Basi di dati II

Esame — 4 luglio 2012 — Compito A

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____ **Ordin.** _____

Domanda 1 (15% per la prova lunga, 30% per la prova breve)
Considerare la relazione sotto schematizzata, definita su vari attributi, uno dei quali è la chiave, i cui valori sono mostrati. Supponendo una disponibilità di buffer abbastanza ampia, ma non sufficiente a caricare in memoria l'intera relazione (supporre ad esempio una disponibilità di 5 buffer, con un fattore di blocco pari a 3 e quindi uno spazio occupato dalla relazione pari a 9 blocchi), considerare l'esecuzione di un mergesort a più vie (e due passate) sulla relazione e mostrare lo stato delle strutture in memoria centrale e secondaria dopo l'esecuzione di cinque chiamate al metodo `next()` sullo scan che implementa il mergesort. In particolare, mostrare i "run" (cioè le porzioni di file ordinate durante prima passata) memorizzati su disco e i buffer in memoria centrale, evidenziando per ciascun buffer il record corrente. Mostrare anche i record prodotti dalle prime cinque chiamate di `next()`.

111	...
421	...
722	...
211	...
521	...
322	...
942	...
871	...
601	...
124	...
406	...
515	...
125	...
635	...
946	...
855	...
705	...
126	...
308	...
219	...
529	...
138	...
447	...
848	...
717	...
637	...

Run su disco

Buffer

Record prodotti dalle prime 5 `next()`

Domanda 2 (10% per la prova lunga, 20% per la prova breve)
Considerare il documento XML qui sotto e lo schema XSD nella pagina successiva. Individuare gli errori nella definizione dello schema che rendono non valido il documento rispetto allo schema, indicando possibili correzioni allo schema.

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="films">
       <xsd:complexType>
           <xsd:sequence minOccurs="0" maxOccurs="unbounded">
               <xsd:element ref="film"/>
           </xsd:sequence>
       </xsd:complexType>
   </xsd:element>

   <xsd:element name="film">
       <xsd:sequence>
           <xsd:element name="titolo" type="xsd:string"/>
           <xsd:element name="anno" type="xsd:int"/>
           <xsd:element name="durata" type="xsd:int"/>
           <xsd:element name="voto" type="xsd:string"/>
           <xsd:element name="generi" type="xsd:string"/>
           <xsd:element ref="staff"/>
       </xsd:sequence>
   </xsd:element>

   <xsd:element name="staff">
       <xsd:complexType>
           <xsd:choice>
               <xsd:element name="regista" type="xsd:string"/>
               <xsd:element name="sceneggiatore" type="xsd:string"/>
               <xsd:element name="attori">
                   <xsd:complexType>
                       <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                           <xsd:element name="attore" type="xsd:string"/>
                       </xsd:sequence>
                   </xsd:complexType>
               </xsd:element>
           </xsd:choice>
       </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Domanda 3 (25% per la prova lunga, 50% per la prova breve)
Con riferimento a documenti come quello mostrato in precedenza (supponendolo memorizzato nel file `esame.xml`), e ripetuto qui sotto per comodità, rispondere alle interrogazioni mostrate nella pagina successiva

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

Si ricorda la disponibilità delle seguenti funzioni:

- `count()` che restituisce il numero di nodi dell'argomento
- `contains()` che ha due argomenti e verifica se la stringa del primo argomento contiene la stringa del secondo argomento

(a) Con XPath, trovare i film che hanno stesso regista e sceneggiatore (il regista è uguale allo sceneggiatore)

(b) Con XPath, trovare il numero di film presenti nel documento che abbiano “Animazione” come uno dei generi

(c) Con XQuery, stampare i film del 2011 raggruppati per voto

(d) Con XQuery, stampare coppie di film con lo stesso regista

Domanda 4 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
x = x + 10	
write(x)	
	x = x + 20
	write(x)
commit	
	commit

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

SERIALIZABLE	READ COMMITTED

Domanda 5 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
	x = x + 20
	write(x)
	commit
read(x)	
commit	

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su multiversioni e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) multiversioni preveda

- SERIALIZABLE: le letture fanno riferimento allo stato della base di dati all'inizio della transazione e le scritture di una transazione T sono soggette ad un lock a due fasi stretto (solo per le scritture) e sono ammesse solo se il dato non è stato modificato, dopo l'inizio di T, da altre transazioni.
- READ COMMITTED: le letture fanno riferimento allo stato della base di dati all'inizio della specifica lettura e le scritture sono soggette ad un lock a due fasi stretto (solo per le scritture).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

READ COMMITTED	SERIALIZABLE

Domanda 6 (20%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria, costituita, in entrambi i casi, da valori interi positivi consecutivi)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres (ma il comportamento su altri sistemi è probabilmente lo stesso), si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where A<500000`
		Hash join

3.	`select A, B, C`
	`from R1 join R2 on C=D`
	`where A<25`
		Nested loop join

Motivare ciò, valutando, per ciascuna delle tre interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop, e supponendo che

- le relazioni abbiano rispettivamente L_1=1.000.000 ed L_2=2.000.000 ennuple, (con fattore di blocco rispettivamente f_1=10 e f_2=20)
- gli indici abbiano entrambi i=4 livelli (contando anche radice e foglie) e il fattore di blocco massimo dell'indice sia, in entrambi i casi f_i=90
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=400.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico (considerare, ovviamente, anche l'eventuale selezione)

	Hash join	Nested loop join
1.		
2.		
3.		

Domanda 7 (10%)
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, 12, 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. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo undo-redo?

3. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo redo-only?

4. È possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 37 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 37 della relazione Impiegati? Spiegare.

6. è possibile che la transazione T_1 abbia modificato su disco il blocco 12 della relazione Impiegati? Spiegare.

Basi di dati II

Esame — 4 luglio 2012 — Compito B

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____ **Ordin.** _____

Domanda 1 (15% per la prova lunga, 30% per la prova breve)
Considerare la relazione sotto schematizzata, definita su vari attributi, uno dei quali è la chiave, i cui valori sono mostrati. Supponendo una disponibilità di buffer abbastanza ampia, ma non sufficiente a caricare in memoria l'intera relazione (supporre ad esempio una disponibilità di 5 buffer, con un fattore di blocco pari a 3 e quindi uno spazio occupato dalla relazione pari a 9 blocchi), considerare l'esecuzione di un mergesort a più vie (e due passate) sulla relazione e mostrare lo stato delle strutture in memoria centrale e secondaria dopo l'esecuzione di cinque chiamate al metodo `next()` sullo scan che implementa il mergesort. In particolare, mostrare i "run" (cioè le porzioni di file ordinate durante prima passata) memorizzati su disco e i buffer in memoria centrale, evidenziando per ciascun buffer il record corrente. Mostrare anche i record prodotti dalle prime cinque chiamate di `next()`.

101	...
411	...
712	...
201	...
511	...
312	...
942	...
871	...
601	...
124	...
406	...
515	...
125	...
635	...
946	...
855	...
705	...
126	...
308	...
219	...
529	...
138	...
447	...
848	...
717	...
637	...

Run su disco

Buffer

Record prodotti dalle prime 5 `next()`

Domanda 2 (10% per la prova lunga, 20% per la prova breve)
Considerare il documento XML qui sotto e lo schema XSD nella pagina successiva. Individuare gli errori nella definizione dello schema che rendono non valido il documento rispetto allo schema, indicando possibili correzioni allo schema.

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="films">
       <xsd:complexType>
           <xsd:sequence minOccurs="0" maxOccurs="unbounded">
               <xsd:element ref="film"/>
           </xsd:sequence>
       </xsd:complexType>
   </xsd:element>

   <xsd:element name="film">
       <xsd:sequence>
           <xsd:element name="titolo" type="xsd:string"/>
           <xsd:element name="anno" type="xsd:int"/>
           <xsd:element name="durata" type="xsd:int"/>
           <xsd:element name="voto" type="xsd:string"/>
           <xsd:element name="generi" type="xsd:string"/>
           <xsd:element ref="staff"/>
       </xsd:sequence>
   </xsd:element>

   <xsd:element name="staff">
       <xsd:complexType>
           <xsd:choice>
               <xsd:element name="regista" type="xsd:string"/>
               <xsd:element name="sceneggiatore" type="xsd:string"/>
               <xsd:element name="attori">
                   <xsd:complexType>
                       <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                           <xsd:element name="attore" type="xsd:string"/>
                       </xsd:sequence>
                   </xsd:complexType>
               </xsd:element>
           </xsd:choice>
       </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Domanda 3 (25% per la prova lunga, 50% per la prova breve)
Con riferimento a documenti come quello mostrato in precedenza (supponendolo memorizzato nel file `esame.xml`), e ripetuto qui sotto per comodità, rispondere alle interrogazioni mostrate nella pagina successiva

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

Si ricorda la disponibilità delle seguenti funzioni:

- `count()` che restituisce il numero di nodi dell'argomento
- `contains()` che ha due argomenti e verifica se la stringa del primo argomento contiene la stringa del secondo argomento

(a) Con XPath, trovare i film per i quali il regista è anche uno degli attori

(b) Con XPath, trovare il numero di film che hanno uno sceneggiatore il cui nome contiene Hayao

(c) Con XQuery , stampare i registi e per ogni regista i film dell'anno 2010

(d) Con XQuery, stampare coppie di film con lo stesso voto

Domanda 4 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
	x = x + 20
	write(x)
	commit
read(x)	
commit	

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su multiversioni e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) multiversioni preveda

- SERIALIZABLE: le letture fanno riferimento allo stato della base di dati all'inizio della transazione e le scritture di una transazione T sono soggette ad un lock a due fasi stretto (solo per le scritture) e sono ammesse solo se il dato non è stato modificato, dopo l'inizio di T, da altre transazioni.
- READ COMMITTED: le letture fanno riferimento allo stato della base di dati all'inizio della specifica lettura e le scritture sono soggette ad un lock a due fasi stretto (solo per le scritture).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

SERIALIZABLE	READ COMMITTED

Domanda 5 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
x = x + 10	
write(x)	
commit	
	x = x + 20
	write(x)
	commit

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

READ COMMITTED	SERIALIZABLE

Domanda 6 (20%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria, costituita, in entrambi i casi, da valori interi positivi consecutivi)

T1(A,B,C), T2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres (ma il comportamento su altri sistemi è probabilmente lo stesso), si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from T1 join T2 on C=D`	Hash join

2.	`select *`
	`from T1 join T2 on C=D`
	`where A<500000`
		Hash join

3.	`select A, B, C`
	`from T1 join T2 on C=D`
	`where A<25`
		Nested loop join

Motivare ciò, valutando, per ciascuna delle tre interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop, e supponendo che

- le relazioni abbiano rispettivamente N_1=2.000.000 ed N_2=1.000.000 ennuple, (con fattore di blocco rispettivamente f_1=20 e f_2=10)
- gli indici abbiano entrambi p=4 livelli (contando anche radice e foglie) e il fattore di blocco massimo dell'indice sia, in entrambi i casi f_i=90
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=400.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico (considerare, ovviamente, anche l'eventuale selezione)

	Hash join	Nested loop join
1.		
2.		
3.		

Domanda 7 (10%)
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, 37, 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. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo undo-redo?

3. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo redo-only?

4. È possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 12 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 12 della relazione Impiegati? Spiegare.

6. è possibile che la transazione T_1 abbia modificato su disco il blocco 37 della relazione Impiegati? Spiegare.

Basi di dati II

Esame — 4 luglio 2012 — Compito C

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____ **Ordin.** _____

Domanda 1 (15% per la prova lunga, 30% per la prova breve)
Considerare la relazione sotto schematizzata, definita su vari attributi, uno dei quali è la chiave, i cui valori sono mostrati. Supponendo una disponibilità di buffer abbastanza ampia, ma non sufficiente a caricare in memoria l'intera relazione (supporre ad esempio una disponibilità di 5 buffer, con un fattore di blocco pari a 3 e quindi uno spazio occupato dalla relazione pari a 9 blocchi), considerare l'esecuzione di un mergesort a più vie (e due passate) sulla relazione e mostrare lo stato delle strutture in memoria centrale e secondaria dopo l'esecuzione di cinque chiamate al metodo `next()` sullo scan che implementa il mergesort. In particolare, mostrare i "run" (cioè le porzioni di file ordinate durante prima passata) memorizzati su disco e i buffer in memoria centrale, evidenziando per ciascun buffer il record corrente. Mostrare anche i record prodotti dalle prime cinque chiamate di `next()`.

101	...
421	...
722	...
201	...
521	...
322	...
942	...
871	...
601	...
124	...
406	...
515	...
125	...
635	...
946	...
855	...
705	...
126	...
308	...
219	...
529	...
138	...
447	...
848	...
717	...
637	...

Run su disco

Buffer

Record prodotti dalle prime 5 `next()`

Domanda 2 (10% per la prova lunga, 20% per la prova breve)
Considerare il documento XML qui sotto e lo schema XSD nella pagina successiva. Individuare gli errori nella definizione dello schema che rendono non valido il documento rispetto allo schema, indicando possibili correzioni allo schema.

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="films">
       <xsd:complexType>
           <xsd:sequence minOccurs="0" maxOccurs="unbounded">
               <xsd:element ref="film"/>
           </xsd:sequence>
       </xsd:complexType>
   </xsd:element>

   <xsd:element name="film">
       <xsd:sequence>
           <xsd:element name="titolo" type="xsd:string"/>
           <xsd:element name="anno" type="xsd:int"/>
           <xsd:element name="durata" type="xsd:int"/>
           <xsd:element name="voto" type="xsd:string"/>
           <xsd:element name="generi" type="xsd:string"/>
           <xsd:element ref="staff"/>
       </xsd:sequence>
   </xsd:element>

   <xsd:element name="staff">
       <xsd:complexType>
           <xsd:choice>
               <xsd:element name="regista" type="xsd:string"/>
               <xsd:element name="sceneggiatore" type="xsd:string"/>
               <xsd:element name="attori">
                   <xsd:complexType>
                       <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                           <xsd:element name="attore" type="xsd:string"/>
                       </xsd:sequence>
                   </xsd:complexType>
               </xsd:element>
           </xsd:choice>
       </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Domanda 3 (25% per la prova lunga, 50% per la prova breve)
Con riferimento a documenti come quello mostrato in precedenza (supponendolo memorizzato nel file `esame.xml`), e ripetuto qui sotto per comodità, rispondere alle interrogazioni mostrate nella pagina successiva

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

Si ricorda la disponibilità delle seguenti funzioni:

- `count()` che restituisce il numero di nodi dell'argomento
- `contains()` che ha due argomenti e verifica se la stringa del primo argomento contiene la stringa del secondo argomento

(a) Con XPath, trovare i film che hanno stesso regista e sceneggiatore (il regista è uguale allo sceneggiatore)

(b) Con XPath, trovare il numero di film presenti nel documento che abbiano “Animazione” come uno dei generi

(c) Con XQuery, stampare i film del 2011 raggruppati per voto

(d) Con XQuery, stampare coppie di film con lo stesso regista

Domanda 4 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
x = x + 10	
write(x)	
	x = x + 20
	write(x)
commit	
	commit

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

SERIALIZABLE	READ COMMITTED

Domanda 5 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
	x = x + 20
	write(x)
	commit
read(x)	
commit	

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su multiversioni e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) multiversioni preveda

- SERIALIZABLE: le letture fanno riferimento allo stato della base di dati all'inizio della transazione e le scritture di una transazione T sono soggette ad un lock a due fasi stretto (solo per le scritture) e sono ammesse solo se il dato non è stato modificato, dopo l'inizio di T, da altre transazioni.
- READ COMMITTED: le letture fanno riferimento allo stato della base di dati all'inizio della specifica lettura e le scritture sono soggette ad un lock a due fasi stretto (solo per le scritture).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

READ COMMITTED	SERIALIZABLE

Domanda 6 (20%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria, costituita, in entrambi i casi, da valori interi positivi consecutivi)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres (ma il comportamento su altri sistemi è probabilmente lo stesso), si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where A<500000`
		Hash join

3.	`select A, B, C`
	`from R1 join R2 on C=D`
	`where A<25`
		Nested loop join

Motivare ciò, valutando, per ciascuna delle tre interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop, e supponendo che

- le relazioni abbiano rispettivamente L_1=1.000.000 ed L_2=2.000.000 ennuple, (con fattore di blocco rispettivamente f_1=10 e f_2=20)
- gli indici abbiano entrambi i=4 livelli (contando anche radice e foglie) e il fattore di blocco massimo dell'indice sia, in entrambi i casi f_i=90
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=400.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico (considerare, ovviamente, anche l'eventuale selezione)

	Hash join	Nested loop join
1.		
2.		
3.		

Domanda 7 (10%)
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, 19, 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. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo undo-redo?

3. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo redo-only?

4. È possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 42 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 42 della relazione Impiegati? Spiegare.

6. è possibile che la transazione T_1 abbia modificato su disco il blocco 19 della relazione Impiegati? Spiegare.

Basi di dati II

Esame — 4 luglio 2012 — Compito D

Rispondere su questo fascicolo.

Tempo a disposizione: due ore e quarantacinque minuti (prova lunga), un'ora e trenta minuti (prova breve).

Cognome _____ **Nome** _____ **Matricola** _____ **Ordin.** _____

Domanda 1 (15% per la prova lunga, 30% per la prova breve)
Considerare la relazione sotto schematizzata, definita su vari attributi, uno dei quali è la chiave, i cui valori sono mostrati. Supponendo una disponibilità di buffer abbastanza ampia, ma non sufficiente a caricare in memoria l'intera relazione (supporre ad esempio una disponibilità di 5 buffer, con un fattore di blocco pari a 3 e quindi uno spazio occupato dalla relazione pari a 9 blocchi), considerare l'esecuzione di un mergesort a più vie (e due passate) sulla relazione e mostrare lo stato delle strutture in memoria centrale e secondaria dopo l'esecuzione di cinque chiamate al metodo `next()` sullo scan che implementa il mergesort. In particolare, mostrare i "run" (cioè le porzioni di file ordinate durante prima passata) memorizzati su disco e i buffer in memoria centrale, evidenziando per ciascun buffer il record corrente. Mostrare anche i record prodotti dalle prime cinque chiamate di `next()`.

111	...
411	...
712	...
211	...
511	...
312	...
942	...
871	...
601	...
124	...
406	...
515	...
125	...
635	...
946	...
855	...
705	...
126	...
308	...
219	...
529	...
138	...
447	...
848	...
717	...
637	...

Run su disco

Buffer

Record prodotti dalle prime 5 `next()`

Domanda 2 (10% per la prova lunga, 20% per la prova breve)
Considerare il documento XML qui sotto e lo schema XSD nella pagina successiva. Individuare gli errori nella definizione dello schema che rendono non valido il documento rispetto allo schema, indicando possibili correzioni allo schema.

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="films">
       <xsd:complexType>
           <xsd:sequence minOccurs="0" maxOccurs="unbounded">
               <xsd:element ref="film"/>
           </xsd:sequence>
       </xsd:complexType>
   </xsd:element>

   <xsd:element name="film">
       <xsd:sequence>
           <xsd:element name="titolo" type="xsd:string"/>
           <xsd:element name="anno" type="xsd:int"/>
           <xsd:element name="durata" type="xsd:int"/>
           <xsd:element name="voto" type="xsd:string"/>
           <xsd:element name="generi" type="xsd:string"/>
           <xsd:element ref="staff"/>
       </xsd:sequence>
   </xsd:element>

   <xsd:element name="staff">
       <xsd:complexType>
           <xsd:choice>
               <xsd:element name="regista" type="xsd:string"/>
               <xsd:element name="sceneggiatore" type="xsd:string"/>
               <xsd:element name="attori">
                   <xsd:complexType>
                       <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                           <xsd:element name="attore" type="xsd:string"/>
                       </xsd:sequence>
                   </xsd:complexType>
               </xsd:element>
           </xsd:choice>
       </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Domanda 3 (25% per la prova lunga, 50% per la prova breve)
Con riferimento a documenti come quello mostrato in precedenza (supponendolo memorizzato nel file `esame.xml`), e ripetuto qui sotto per comodità, rispondere alle interrogazioni mostrate nella pagina successiva

```
<films>
      <film>
            <titolo>Spirited Away</titolo>
            <anno>2001</anno>
            <durata>125 min</durata>
            <voto>8.6</voto>
            <generi>Animazione, Avventura, Famiglia</generi>
            <staff>
                  <regista>Hayao Miyazaki</regista>
                  <sceneggiatore>Hayao Miyazaki</sceneggiatore>
                  <attori>
                      <attore>Daveigh Chase</attore>
                      <attore>Suzanne Pleshette</attore>
                      <attore>Miyu Irino</attore>
                  </attori>
            </staff>
      </film>
      <film>
            <titolo>Cowboys and Aliens</titolo>
            <anno>2011</anno>
            <durata>119 min</durata>
            <voto>6.2</voto>
            <generi>Azione, Fantascienza, Thriller</generi>
            <staff>
                  <regista>John Favreau</regista>
                  <attori>
                        <attore>Daniel Craig</attore>
                        <attore>Harrison Ford</attore>
                        <attore>Olivia Wilde</attore>
                  </attori>
            </staff>
      </film>
</films>
```

Si ricorda la disponibilità delle seguenti funzioni:

- `count()` che restituisce il numero di nodi dell'argomento
- `contains()` che ha due argomenti e verifica se la stringa del primo argomento contiene la stringa del secondo argomento

(a) Con XPath, trovare i film per i quali il regista è anche uno degli attori

(b) Con XPath, trovare il numero di film che hanno uno sceneggiatore il cui nome contiene Hayao

(c) Con XQuery , stampare i registi e per ogni regista i film dell'anno 2010

(d) Con XQuery, stampare coppie di film con lo stesso voto

Domanda 4 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
	x = x + 20
	write(x)
	commit
read(x)	
commit	

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su multiversioni e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) multiversioni preveda

- SERIALIZABLE: le letture fanno riferimento allo stato della base di dati all'inizio della transazione e le scritture di una transazione T sono soggette ad un lock a due fasi stretto (solo per le scritture) e sono ammesse solo se il dato non è stato modificato, dopo l'inizio di T, da altre transazioni.
- READ COMMITTED: le letture fanno riferimento allo stato della base di dati all'inizio della specifica lettura e le scritture sono soggette ad un lock a due fasi stretto (solo per le scritture).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

SERIALIZABLE	READ COMMITTED

Domanda 5 (10%)
Considerare il seguente scenario in cui due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci immediatamente la stessa transazione.

client 1	client 2
read(x)	
	read(x)
x = x + 10	
write(x)	
commit	
	x = x + 20
	write(x)
	commit

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

READ COMMITTED	SERIALIZABLE

Domanda 6 (20%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria, costituita, in entrambi i casi, da valori interi positivi consecutivi)

T1(A,B,C), T2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres (ma il comportamento su altri sistemi è probabilmente lo stesso), si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from T1 join T2 on C=D`	Hash join

2.	`select *`
	`from T1 join T2 on C=D`
	`where A<500000`
		Hash join

3.	`select A, B, C`
	`from T1 join T2 on C=D`
	`where A<25`
		Nested loop join

Motivare ciò, valutando, per ciascuna delle tre interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop, e supponendo che

- le relazioni abbiano rispettivamente N_1=2.000.000 ed N_2=1.000.000 ennuple, (con fattore di blocco rispettivamente f_1=20 e f_2=10)
- gli indici abbiano entrambi p=4 livelli (contando anche radice e foglie) e il fattore di blocco massimo dell'indice sia, in entrambi i casi f_i=90
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=400.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico (considerare, ovviamente, anche l'eventuale selezione)

	Hash join	Nested loop join
1.		
2.		
3.		

Domanda 7 (10%)
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, 42, 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. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo undo-redo?

3. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery di tipo redo-only?

4. È possibile che la transazione T_1 abbia modificato il buffer contenente il blocco 19 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 19 della relazione Impiegati? Spiegare.

6. è possibile che la transazione T_1 abbia modificato su disco il blocco 42 della relazione Impiegati? Spiegare.