

## Basi di dati II — Esame — 5 luglio 2017

Tempo a disposizione: un'ora e quindici minuti per la prova breve e due ore e trenta minuti per la prova completa.

Cognome \_\_\_\_\_ Nome \_\_\_\_\_ Matricola \_\_\_\_\_

**Domanda 1** (15% per la prova completa, 35% per la prova breve)

Si consideri un sistema con blocchi di dimensione  $B = 4$  KB e indirizzi dei record di lunghezza  $b = 8$  byte e una relazione  $R$  con  $N_R = 800.000$  ennuple, di  $c = 40$  byte ciascuna e (fra gli altri) tre attributi,  $A$ ,  $C$ ,  $E$  ciascuno con  $v = 20$  valori diversi.

1. (A) Valutare lo spazio  $SB$  (in blocchi) necessario per le foglie di un indice bitmap su  $A$ .  
 (B) Indicare quanto sarebbe invece lo spazio necessario  $SB'$  se i valori diversi di  $A$  fossero  $v' = 2000$ .

Poi, valutare (nei due casi) il costo (in termini di numero di accessi a memoria secondaria, supponendo che l'albero di accesso al bitmap sia tutto nei buffer; basta quindi contare il numero di blocchi del bitmap da visitare e l'accesso alle ennuple; per quest'ultimo si può considerare per ciascuna **un costo pari a 2 accessi**, per accedere prima alla struttura intermedia e poi al record vero e proprio) per le seguenti interrogazioni

2. `SELECT * FROM R WHERE A = 5` (supporre che il risultato contenga circa  $N_R/v$  ennuple)
3. `SELECT COUNT(*) FROM R WHERE A = 5`
4. `SELECT * FROM R WHERE A = 15 AND C = 14 AND E = 13`, assumendo che ci sia un indice bitmap su ciascuno degli attributi coinvolti (non potendo sapere quanto i valori siano indipendenti, assumere che il risultato contenga 100 ennuple nel caso (A) e 5 nel caso (B))

Riportare le risposte nella tabella sottostante, nelle righe da 1 a 4, indicando formula e valore numerico.

Risposte

	(A) $v$ valori diversi ( $v = 20$ )	(B) $v'$ valori diversi ( $v' = 2000$ )
1		
2		
3		
4		

## Basi di dati II — 5 luglio 2017

**Domanda 2** (20% per la prova completa, 45% per la prova breve)

Si consideri la seguente porzione dello schema dell'archivio delle carriere degli studenti di una anagrafe ministeriale:

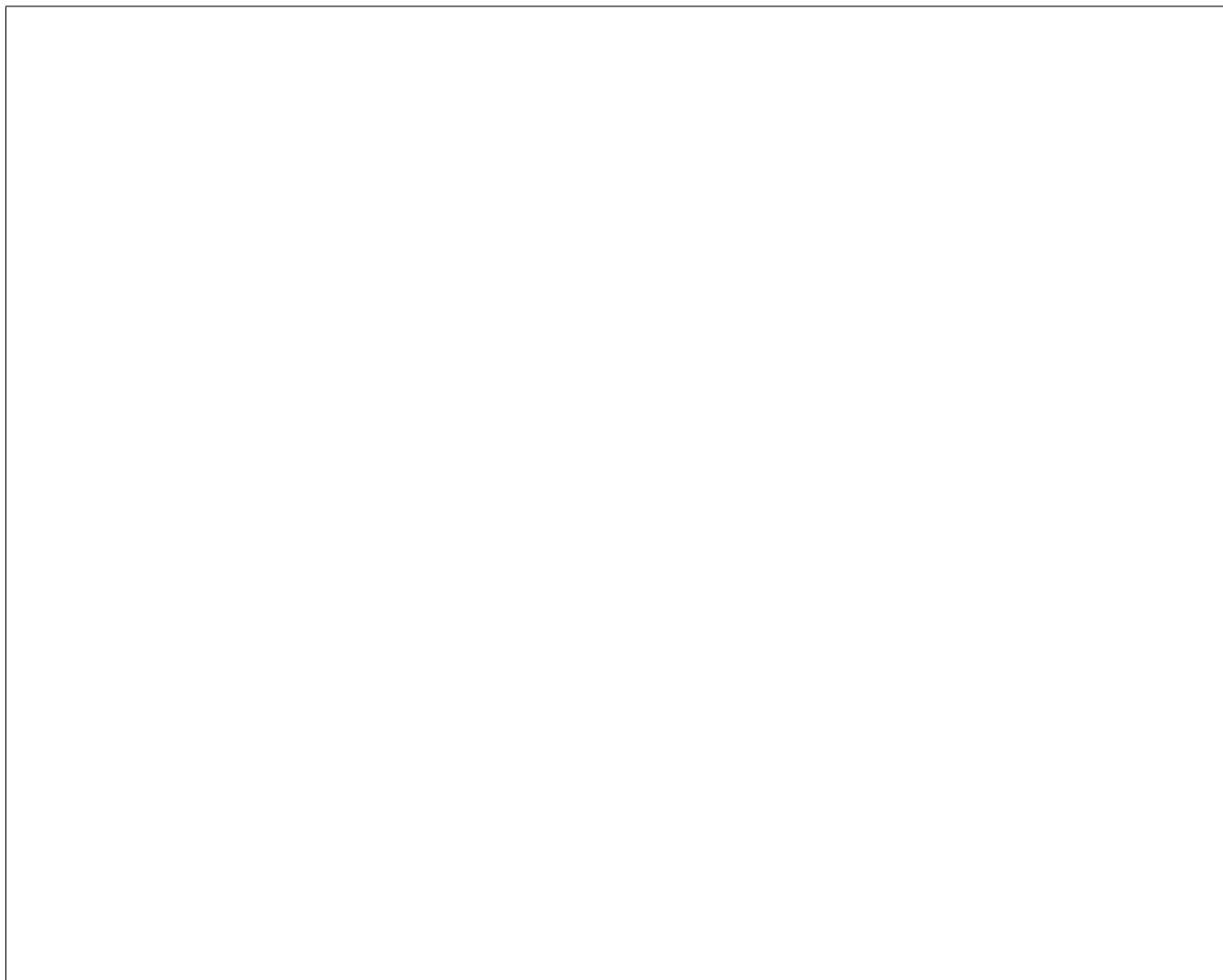
- **STUDENTI**(CodiceFiscale, Cognome, Nome, DataNascita, TipoMaturità)
- **ISCRIZIONI**(CodiceFiscale, AnnoAccademico, CorsoDiStudio, AnnoDiCorso, AnnoDiImmatricolazione)
- **CORSIDI STUDIO**(CodiceCdS, Titolo, Livello, Classe, Università)
- **LAUREE**(CodiceFiscale, CodiceCdS, Data, Voto)

Progettare uno schema dimensionale che permetta di rispondere, fra le altre, alle seguenti interrogazioni:

- calcolare il numero di studenti (con la relativa media dei voti) che si sono laureati in un certo corso di studio (inteso come corso di studio presso una università) in un certo anno accademico (si supponga che, per la data di laurea, l'unico dettaglio rilevante sia l'anno accademico e che esista un modo univoco per associare un anno accademico ad una data di laurea)
- calcolare il numero di laureati per una classe di corsi studio, distinto per tipo di maturità e per numero di anni impiegati per conseguire il titolo (ad esempio, 2, 3, 4, 5, 6, più di 6)
- calcolare il numero di laureati per una classe di corsi studio, distinto per "età alla laurea" (ad esempio, 21, 22, 23, ...26, più di 26)

Assumere che, per ragioni di privatezza e di compattezza, sia opportuno limitare la cardinalità della tabella dei fatti, a patto di permettere la risposta alle precedenti interrogazioni.

Mostrare lo schema dimensionale, specificando la grana scelta.



**Basi di dati II — 5 luglio 2017**

Descrivere, informalmente, ma in modo strutturato e comprensibile, il processo di ETL che porta alla tabella dei fatti mostrata in risposta alla domanda precedente

## Basi di dati II — 5 luglio 2017

**Domanda 3** (10% per la prova completa, 20% per la prova breve)

Si consideri uno schema dimensionale con la seguente tabella dei fatti (in cui il significato delle dimensioni e misure è intuitivo):

- FattiVendite(KProdotto, KNegozio, KMese, Ricavo)

Con riferimento a tale schema, si supponga che

- la tabella sia “densa” (cioè che esista una ennupla per ciascun mese, negozio e prodotto) e che i prodotti diversi siano  $p = 1000$ , i mesi di interesse  $m = 36$  (tre anni) e i negozi  $n = 10$ .
- le operazioni più frequenti siano:
  1. ricavi complessivi per un prodotto (tutti i negozi) mese per mese (nei tre anni), con frequenza  $f_1 = 100$
  2. ricavi complessivi (tutti i prodotti e tutti i negozi) mese per mese (nei tre anni), con frequenza  $f_2 = 10$
  3. ricavi complessivi per un negozio (tutti i prodotti) mese per mese (nei tre anni), con frequenza  $f_3 = 50$
- sia possibile realizzare una sola vista materializzata.

Scegliere, fra le tre viste sostanzialmente corrispondenti alle tre interrogazioni, quella che si ritiene supporti meglio il carico applicativo.

**Domanda 4** (20%)

Considerare lo scenario a fianco in cui tre client diversi inviano richieste ad un gestore della 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 al 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 la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato). Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** (come in Postgres) e livello di isolamento **READ COMMITTED** sulle prime due transazioni e **SERIALIZABLE** sulla terza. Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia 200 e quello dell'oggetto y sia 500. Indicare, nell'ordine, le operazioni che vengono eseguite da ciascun client, specificando, per ciascuna, il valore che viene letto o scritto. In conclusione, dire se si verificano o meno anomalie.

client 1	client 2	client 3
begin read(x) x = x + 10 write(x)	begin read(y) y = y + 20 write(y) read(x) x = x - 20 write(x) commit	begin read(x)
read(y) y = y - 10 write(y) commit		<i>(dopo molto tempo)</i> read(x) commit

client 1	client 2	client 3

Si verificano anomalie?

Considerare nuovamente lo scenario della pagina precedente, ripetuto qui a fianco per comodità. Considerare uno scheduler con controllo di concorrenza basato su **2PL stretto** con livello di isolamento **SERIALIZABLE** sulle prime due transazioni e **READ COMMITTED** sulla terza. Mostrare il comportamento dello scheduler, supponendo ancora che il valore iniziale dell'oggetto x sia 200 e quello dell'oggetto y sia **500**.

client 1	client 2	client 3
begin read(x) x = x + 10 write(x)	begin read(y) y = y + 20 write(y) read(x) x = x - 20 write(x) commit	begin read(x)
read(y) y = y - 10 write(y) commit		<i>(dopo molto tempo)</i> read(x) commit

client 1	client 2	client 3

Si verificano anomalie?

**Domanda 5** (20%)

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 3 buffer, con un fattore di blocco pari a 2 e quindi uno spazio occupato dalla relazione pari a 8 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 sei 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 sei chiamate di `next()`.

	Run su disco	Buffer	Record prodotti dalle prime sei <code>next()</code>
	131 ...		
	441 ...		
	742 ...		
	231 ...		
	541 ...		
	342 ...		
	942 ...		
	871 ...		
	601 ...		
	174 ...		
	496 ...		
	575 ...		
	145 ...		
	635 ...		
	946 ...		
	855 ...		

Basi di dati II — 5 luglio 2017

**Domanda 6** (15%) Considerare un sistema distribuito su cui viene eseguita una transazione che coinvolge tre nodi, un coordinatore M e due partecipanti R1 e R2. Dopo la richiesta di **prepare** da parte del coordinatore M, i due partecipanti ricevono e rispondono correttamente, e uno dei due, R1, va in crash subito dopo aver risposto. Il coordinatore riceve le risposte, prende la decisione, invia i relativi messaggi e subito dopo va anch'esso in crash (quindi senza fare in tempo a ricevere le conferme). Indicare, nello schema sottostante, una possibile sequenza di scritture sui log e invio di messaggi, supponendo che entrambi i nodi siano ripristinati abbastanza presto (ma che il coordinatore perda i messaggi di risposta inviati ad esso a seguito del commit). Per semplicità, si fa riferimento ad una sola transazione e quindi non c'è bisogno di indicarla. Per i messaggi si usi la notazione *tipo*→*destinatari* (come nell'esempio: **prepare**→R1,R2). Supporre che nel log del coordinatore si scrivano solo i record di **prepare**, **commit** e **complete**, mentre i messaggi sono gestiti in memoria.

Nodo M		Nodo R1		Nodo R2	
Log	Messaggi	Log	Messaggi	Log	Messaggi
prepare(R1,R2)	prepare→R1,R2				
			crash		
	crash				
	restart				
			restart		

Eventuali commenti:



# Basi di dati II — Esame — 5 luglio 2017

## Cenni sulle soluzioni

Tempo a disposizione: un'ora e quindici minuti per la prova breve e due ore e trenta minuti per la prova completa.

Cognome \_\_\_\_\_ Nome \_\_\_\_\_ Matricola \_\_\_\_\_

**Domanda 1** (15% per la prova completa, 35% per la prova breve)

Si consideri un sistema con blocchi di dimensione  $B = 4$  KB e indirizzi dei record di lunghezza  $b = 8$  byte e una relazione R con  $N_R = 800.000$  ennuple, di  $c = 40$  byte ciascuna e (fra gli altri) tre attributi, A, C, E ciascuno con  $v = 20$  valori diversi.

1. (A) Valutare lo spazio SB (in blocchi) necessario per le foglie di un indice bitmap su A.  
(B) Indicare quanto sarebbe invece lo spazio necessario SB' se i valori diversi di A fossero  $v' = 2000$ .

Poi, valutare (nei due casi) il costo (in termini di numero di accessi a memoria secondaria, supponendo che l'albero di accesso al bitmap sia tutto nei buffer; basta quindi contare il numero di blocchi del bitmap da visitare e l'accesso alle ennuple; per quest'ultimo si può considerare per ciascuna **un costo pari a 2 accessi**, per accedere prima alla struttura intermedia e poi al record vero e proprio) per le seguenti interrogazioni

2. SELECT \* FROM R WHERE A = 5 (supporre che il risultato contenga circa  $N_R/v$  ennuple)
3. SELECT COUNT(\*) FROM R WHERE A = 5
4. SELECT \* FROM R WHERE A = 15 AND C = 14 AND E = 13, assumendo che ci sia un indice bitmap su ciascuno degli attributi coinvolti (non potendo sapere quanto i valori siano indipendenti, assumere che il risultato contenga 100 ennuple nel caso (A) e 5 nel caso (B))

Riportare le risposte nella tabella sottostante, nelle righe da 1 a 4, indicando formula e valore numerico.

Risposte		
	(A) $v$ valori diversi ( $v = 20$ )	(B) $v'$ valori diversi ( $v' = 2000$ )
1	$SB = \frac{v \times N_R}{8 \times B} = \frac{20 \times 800.000}{8 \times 4000} = 500$	$SB' = \frac{v' \times N_R}{8 \times B} = \frac{2000 \times 800.000}{8 \times 4000} = 50.000$
2	<p>Usando l'indice, accesso ad un vettore (i) e poi ai record del file, attraverso la struttura intermedia (ii).</p> <p>(i) <math>\frac{N_R}{8 \times B} = \frac{800.000}{8 \times 4000} = 25</math></p> <p>(ii) <math>2 \times \frac{N_R}{v} = 2 \times \frac{800.000}{20} = 80.000</math></p> <p>la componente (ii) domina. In effetti, visto che il file occupa <math>\frac{N_R \times c}{B} = \frac{800.000 \times 40}{4000} = 8000</math> blocchi, può convenire la scansione sequenziale</p>	<p>Rispetto al caso (A), cambiano solo i valori per (ii)</p> <p>(ii) <math>2 \times \frac{N_R}{v'} = 2 \times \frac{800.000}{2000} = 800</math></p>
3	<p>basta accedere ad un vettore:</p> $\frac{N_R}{8 \times B} = \frac{800.000}{8 \times 4000} = 25$	<p>Idem</p>
4	<p>Accesso ad un vettore per ciascun indice (i) e poi ai record del file, attraverso la struttura intermedia (ii).</p> <p>(i) <math>3 \times \frac{N_R}{8 \times B} = \frac{3 \times 800.000}{8 \times 4000} = 75</math></p> <p>(ii) numero dei record nel risultato (moltiplicato per due)</p> <p>Totale 275</p>	<p>Rispetto al caso (A), cambiano solo i valori (per (ii))</p> <p>Totale accessi circa 85.</p>

## Basi di dati II — 5 luglio 2017

**Domanda 2** (20% per la prova completa, 45% per la prova breve)

Si consideri la seguente porzione dello schema dell'archivio delle carriere degli studenti di una anagrafe ministeriale:

- **STUDENTI**(CodiceFiscale, Cognome, Nome, DataNascita, TipoMaturità)
- **ISCRIZIONI**(CodiceFiscale, AnnoAccademico, CorsoDiStudio, AnnoDiCorso, AnnoDiImmatricolazione)
- **CORSIDI STUDIO**(CodiceCdS, Titolo, Livello, Classe, Università)
- **LAUREE**(CodiceFiscale, CodiceCdS, Data, Voto)

Progettare uno schema dimensionale che permetta di rispondere, fra le altre, alle seguenti interrogazioni:

- calcolare il numero di studenti (con la relativa media dei voti) che si sono laureati in un certo corso di studio (inteso come corso di studio presso una università) in un certo anno accademico (si supponga che, per la data di laurea, l'unico dettaglio rilevante sia l'anno accademico e che esista un modo univoco per associare un anno accademico ad una data di laurea)
- calcolare il numero di laureati per una classe di corsi studio, distinto per tipo di maturità e per numero di anni impiegati per conseguire il titolo (ad esempio, 2, 3, 4, 5, 6, più di 6)
- calcolare il numero di laureati per una classe di corsi studio, distinto per "età alla laurea" (ad esempio, 21, 22, 23, ...26, più di 26)

Assumere che, per ragioni di privatezza e di compattezza, sia opportuno limitare la cardinalità della tabella dei fatti, a patto di permettere la risposta alle precedenti interrogazioni.

Mostrare lo schema dimensionale, specificando la grana scelta.

Grana: insieme di studenti laureati in corso di studio, in un anno accademico, con un tipo di maturità, che ha impiegato uno stesso numero di anni, con una stessa età.

Tabella dei fatti:

FATTILAUREE(KCdS, KAnnoAccademico, KTipoMaturità, KDurataStudi, KEtàAllaLaurea, NumStudenti, VotoMedio)

Dimensioni:

CORSODI STUDIO(KCdS, CodiceCdS, NomeCdS, Università, ..., Livello, CodiceClasse, NomeClasse...)

ANNOACCADEMICO(KAnnoAccademico, ...)

TIPOMATURITÀ(KTipoMaturità, ...)

DURATASTUDI(KDurataStudi, ...)

ETÀALLALAUREA(KEtàAllaLaurea, ...)

## Basi di dati II — 5 luglio 2017

Descrivere, informalmente, ma in modo strutturato e comprensibile, il processo di ETL che porta alla tabella dei fatti mostrata in risposta alla domanda precedente

- join delle relazioni STUDENTI, ISCRIZIONI, e LAUREE
- aggiunta di un attributo con l'età dello studente alla laurea e la durata degli studi (in modo coerente con le scelte fatte nelle dimensioni corrispondenti)
- aggregazione sulla base delle dimensioni, con calcolo del numeri di laureati e del voto medio
- sostituzione degli identificatori o dei valori (a seconda dei casi) con le chiavi surrogate delle dimensioni

**Domanda 3** (10% per la prova completa, 20% per la prova breve)

Si consideri uno schema dimensionale con la seguente tabella dei fatti (in cui il significato delle dimensioni e misure è intuitivo):

- FattiVendite(KProdotto, KNegozi, KMese, Ricavo)

Con riferimento a tale schema, si supponga che

- la tabella sia “densa” (cioè che esista una ennupla per ciascun mese, negozio e prodotto) e che i prodotti diversi siano  $p = 1000$ , i mesi di interesse  $m = 36$  (tre anni) e i negozi  $n = 10$ .
- le operazioni più frequenti siano:
  1. ricavi complessivi per un prodotto (tutti i negozi) mese per mese (nei tre anni), con frequenza  $f_1 = 100$
  2. ricavi complessivi (tutti i prodotti e tutti i negozi) mese per mese (nei tre anni), con frequenza  $f_2 = 10$
  3. ricavi complessivi per un negozio (tutti i prodotti) mese per mese (nei tre anni), con frequenza  $f_3 = 50$
- sia possibile realizzare una sola vista materializzata.

Scegliere, fra le tre viste sostanzialmente corrispondenti alle tre interrogazioni, quella che si ritiene supporti meglio il carico applicativo.

Le tre viste:

1. RICAVIPRODOTTOMESE(KProdotto, KMese, Ricavo), cardinalità  $p \times m = 36.000$ , supporta le interrogazioni 1 e 2
2. RICAVIMESE(KMese, Ricavo), cardinalità  $m = 36$ , supporta solo l’interrogazione 2
3. RICAVINEGOZIOMESE(KNegozi, KMese, Ricavo), cardinalità  $n \times m = 360$ , supporta le interrogazioni 2 e 3

Costo unitario per ogni operazione con ciascuna vista:

	con vista 1	con vista 2	con vista 3
$c_1$	Si deve accedere alle ennuple di un prodotto, in tutti i mesi; costo unitario: $m = 36$	la vista non aiuta, si deve accedere a tutte le ennuple del prodotto nella relazione base: $m \times n = 360$	la vista non aiuta, si deve accedere a tutte le ennuple del prodotto nella relazione base: $m \times n = 360$
$c_2$	Si deve accedere a tutte le ennuple della vista: $p \times m = 36.000$	Si deve accedere a tutte le ennuple della vista: $m = 36$	Si deve accedere a tutte le ennuple della vista: $n \times m = 360$
$c_3$	La vista non aiuta, si deve accedere alle ennuple di un negozio, in tutti i mesi; costo unitario: $p \times m = 36.000$	la vista non aiuta, si deve accedere a tutte le ennuple del negozio nella relazione base: $m \times p = 36.000$	Si deve accedere alle ennuple di un negozio, in tutti i mesi; costo unitario: $m = 36$
$\sum_i c_i \times f_i$	$36 \times 100 + 36.000 \times 10 + 36.000 \times 50 \approx 2.2 \times 10^6$	$360 \times 100 + 36 \times 10 + 36.000 \times 50 \approx 1.8 \times 10^6$	$360 \times 100 + 360 \times 10 + 36 \times 50 \approx 4 \times 10^4$

Conviene quindi scegliere la vista 3

**Domanda 4** (20%)

Considerare lo scenario a fianco in cui tre client diversi inviano richieste ad un gestore della 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 al 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 la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato). Considerare uno scheduler con controllo di concorrenza basato su **Multversioni** (come in Postgres) e livello di isolamento **READ COMMITTED** sulle prime due transazioni e **SERIALIZABLE** sulla terza. Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia 200 e quello dell'oggetto y sia 500. Indicare, nell'ordine, le operazioni che vengono eseguite da ciascun client, specificando, per ciascuna, il valore che viene letto o scritto. In conclusione, dire se si verificano o meno anomalie.

client 1	client 2	client 3
begin read(x) x = x + 10 write(x)	begin read(y) y = y + 20 write(y) read(x) x = x - 20 write(x) commit	begin read(x)
read(y) y = y - 10 write(y) commit		<i>(dopo molto tempo)</i> read(x) commit

client 1	client 2	client 3
begin read(x) — legge 200 x = x + 10 write(x) — scrive 210	begin read(y) — legge 500 y = y + 20 write(y) — scrive 520 read(x) — legge 200 x = x - 20 wlock(x) — bloccata	begin read(x) — legge 200
read(y) — legge 500 y = y - 10 wlock(y) — bloccata	...	
write(y) — scrive 490 commit	abort	
	begin read(y) — legge 490 y = y + 20 write(y) — scrive 510 read(x) — legge 210 x = x - 20 write(x) — scrive 190 commit	
		read(x) commit read(x) — legge 200

Si verificano anomalie?  
No

Considerare nuovamente lo scenario della pagina precedente, ripetuto qui a fianco per comodità. Considerare uno scheduler con controllo di concorrenza basato su **2PL stretto** con livello di isolamento **SERIALIZABLE** sulle prime due transazioni e **READ COMMITTED** sulla terza. Mostrare il comportamento dello scheduler, supponendo ancora che il valore iniziale dell'oggetto x sia 200 e quello dell'oggetto y sia **500**.

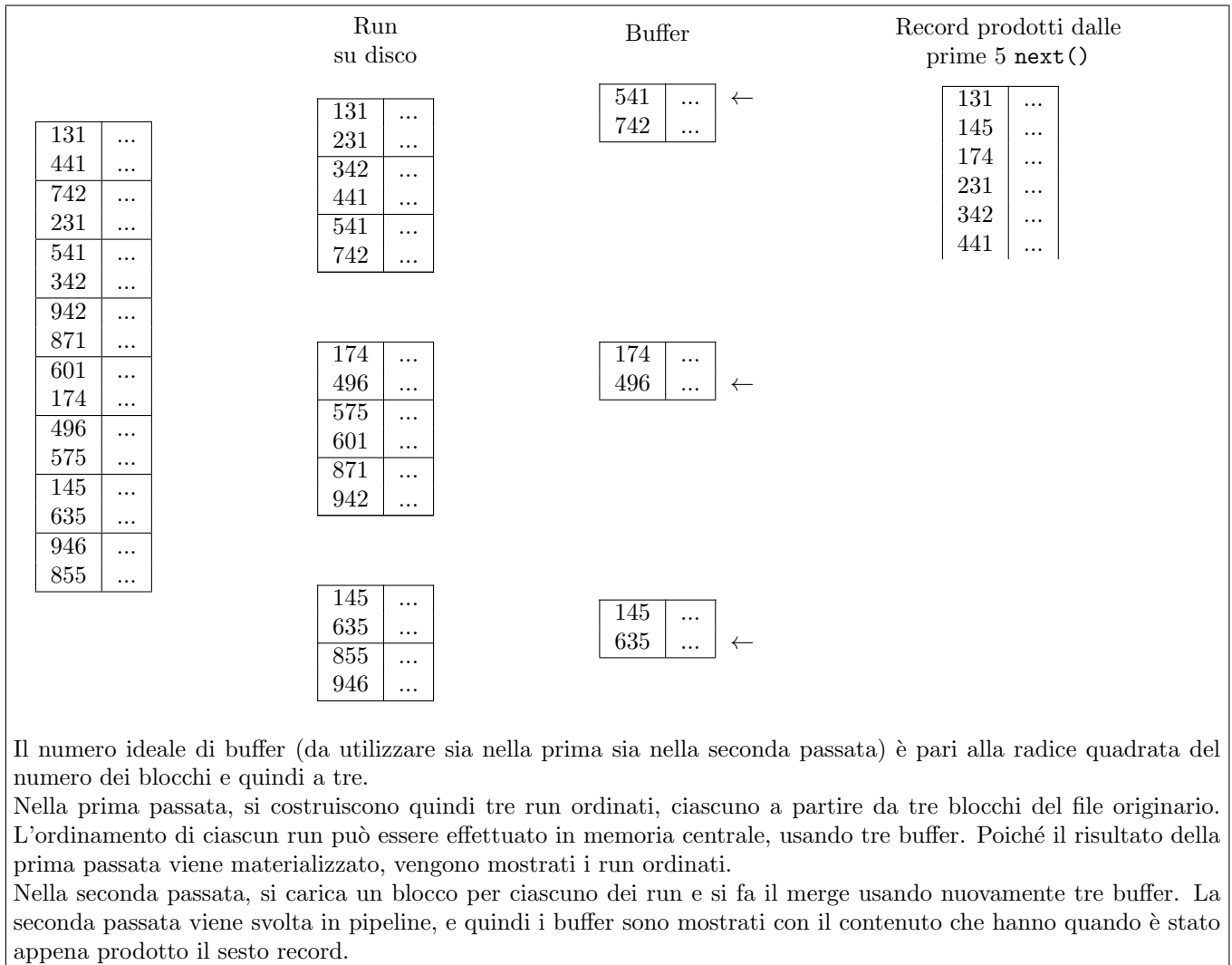
client 1	client 2	client 3
begin read(x) x = x + 10 write(x)	begin read(y) y = y + 20 write(y) read(x) x = x - 20 write(x) commit	begin read(x)
read(y) y = y - 10 write(y) commit		( <i>dopo molto tempo</i> ) read(x) commit

client 1	client 2	client 3
begin read(x) — legge 200 x = x + 10 write(x) — scrive 210	begin read(y) — legge 500 y = y + 20 write(y) — scrive 520 rlock(x) — bloccata	begin read(x) — legge 200 (rilascia il lock)
rlock(y) — bloccata	... abort	
read(y) — legge 500 y = y - 10 write(y) — scrive 490 commit	begin read(y) — legge 490 y = y + 20 write(y) — scrive 510 read(x) — legge 210 x = x - 20 write(x) — scrive 190 commit	read(x) — legge 190 commit

Si verificano anomalie?  
 Lettura inconsistente per il client 3

**Domanda 5** (20%)

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 3 buffer, con un fattore di blocco pari a 2 e quindi uno spazio occupato dalla relazione pari a 8 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 sei 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 sei chiamate di `next()`.



Basi di dati II — 5 luglio 2017

**Domanda 6** (15%) Considerare un sistema distribuito su cui viene eseguita una transazione che coinvolge tre nodi, un coordinatore M e due partecipanti R1 e R2. Dopo la richiesta di **prepare** da parte del coordinatore M, i due partecipanti ricevono e rispondono correttamente, e uno dei due, R1, va in crash subito dopo aver risposto. Il coordinatore riceve le risposte, prende la decisione, invia i relativi messaggi e subito dopo va anch'esso in crash (quindi senza fare in tempo a ricevere le conferme). Indicare, nello schema sottostante, una possibile sequenza di scritture sui log e invio di messaggi, supponendo che entrambi i nodi siano ripristinati abbastanza presto (ma che il coordinatore perda i messaggi di risposta inviati ad esso a seguito del commit). Per semplicità, si fa riferimento ad una sola transazione e quindi non c'è bisogno di indicarla. Per i messaggi si usi la notazione *tipo*→*destinatari* (come nell'esempio: **prepare**→R1,R2). Supporre che nel log del coordinatore si scrivano solo i record di **prepare**, **commit** e **complete**, mentre i messaggi sono gestiti in memoria.

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

Eventuali commenti: