 1 e 2.100 e distribuiti omogeneamente) e che ciascun valore di A sia associato a circa 2.000 valori diversi di B e viceversa, con una varianza abbastanza piccola. Supponiamo che le foglie contengano circa 100 riferimenti ciascuna (sia nel caso del *quad tree* sia in quello del $B+$ -tree) e di avere un sistema che permetta di caricare nel buffer tutti i livelli intermedi (ma non le foglie) sia per i *quad tree* sia per i $B+$ -tree. Si noti che in questo caso “quadrato” sopra citato contiene riferimenti con (circa) 10 valori di A e 10 di B : una foglia intermedia avrà ad esempio valori fra 50 e 60 per A e 120 e 130 per B .

Calcolare il costo (cioè il numero di foglie da visitare e, se serve, dei record del file) delle seguenti operazioni nel caso in cui sulla relazione (disordinata) è definito un *quad tree* e in quello in cui è definito un $B+$ -tree su A, B :

1. conteggio delle ennuple con un dato valore di A
2. accesso alle ennuple con un dato valore di A
3. conteggio delle ennuple con un dato valore di B
4. conteggio delle ennuple con valore di A compreso fra 100 e 115 (si tratta di circa 30.000 ennuple)
5. conteggio delle ennuple con valore di B compreso fra 100 e 115 (si tratta di circa 30.000 ennuple)
6. conteggio delle ennuple con valore di A compreso fra 100 e 115 e valore di B compreso fra 100 e 115 (si tratta di circa 200 ennuple)

Rispondere riempiendo la tabella alla pagina seguente, con qualche commento che spieghi la risposta:

Basi di dati II — 7 settembre 2015 — Compito B

	quad tree su A, B	B+-tree su A, B (nell'ordine)
1		
2		
3		
4		
5		
6		

Domanda 4 (15%)

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, 2>
<SETSTRING, 2, Impiegati, 33, 0, xxxx, Rossi>
<START, 1>
<SETSTRING, 1, Impiegati, 17, 0, xxxx, Neri>
<START, 3>
<COMMIT, 2>
<SETSTRING, 3, Impiegati, 33, 0, Rossi, Verdi>
<START, 4>
<SETSTRING, 4, Impiegati, 35, 0, xxxx, Bruni>
<NQCKPT, 1, 3, 4>
<SETSTRING, 4, Impiegati, 35, 0, Bruni, Neri>
<SETSTRING, 4, Impiegati, 66, 0, xxxx, Bianchi>
<START, 5>
<COMMIT, 4>
```

1. Fino a quale record del log arriva la scansione a ritroso?

2. Indicare le azioni di undo e redo che debbono essere eseguite (nell’ordine) durante un recovery di tipo undo-redo.

3. Indicare le azioni di redo che debbono essere eseguite (nell’ordine) durante un recovery di tipo redo-only.

4. È possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 24 della relazione Impiegati? Spiegare perché e in caso affermativo spiegarne le conseguenze.

5. In caso di strategia redo-only, è possibile che la transazione T_1 abbia modificato su disco il blocco 24 della relazione Impiegati? Spiegare.

6. È possibile che la transazione T_1 abbia modificato su disco il blocco 17 della relazione Impiegati? Spiegare.

Basi di dati II — 7 settembre 2015 — Compito B

Domanda 5 (15%)

Considerare un sistema distribuito su cui vengono eseguite due transazioni che coinvolgono tre nodi, un coordinatore C (lo stesso per entrambe le transazioni) e due partecipanti N1 e N2. Dopo la richiesta del coordinatore C di **prepare** (abbreviata con **prep**) per la prima transazione, i due partecipanti ricevono e rispondono correttamente, e uno dei due, N2, va in crash subito dopo aver risposto. Il coordinatore riceve le risposte, prende la decisione, invia i relativi messaggi (questi passi **non** sono stati indicati sotto e vanno quindi scritti) e subito dopo va anch'esso in crash (quindi senza fare in tempo a ricevere le conferme). Viceversa, per la seconda transazione, il coordinatore invia il messaggio di **prepare** ma non fa in tempo a ricevere le risposte. Indicare, nello schema sottostante, una possibile sequenza di scritture sui log e invio di messaggi (che includa anche i passi sopra illustrati), supponendo che entrambi i nodi siano ripristinati abbastanza presto (ma che il coordinatore perda alcuni messaggi di risposta inviati ad esso a seguito del commit). Per i messaggi si usi la notazione *tipo(transaz)→destinatari* (come nell'esempio: **prepare**(T1)→N1,N2). Supporre che nel log del coordinatore si scrivano solo i record di **prepare**, **commit** e **complete**, con i messaggi gestiti invece in memoria. Indicare ragionevoli istanti per i timeout, che permettano di concludere entrambe le transazioni.

Nodo C		Nodo N1		Nodo N2	
Log	Messaggi	Log	Messaggi	Log	Messaggi
prep(T1,N1,N2)	prep(T1)→N1,N2				<i>crash</i>
prep(T2,N1,N2)	prep(T2)→N1,N2				
	<i>crash</i>				
	<i>restart</i>				
					<i>restart</i>

Basi di dati II — 7 settembre 2015 — Compito B

Domanda 6 (10%) Si consideri uno schema dimensionale utilizzato per analizzare le vendite in una catena di supermercati che, fra le dimensioni, ne preveda una sui negozi, come la seguente:

<u>KNegozio</u>	Nome	Città	Provincia	Regione
101	Pane buono	Golfo Aranci	SS	Sardegna
102	Bontà	Golfo Aranci	SS	Sardegna
103	Pane e vino	Budoni	NU	Sardegna
104	Pane Sardo	Nuoro	NU	Sardegna
105	Pasta e pane	Palermo	PA	Sicilia
...

Si supponga ora che si presenti la seguente esigenza di modifica:

- sia pure molto raramente, le province cambiano; specificamente, si supponga che interessi gestire la modifica delle province della Sardegna avvenuta nel 2005 (ad esempio, da tale data il comune di Golfo Aranci e quello di Budoni appartengono alla provincia OT, Olbia-Tempio); in questo caso, si supponga che non interessi tanto la correlazione fra data della vendita e provincia nel momento della vendita, quanto la possibilità di fare analisi con riferimento alle due versioni del territorio, quella con le vecchie province e quella con le nuove (N.B. supporre che non interessino altre versioni oltre a queste due).

Modificare la dimensione (mostrando la nuova versione per la tabella in figura, con brevi commenti se necessario).



Basi di dati II — 7 settembre 2015 — Compito B

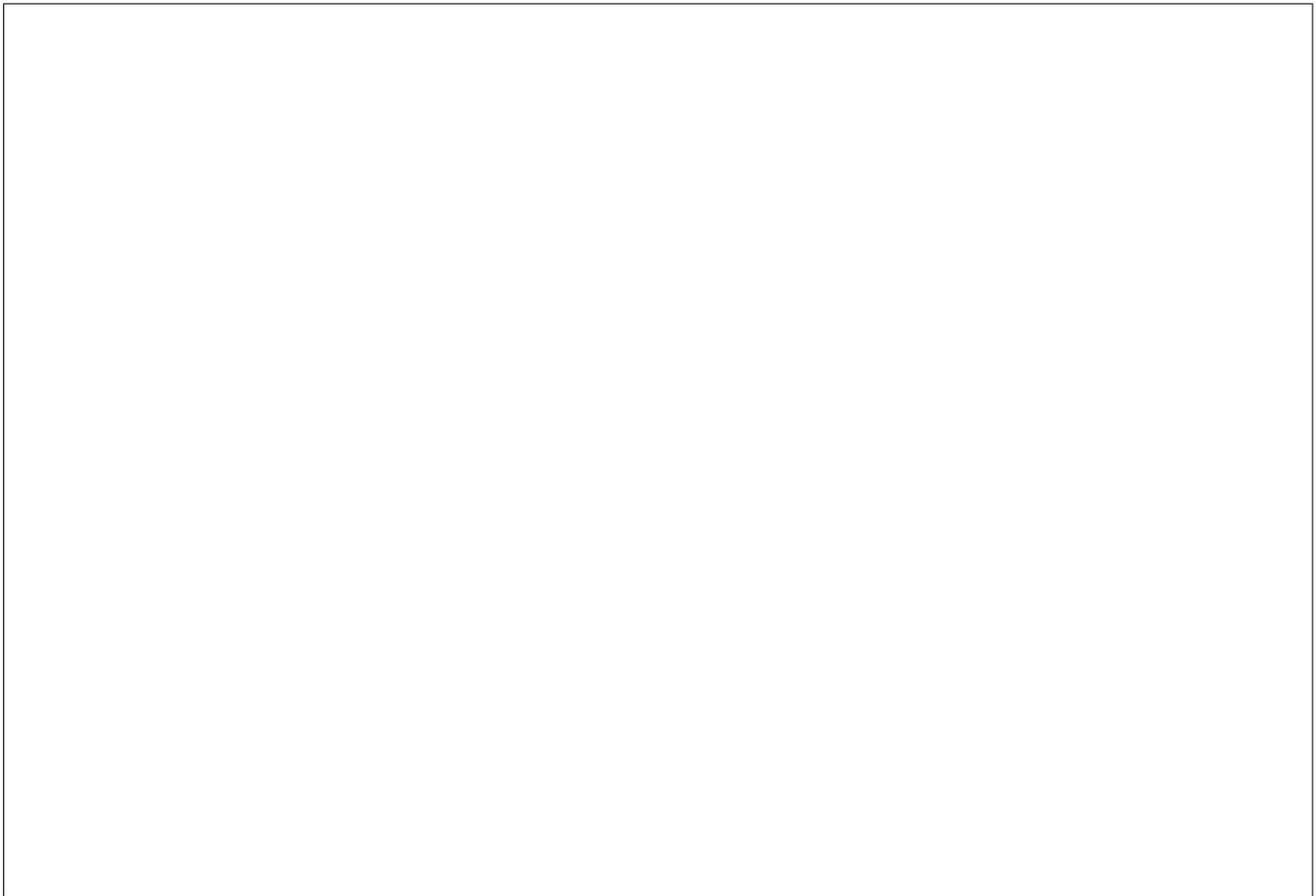
Domanda 7 (10%) Con riferimento allo schema dimensionale citato nella domanda precedente si supponga che la seguente sia la struttura della tabella dei fatti, con alcune delle ennuple:

<u>KNegozio</u>	<u>KProdotto</u>	<u>KData</u>	Quantità	Incasso
101	201	301	243	2200,00
101	202	301	4	32,00
101	202	302	6	49,00
102	201	301	22	262,00
...

Si supponga ora che:

- per ogni negozio, interessa rappresentare anche il direttore, per svolgere analisi sulle vendite ascrivibili al direttore stesso; i direttori cambiano nel tempo e passano da un negozio all'altro (e possono anche dirigere due negozi nello stesso momento; ma ogni negozio ha, in un certo giorno, un solo direttore); è disponibile l'informazione relativa ai direttori dei negozi nel tempo (per tutto il periodo, anche passato, di interesse).

Modificare la tabella dei fatti (discutendo in particolare come si potrebbero aggiornare le sue ennuple e osservando se e di quanto varia la sua cardinalità; mostrare anche la nuova versione della tabella in figura).



Basi di dati II — Esame — 7 settembre 2015 — Compito A

Cenni sulle soluzioni (solo Compito A, le varianti del testo sono in rosso)

Tempo a disposizione: due ore e trenta minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%) Si consideri una base di dati sulle seguenti relazioni, ognuna delle quali ha una struttura heap e un indice secondario sulla chiave (si suppongano le relazioni tutte molto grandi, con R_1 più piccola delle altre due)

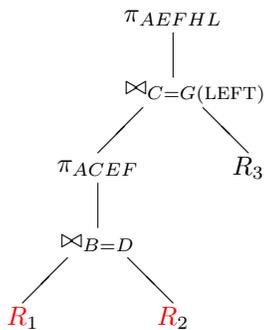
- $R_1(\underline{A}BC)$
- $R_2(\underline{D}EF)$
- $R_3(\underline{G}HL)$

e con una vista definita come segue:

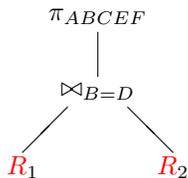
- `CREATE VIEW V AS SELECT * FROM (R1 JOIN R2 ON B=D) LEFT JOIN R3 ON C=G`

In tale contesto, supponendo che il sistema esegua join solo con nested loop (senza trarre profitto significativo dal buffer), utilizzando gli indici ove definiti, mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni, quindi indicando per il come viene eseguito) per ciascuna delle seguenti interrogazioni

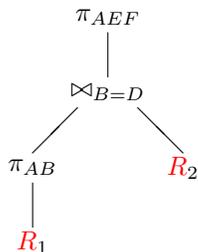
SELECT A, E, F, H, L FROM V



SELECT A, B, C, E, F FROM V



SELECT A, E, F FROM V



Nei join è indicata a sinistra la tabella esterna e viene eseguito un accesso diretto su quella interna.

Domanda 2 (15%) Una catena di supermercati ha una base di dati dei propri clienti che dispongono di una “tessera fedeltà,” con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l’uso della tessera e (b) il negozio della catena cui fa riferimento (quello presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l’interrogazione che calcola, per ciascun negozio, il numero dei clienti, la somma dei punti dei clienti e la relativa media per cliente. Indicare quale livello di isolamento (READ UNCOMMITTED, ... si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sballati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L’operazione è eseguita **mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti)**, con la finalità di **acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi**.
2. L’operazione è eseguita **mentre vengono inseriti molti nuovi clienti**, con la finalità di **acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi**.
3. L’operazione è eseguita **mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti)**, con la finalità di **individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti**.
4. L’operazione è eseguita **mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi)**, con la finalità di **individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti**.
5. L’operazione è eseguita **in un momento in cui non ci sono aggiornamenti di alcun genere**, con la finalità di **acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi**.

Risposte				
1.	2.	3.	4.	5.
RC	S	S	RR	RU

Domanda 3 (15%) Un *quad tree*, nella sua versione più semplice, è un indice definito con riferimento a due diversi attributi di una relazione. Siano A e B gli attributi in questione. La struttura è costituita da un albero in cui ogni nodo intermedio ha due etichette, una per ciascuno degli attributi, e quattro sottoalberi. Se i valori di un nodo sono (a, b) , allora

- il primo sottoalbero contiene riferimenti a record con valori di A minori di a e valori di B minori di b
- il secondo a record con valori di A minori di a e valori di B maggiori o uguali a b
- il terzo a record con valori di A maggiori o uguali ad a e valori di B minori di b
- il quarto a record con valori di A maggiori o uguali ad a e valori di B maggiori o uguali a b

Intuitivamente, ogni nodo divide lo spazio in quattro parti e quindi una foglia di un *quad tree* contiene riferimenti corrispondenti ad un “quadrato” dello spazio bidimensionale, con valori di A compresi in un intervallo $[a_1, a_2]$ e quelli di B in un intervallo $[b_1, b_2]$; in contrasto, una foglia di un $B+$ -tree definito su A, B (nell’ordine) contiene riferimenti “linearizzati,” cioè ordinati per A e, a parità di A , per B .

Consideriamo ora una relazione di $N = 4.000.000$ di ennuple, con fattore di blocco $F = 10$, in cui i due attributi A e B , insieme, formino una chiave e in cui vi siano circa $v = 2.000$ valori diversi per ciascun attributo (abbastanza densi, ad esempio valori numerici compresi fra 1 e 2.100 e distribuiti omogeneamente) e che ciascun valore di A sia associato a circa 2.000 valori diversi di B e viceversa, con una varianza abbastanza piccola. Supponiamo che le foglie contengano circa 100 riferimenti ciascuna (sia nel caso del *quad tree* sia in quello del $B+$ -tree) e di avere un sistema che permetta di caricare nel buffer tutti i livelli intermedi (ma non le foglie) sia per i *quad tree* sia per i $B+$ -tree. Si noti che in questo caso “quadrato” sopra citato contiene riferimenti con (circa) 10 valori di A e 10 di B : una foglia intermedia avrà ad esempio valori fra 50 e 60 per A e 120 e 130 per B .

Calcolare il costo (cioè il numero di foglie da visitare e, se serve, dei record del file) delle seguenti operazioni nel caso in cui sulla relazione (disordinata) è definito un *quad tree* e in quello in cui è definito un $B+$ -tree su A, B :

1. conteggio delle ennuple con un dato valore di B
2. accesso alle ennuple con un dato valore di A
3. conteggio delle ennuple con un dato valore di A
4. conteggio delle ennuple con valore di B compreso fra 100 e 115 (si tratta di circa 30.000 ennuple)
5. conteggio delle ennuple con valore di A compreso fra 100 e 115 (si tratta di circa 30.000 ennuple)
6. conteggio delle ennuple con valore di A compreso fra 100 e 115 e valore di B compreso fra 100 e 115 (si tratta di circa 200 ennuple)

Rispondere riempiendo la tabella alla pagina seguente, con qualche commento che spieghi la risposta:

Basi di dati II — 7 settembre 2015 — Compito A

Commento sulla soluzione: tutto dipende dall'organizzazione ordinata su A nel B+-tree vs "quadrato" nel quad tree. Data l'ipotesi sui buffer, è sufficiente contare le foglie coinvolte in ciascuna operazione (per semplicità assumiamo un riempimento pieno delle foglie, per entrambi i tipi di albero)

	quad tree su A, B	B+-tree su A, B (nell'ordine)
conteggio puntuale su A	ca. $v/10 = 2000/10 = 200$ (perché in ogni foglia ci sono riferimenti a 10 valori di A e 10 valori di B)	ca. $v/100 = 2000/100 = 20$ (perché in ogni foglia ci sono riferimenti a 1 valore di A e 100 valori di B)
accesso puntuale su A	ca. $200 + 2000$ (come sopra, più l'accesso ai record)	ca. $20 + 2000$ (come sopra, più l'accesso ai record)
conteggio puntuale su B	ca. $2000/10 = 200$	ca. 2000 (uno per ciascun valore di A)
conteggio in intervallo su A	ca. $30.000/100 \times 10 = 3000$ (per ogni valore di A ci sono, nello stesso blocco, i riferimenti ai record con valori "vicini" di B)	ca. $30.000/100 = 300$ (i riferimenti sono compattati in blocchi ad essi dedicati)
conteggio in intervallo su B	ca. $30.000/100 \times 10 = 3000$ (la struttura è simmetrica))	ca. 30.000 (i riferimenti sono sparpagliati, perché compattati rispetto ad A e quindi è necessario accedere ad una foglia per ogni record)
conteggio in intervallo su A e B	ca. 4-9 (in sostanza foglie adiacenti a formare un "quadrato")	ca. 15-30 (una o due foglie per ogni valore di A , poi i record sono consecutivi)

Domanda 4 (15%)

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, 2>
<SETSTRING, 2, Impiegati, 33, 0, xxxx, Rossi>
<START, 1>
<SETSTRING, 1, Impiegati, 13, 0, xxxx, Neri>
<START, 3>
<COMMIT, 2>
<SETSTRING, 3, Impiegati, 33, 0, Rossi, Verdi>
<START, 4>
<SETSTRING, 4, Impiegati, 35, 0, xxxx, Bruni>
<NQCKPT, 1, 3, 4>
<SETSTRING, 4, Impiegati, 35, 0, Bruni, Neri>
<SETSTRING, 4, Impiegati, 66, 0, xxxx, Bianchi>
<START, 5>
<COMMIT, 4>
```

1. Fino a quale record del log arriva la scansione a ritroso?

<START, 1>

2. Indicare le azioni di undo e redo che debbono essere eseguite (nell’ordine) durante un recovery di tipo undo-redo.

undo(SETSTRING,3,...), undo(SETSTRING,1,...), redo(SETSTRING, 4, ...,35, Bruni), redo(SETSTRING, 4, ...,35, Neri), redo(SETSTRING, 4, ...,35, Bianchi)

In effetti, la prima delle redo non serve, se il checkpoint, come succede di solito (nelle strategie diverse dalla redo-only), salva tutte le pagine sporche del buffer

3. Indicare le azioni di redo che debbono essere eseguite (nell’ordine) durante un recovery di tipo redo-only.

redo(SETSTRING, 4, ...,35, Bruni), redo(SETSTRING, 4, ...,35, Neri), redo(SETSTRING, 4, ...,35, Bianchi)

In questo caso serve anche la prima redo, perché al checkpoint si possono salvare solo le pagine sporche modificate da transazioni andate in commit

4. È possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 28 della relazione Impiegati? Spiegare perché e in caso affermativo spiegarne le conseguenze.

Sì, perché la modifica può essere avvenuta con il record corrispondente pure scritto nel log, ma senza che il log sia stato scritto su disco. Nessuna conseguenza.

5. In caso di strategia redo-only, è possibile che la transazione T_1 abbia modificato su disco il blocco 28 della relazione Impiegati? Spiegare.

No (con nessuna strategia), perché la regola “write-ahead log” richiede che la modifica su disco venga fatta solo dopo la scrittura su disco del corrispondente record di log

6. È possibile che la transazione T_1 abbia modificato su disco il blocco 13 della relazione Impiegati? Spiegare.

Nel caso di strategia redo-only, non può essere avvenuta. Negli altri casi, di regola è avvenuta (almeno al checkpoint)

Domanda 5 (15%)

Considerare un sistema distribuito su cui vengono eseguite due transazioni che coinvolgono tre nodi, un coordinatore **M** (lo stesso per entrambe le transazioni) e due partecipanti **R1** e **R2**. Dopo la richiesta del coordinatore **M** di **prepare** (abbreviata con **prep**) per la prima transazione, i due partecipanti ricevono e rispondono correttamente, e uno dei due, **R2**, va in crash subito dopo aver risposto. Il coordinatore riceve le risposte, prende la decisione, invia i relativi messaggi (questi passi **non** sono stati indicati sotto e vanno quindi scritti) e subito dopo va anch'esso in crash (quindi senza fare in tempo a ricevere le conferme). Viceversa, per la seconda transazione, il coordinatore invia il messaggio di **prepare** ma non fa in tempo a ricevere le risposte. Indicare, nello schema sottostante, una possibile sequenza di scritture sui log e invio di messaggi (che includa anche i passi sopra illustrati), supponendo che entrambi i nodi siano ripristinati abbastanza presto (ma che il coordinatore perda alcuni messaggi di risposta inviati ad esso a seguito del commit). Per i messaggi si usi la notazione *tipo(transaz)→destinatari* (come nell'esempio: **prepare(T1)→R1,R2**). Supporre che nel log del coordinatore si scrivano solo i record di **prepare**, **commit** e **complete**, con i messaggi gestiti invece in memoria. Indicare ragionevoli istanti per i timeout, che permettano di concludere entrambe le transazioni.

Nodo M		Nodo R1		Nodo R2	
Log	Messaggi	Log	Messaggi	Log	Messaggi
prep(T1, R1 , R2)	prep(T1)→ R1,R2	ready(T1)	ready(T1)→ M	ready(T1)	ready(T1)→ M
commit(T1)	commit(T1)→ R1,R2	commit(T1)	ack(T1)→ M	<i>crash</i>	
prep(T2, R1 , R2)	prep(T2)→ R1,R2	ready(T2)	ready(T2)→ M		
	<i>crash</i>				
	<i>restart</i>				
abort(T2)	commit(T1)→ R1,R2 abort(T2)→ R1,R2	abort(T2)	ack(T1)→ M ack(T2)→ M		
complete(T1)	commit(T1)→ R1 abort(T2)→ R1			commit(T1)	ack(T1)→ M
complete(T2)				abort(T2)	ack(T2)→ M

Basi di dati II — 7 settembre 2015 — Compito A

Domanda 6 (10%) Si consideri uno schema dimensionale utilizzato per analizzare le vendite in una catena di supermercati che, fra le dimensioni, ne preveda una sui negozi, come la seguente:

<u>KNegozio</u>	Nome	Città	Provincia	Regione
101	Pane buono	Arzachena	SS	Sardegna
102	Bontà	Arzachena	SS	Sardegna
103	Pane e vino	S. Teodoro	NU	Sardegna
104	Pane Sardo	Nuoro	NU	Sardegna
105	Pasta e pane	Palermo	PA	Sicilia
...

Si supponga ora che si presenti la seguente esigenza di modifica:

- sia pure molto raramente, le province cambiano; specificamente, si supponga che interessi gestire la modifica delle province della Sardegna avvenuta nel 2005 (ad esempio, da tale data il comune di **Arzachena** e quello di **S. Teodoro** appartengono alla provincia OT, Olbia-Tempio); in questo caso, si supponga che non interessi tanto la correlazione fra data della vendita e provincia nel momento della vendita, quanto la possibilità di fare analisi con riferimento alle due versioni del territorio, quella con le vecchie province e quella con le nuove (N.B. supporre che non interessino altre versioni oltre a queste due).

Modificare la dimensione (mostrando la nuova versione per la tabella in figura, con brevi commenti se necessario).

La nuova tabella dimensione

<u>KNeg</u>	Nome	Città	ProvVecchia	ProvNuova	Regione
101	Pane buono	Arzachena	SS	OT	Sardegna
102	Bontà	Arzachena	SS	OT	Sardegna
103	Pane e vino	S. Teodoro	NU	OT	Sardegna
104	Pane Sardo	Nuoro	NU	NU	Sardegna
105	Pasta e pane	Palermo	PA	PA	Sicilia
...

Basi di dati II — 7 settembre 2015 — Compito A

Domanda 7 (10%) Con riferimento allo schema dimensionale citato nella domanda precedente si supponga che la seguente sia la struttura della tabella dei fatti, con alcune delle ennuple:

<u>KNegozio</u>	<u>KProdotto</u>	<u>KData</u>	Quantità	Incasso
101	201	301	243	700,50
101	202	301	4	32,00
101	202	302	6	49,00
102	201	301	22	262,00
...

Si supponga ora che:

- per ogni negozio, interessi rappresentare anche il direttore, per svolgere analisi sulle vendite ascrivibili al direttore stesso; i direttori cambiano nel tempo e passano da un negozio all'altro (e possono anche dirigere due negozi nello stesso momento; ma ogni negozio ha, in un certo giorno, un solo direttore); è disponibile l'informazione relativa ai direttori dei negozi nel tempo (per tutto il periodo, anche passato, di interesse).

Modificare la tabella dei fatti (discutendo in particolare come si potrebbero aggiornare le sue ennuple e osservando se e di quanto varia la sua cardinalità; mostrare anche la nuova versione della tabella in figura).

La nuova tabella fatti ha la stessa cardinalità di quella originaria e un attributo in più, che corrisponde alla dimensione degenera Direttore, che dipende funzionalmente dalle altre, in particolare da Negozio e Data. I valori del nuovo attributo sono quindi disponibili, come scritto nel testo: “è disponibile l'informazione relativa ai direttori dei negozi nel tempo (per tutto il periodo, anche passato, al quale è relativo il data mart)”. Ad esempio, il direttore con KDirettore 501 era direttore del negozio 101 nella data 301.

<u>KNegozio</u>	<u>KProdotto</u>	<u>KData</u>	<u>KDirettore</u>	Quantità	Incasso
101	201	301	501	243	700,50
101	202	301	501	4	32,00
101	202	302	502	6	49,00
102	201	301	503	22	262,00
...