

Basi di dati II

Esame — 1 luglio 2021

Tempo a disposizione: 2h15 per la prova completa e 1h15 per la prova breve.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (15% per la prova completa, 40% per la prova breve) Si consideri un sistema con blocchi di dimensione $B = 4$ KB e una relazione R con $N_R = 800.000$ ennuple e (fra gli altri) quattro attributi, A, C, E, F ciascuno con $v = 10$ valori diversi.

- (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 $v' = 1000$.

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 COUNT(*) FROM R WHERE A = 5`
3. `SELECT * FROM R WHERE A = 2 AND C = 4 AND E = 3 AND F = 5`, 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 2 nel caso (B))

Riportare le risposte nella tabella sottostante, nelle righe da 1 a 3, indicando formula e valore numerico.

Risposte

	(A) v valori diversi ($v = 10$)	(B) v' valori diversi ($v' = 1000$)
1		
2		
3		

Basi di dati II — 1 luglio 2021

Domanda 2 (25% per la prova completa, 60% per la prova breve) Un'agenzia di viaggio on-line vende ai clienti pacchetti di viaggio proposti dai tour operator e utilizza una base di dati con il seguente schema (ovviamente semplificato rispetto alla realtà):

- PACCHETTO (CodicePacchetto, Titolo, CodiceTourOperator, TipoSistemazione, CodiceDestinazione, PrezzoAPersona, GuadagnoAgenzia)
- TOUROPERATOR (CodiceTourOperator, NomeTourOperator)
- SISTEMAZIONE (TipoSistemazione, DescSistemazione); le sistemazioni sono “albergo”, “residence”, ...
- DESTINAZIONE (CodiceDestinazione, Nome, CodiceRegione, CodiceCategoriaDest)
- CATEGORIADESTINAZIONE (CodiceCategoriaDest, Categoria); le categorie sono “mare”, “montagna”, ...
- REGIONE (CodiceRegione, NomeRegione, CodiceNazione)
- NAZIONE (CodiceNazione, NomeNazione, CodiceContinente)
- CONTINENTE (CodiceContinente, NomeContinente)
- CLIENTE (CodiceCliente, Cognome, Nome, Professione, DataNascita)
- VIAGGIO (CodiceCliente, CodicePacchetto, DataPartenza, NumeroPersone)

L'agenzia desidera un sistema di supporto alle proprie attività che permetta di effettuare analisi sui pacchetti venduti in passato, al fine di comprendere sia le scelte dei clienti sulla base delle loro caratteristiche sia i profitti che derivano dalle vendite dei vari pacchetti. In particolare, si vuole realizzare uno schema dimensionale che permetta di conoscere il numero di viaggi venduti, con l'incasso complessivo e il profitto per l'agenzia, rispetto a varie dimensioni, quali il tour operator, le caratteristiche fondamentali dei clienti (fascia di età, professione), la destinazione (regione, località e sue caratteristiche), il tipo di sistemazione (categoria di albergo, appartamento, etc.)

Assumere che, per ragioni di privacy e di compattezza, non si vogliano rappresentare i singoli viaggi.

Grana dei fatti:

Schema dimensionale:

Basi di dati II — 1 luglio 2021

Descrivere, in modo strutturato e comprensibile, il processo di ETL che porta alla tabella dei fatti mostrata in risposta alla domanda precedente. Non sono necessari tutti i dettagli, ma è opportuno elencare i vari passi e mostrare i dettagli SQL di alcuni di essi.

Basi di dati II — 1 luglio 2021

Domanda 3 (20%)

Considerare il seguente scenario in cui tre 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 la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato).

client 1	client 2	client 3
begin read(x)	begin read(x)	begin read(x)
x = x + 100 write(x)	x = x + 200 write(x)	
commit	commit	
		<i>(dopo molto tempo)</i> read(x) commit

Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** (come in Postgres) e livello di isolamento **SERIALIZABLE** sulle prime due transazioni e **READ COMMITTED** sulla terza. Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia **10.000**. 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

Si verificano anomalie?

Basi di dati II — 1 luglio 2021

Considerare nuovamente lo scenario della pagina precedente, ripetuto qui sotto per comodità.

client 1	client 2	client 3
begin read(x) x = x + 100 write(x) commit	begin read(x) x = x + 200 write(x) commit	begin read(x) <i>(dopo molto tempo)</i> read(x) commit

Considerare uno scheduler con controllo di concorrenza ancora basato su **Multversioni** (come in Postgres) ma con 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 ancora **10.000**.

client 1	client 2	client 3

Si verificano anomalie?

Basi di dati II — 1 luglio 2021

Domanda 4 (10%)

Considerare nuovamente lo scenario della domanda precedente, ripetuto qui sotto per comodità.

client 1	client 2	client 3
begin read(x)		begin read(x)
	begin read(x) $x = x + 200$ write(x)	
$x = x + 100$ write(x)		
commit	commit	
		<i>(dopo molto tempo)</i> read(x) commit

Considerare uno scheduler con controllo di concorrenza basato su **2PL** con 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 ancora **10.000**.

client 1	client 2	client 3

Si verificano anomalie?

Basi di dati II — 1 luglio 2021

Domanda 5 (20%) Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione $R(ID, CodiceFiscale, Cognome, \dots)$ di cardinalità pari circa a $L = 400.000$, con ennuple di $e = 80$ byte, con due chiavi, ID e $CodiceFiscale$ (cioè il valore di ciascuna di esse, da solo, identifica univocamente una ennupla). Supporre che il sistema offra

- strutture primarie disordinate, ordinate o hash
- indici di tipo B-tree secondari o anche primari

Considerare un carico applicativo che preveda le seguenti operazioni **tutte di lettura**

1. ricerca di una ennupla sulla base del valore completo di ID , frequenza oraria $f_1 = 100$
2. ricerca di ennuple sulla base del $CodiceFiscale$, eventualmente parziale, con frequenza oraria $f_2 = 10$; supporre che il valore parziale sia molto selettivo e porti alla identificazione, in media, di $s = 2$ ennuple;
3. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) dell'attributo $Cognome$, con frequenza oraria $f_3 = 1000$; supporre che il valore parziale porti alla identificazione, in media, di $s = 10$ ennuple.

Progettare l'organizzazione fisica della relazione, individuando la struttura primaria (hash, disordinata o ordinata) e gli eventuali indici (da nessuno a tre). Ragionare in termini di numero di accessi a memoria secondaria, assumendo che: (i) gli indici abbiano profondità $p = 4$, (ii) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
struttura primaria indici utilizzati			
Costo Op. 1			
Costo Op. 2			
Costo Op. 3			
Costo tot			

Basi di dati II — 1 luglio 2021

Domanda 6 (10%) Considerare un sistema che utilizzi blocchi di lunghezza $D = 4$ KB (approssimabili a 4000 byte) e una tabella R con una struttura fisica heap con record a lunghezza fissa che occupano $L = 20$ byte ciascuno, in cui vengono inserite $M = 100.000$ ennuple, con valori della chiave tutti diversi fra loro e da quelli già nella relazione (quindi il sistema verifica il soddisfacimento del vincolo di chiave e ammette tutte le operazioni).

Rispondere alle domande seguenti, indicando formule e valori numerici:

Indicare il numero di scritture in memoria secondaria necessarie per realizzare i 100.000 inserimenti, supponendo che i record di log abbiano una lunghezza pari a circa il triplo di quella dei record del file, con riferimento ad un programma che utilizzi una transazione separata per ciascun inserimento

- numero di scritture di pagine di log:

- numero di scritture di pagine della relazione, assumendo che il sistema utilizzi una strategia undo-redo senza vincoli particolari

Come nel caso precedente, ma con riferimento ad un programma che, per realizzare i 100.000 inserimenti, utilizzi complessivamente $k = 10$ transazioni, ognuna con 10.000 inserimenti (e supponendo che non vi siano altre transazioni attive)

- numero di scritture di pagine di log:

- numero di scritture di pagine della relazione, sempre assumendo una strategia undo-redo senza vincoli particolari

Basi di dati II
Esame — 1 luglio 2021
Cenni sulle soluzioni

Tempo a disposizione: 2h15 per la prova completa e 1h15 per la prova breve.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (15% per la prova completa, 40% per la prova breve) Si consideri un sistema con blocchi di dimensione $B = 4$ KB e una relazione R con $N_R = 800.000$ ennuple e (fra gli altri) quattro attributi, A, C, E, F ciascuno con $v = 10$ 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 $v' = 1000$.

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 COUNT(*) FROM R WHERE A = 5
3. SELECT * FROM R WHERE A = 2 AND C = 4 AND E = 3 AND F = 5, 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 2 nel caso (B))

Riportare le risposte nella tabella sottostante, nelle righe da 1 a 3, indicando formula e valore numerico.

Risposte		
	(A) v valori diversi ($v = 10$)	(B) v' valori diversi ($v' = 1000$)
1	$SB = \frac{v \times N_R}{8 \times B} = \frac{10 \times 800.000}{8 \times 4000} = 250$	$SB' = \frac{v' \times N_R}{8 \times B} = \frac{2000 \times 800.000}{8 \times 4000} = 25.000$
2	basta accedere ad un vettore: $\frac{N_R}{8 \times B} = \frac{800.000}{8 \times 4000} = 25$	Idem
3	Accesso ad un vettore per ciascun indice (i) e poi ai record del file, attraverso la struttura intermedia (ii). (i) $4 \times \frac{N_R}{8 \times B} = \frac{4 \times 800.000}{8 \times 4000} = 100$ (ii) numero dei record nel risultato (moltiplicato per due) Totale 300	Rispetto al caso (A), cambiano solo i valori (per (ii)) Totale accessi circa 104.

Basi di dati II — 1 luglio 2021

Domanda 2 (25% per la prova completa, 60% per la prova breve) Un'agenzia di viaggio on-line vende ai clienti pacchetti di viaggio proposti dai tour operator e utilizza una base di dati con il seguente schema (ovviamente semplificato rispetto alla realtà):

- PACCHETTO (CodicePacchetto, Titolo, CodiceTourOperator, TipoSistemazione, CodiceDestinazione, PrezzoAPersona, GuadagnoAgenzia)
- TOUROPERATOR (CodiceTourOperator, NomeTourOperator)
- SISTEMAZIONE (TipoSistemazione, DescSistemazione); le sistemazioni sono “albergo”, “residence”, ...
- DESTINAZIONE (CodiceDestinazione, Nome, CodiceRegione, CodiceCategoriaDest)
- CATEGORIADESTINAZIONE (CodiceCategoriaDest, Categoria); le categorie sono “mare”, “montagna”, ...
- REGIONE (CodiceRegione, NomeRegione, CodiceNazione)
- NAZIONE (CodiceNazione, NomeNazione, CodiceContinente)
- CONTINENTE (CodiceContinente, NomeContinente)
- CLIENTE (CodiceCliente, Cognome, Nome, Professione, DataNascita)
- VIAGGIO (CodiceCliente, CodicePacchetto, DataPartenza, NumeroPersone)

L'agenzia desidera un sistema di supporto alle proprie attività che permetta di effettuare analisi sui pacchetti venduti in passato, al fine di comprendere sia le scelte dei clienti sulla base delle loro caratteristiche sia i profitti che derivano dalle vendite dei vari pacchetti. In particolare, si vuole realizzare uno schema dimensionale che permetta di conoscere il numero di viaggi venduti, con l'incasso complessivo e il profitto per l'agenzia, rispetto a varie dimensioni, quali il tour operator, le caratteristiche fondamentali dei clienti (fascia di età, professione), la destinazione (regione, località e sue caratteristiche), il tipo di sistemazione (categoria di albergo, appartamento, etc.)

Assumere che, per ragioni di privatezza e di compattezza, non si vogliano rappresentare i singoli viaggi.

Grana dei fatti:

Sono possibili diverse alternative. Se ne riporta una:

i viaggi effettuati in un certo periodo temporale (ad esempio sulla base del giorno di partenza), relativi ad un certo pacchetto, da parte di clienti di una certa fascia d'età e di una certa professione. In questa proposta viene esplicitato il pacchetto, che quindi, a livello di grana, rende non necessaria la specifica di tour operator, destinazione e tipo di sistemazione, che dipendono dal pacchetto

Schema dimensionale:

La grana sopra descritta permette diverse strutture per lo schema dimensionale. In particolare:

- per le caratteristiche del cliente, si può pensare a due dimensioni separate per età (o fascia) e professione
- per il pacchetto, si segnalano almeno tre possibilità
 - pacchetto viene denormalizzato includendo tour operator, sistemazione, destinazione (con gli elementi geografici)
 - quattro dimensioni per pacchetto, tour operator, sistemazione e destinazione; oppure tre mettendo insieme pacchetto e tour operator

Lo schema dimensionale deriva dalle considerazioni precedenti (non viene riportato qui, ma era ovviamente richiesto in sede di esame)

Basi di dati II — 1 luglio 2021

Descrivere, in modo strutturato e comprensibile, il processo di ETL che porta alla tabella dei fatti mostrata in risposta alla domanda precedente. Non sono necessari tutti i dettagli, ma è opportuno elencare i vari passi e mostrare i dettagli SQL di alcuni di essi.

Soluzione omessa per brevità. Dipende dalle scelte fatte in precedenza

Basi di dati II — 1 luglio 2021

Domanda 3 (20%)

Considerare il seguente scenario in cui tre 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 la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato).

client 1	client 2	client 3
begin read(x)	begin read(x)	begin read(x)
x = x + 100 write(x)	x = x + 200 write(x)	
commit	commit	
		<i>(dopo molto tempo)</i> read(x) commit

Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** (come in Postgres) e livello di isolamento **SERIALIZABLE** sulle prime due transazioni e **READ COMMITTED** sulla terza. Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia **10.000**. 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) <i>legge 10.000</i>	begin read(x) <i>legge 10.000</i>	begin read(x) <i>legge 10.000</i>
x = x + 100 xlock(x) <i>attesa</i>	x = x + 200 write(x) <i>scrive 10.200</i>	
write(x) <i>prova a scrivere</i> abort	commit	
begin read(x) <i>legge 10.200</i>		
x = x + 10 write(x) <i>scrive 10.300</i> commit		
		read(x) <i>legge 10.300</i> commit

Si verificano anomalie?

Lettura inconsistente per il client 3

Basi di dati II — 1 luglio 2021

Considerare nuovamente lo scenario della pagina precedente, ripetuto qui sotto per comodità.

client 1	client 2	client 3
begin read(x) x = x + 100 write(x) commit	begin read(x) x = x + 200 write(x) commit	begin read(x) <i>(dopo molto tempo)</i> read(x) commit

Considerare uno scheduler con controllo di concorrenza ancora basato su **Multversioni** (come in Postgres) ma con 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 ancora **10.000**. **Correggere soluzione**

client 1	client 2	client 3
begin read(x) <i>legge 10.000</i> x = x + 100 xlock(x) <i>attesa</i> write(x) <i>scrive 10.100</i> commit	begin read(x) <i>legge 10.000</i> x = x + 200 write(x) <i>scrive 10.200</i> commit	begin read(x) <i>legge 10.000</i> read(x) <i>legge 10.000</i> commit

Si verificano anomalie?
 Perdita di aggiornamento fra il client 1 e il client 2

Basi di dati II — 1 luglio 2021

Domanda 4 (10%)

Considerare nuovamente lo scenario della domanda precedente, ripetuto qui sotto per comodità.

client 1	client 2	client 3
begin read(x)		begin read(x)
	begin read(x) $x = x + 200$ write(x)	
$x = x + 100$ write(x)		
commit	commit	
		(<i>dopo molto tempo</i>) read(x) commit

Considerare uno scheduler con controllo di concorrenza basato su **2PL** con 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 ancora **10.000**.

Ci sono due possibili soluzioni. Vista la durata della transazione 3, le altre due sicuramente abortiscono per timeout. Potrebbero anche ripartire e andare ancora in timeout. Dopo la conclusione della 3, possono ripartire insieme e andare in in conflitto, oppure fortunatamente essere sfalsate.

client 1	client 2	client 3
begin read(x) <i>legge 10.000</i>		begin read(x) <i>legge 10.000</i>
	begin read(x) <i>legge 10.000</i> $x = x + 200$ xlock <i>attesa</i>	
$x = x + 100$ xlock(x) <i>attesa</i>		
...	...	
<i>abort</i>	<i>abort</i>	
		read(x) <i>legge 10.000</i> commit
...	...	

Si verificano anomalie?
vedere sopra

Basi di dati II — 1 luglio 2021

Domanda 5 (20%) Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione $R(ID, CodiceFiscale, Cognome, \dots)$ di cardinalità pari circa a $L = 400.000$, con ennuple di $e = 80$ byte, con due chiavi, ID e $CodiceFiscale$ (cioè il valore di ciascuna di esse, da solo, identifica univocamente una ennupla). Supporre che il sistema offra

- strutture primarie disordinate, ordinate o hash
- indici di tipo B-tree secondari o anche primari

Considerare un carico applicativo che preveda le seguenti operazioni **tutte di lettura**

1. ricerca di una ennupla sulla base del valore completo di ID , frequenza oraria $f_1 = 100$
2. ricerca di ennuple sulla base del $CodiceFiscale$, eventualmente parziale, con frequenza oraria $f_2 = 10$; supporre che il valore parziale sia molto selettivo e porti alla identificazione, in media, di $s = 2$ ennuple;
3. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) dell'attributo $Cognome$, con frequenza oraria $f_3 = 1000$; supporre che il valore parziale porti alla identificazione, in media, di $s = 10$ ennuple.

Progettare l'organizzazione fisica della relazione, individuando la struttura primaria (hash, disordinata o ordinata) e gli eventuali indici (da nessuno a tre). Ragionare in termini di numero di accessi a memoria secondaria, assumendo che: (i) gli indici abbiano profondità $p = 4$, (ii) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
	Struttura ordinata su Cognome e indici secondari su ID e CodiceFiscale	Indici secondari su ID, CodiceFiscale e Cognome	Hash su ID e indici secondari su Cognome e CodiceFiscale
Costo Op. 1	$(p-2+1) = \text{ca. } 3$: accesso alla foglia dell'indice e poi ad un record	$(p-2+1) = \text{ca. } 3$: accesso alla foglia dell'indice e poi ad un record	ca. 1: accesso hash
Costo Op. 2	$(p-2+2) = \text{ca. } 4$: accesso alla foglia dell'indice e poi a due record	$(p-2+2) = \text{ca. } 4$: accesso alla foglia dell'indice e poi ad un record	$(p-2+2) = \text{ca. } 4$: accesso alla foglia dell'indice e poi ad un record
Costo Op. 3	$(p-2+1) = \text{ca. } 3$: accesso alla foglia dell'indice e poi ai record (che sono nello stesso blocco o in due)	$(p-2+10) = \text{ca. } 12$: accesso alla foglia dell'indice e poi ai record (che sono in blocchi diversi)	$(p-2+10) = \text{ca. } 12$: accesso alla foglia dell'indice e poi ai record (che sono in blocchi diversi)
Totale	$3 \times 100 + 4 \times 10 + 3 \times 1000 = \text{ca. } 3300$	$3 \times 100 + 4 \times 10 + 12 \times 1000 = \text{ca. } 12.300$	$1 \times 100 + 4 \times 10 + 12 \times 1000 = \text{ca. } 12.100$

Basi di dati II — 1 luglio 2021

Domanda 6 (10%) Considerare un sistema che utilizzi blocchi di lunghezza $D = 4$ KB (approssimabili a 4000 byte) e una tabella R con una struttura fisica heap con record a lunghezza fissa che occupano $L = 20$ byte ciascuno, in cui vengono inserite $M = 100.000$ ennuple, con valori della chiave tutti diversi fra loro e da quelli già nella relazione (quindi il sistema verifica il soddisfacimento del vincolo di chiave e ammette tutte le operazioni).

Rispondere alle domande seguenti, indicando formule e valori numerici:

Indicare il numero di scritture in memoria secondaria necessarie per realizzare i 100.000 inserimenti, supponendo che i record di log abbiano una lunghezza pari a circa il triplo di quella dei record del file, con riferimento ad un programma che utilizzi una transazione separata per ciascun inserimento

- numero di scritture di pagine di log:

Risposta: almeno $M = 100.000$, una per transazione

- numero di scritture di pagine della relazione, assumendo che il sistema utilizzi una strategia undo-redo senza vincoli particolari

Risposta: al massimo una per transazione, $M = 100.000$; probabilmente di meno, anche solo $M/f = 500$, una per blocco (dove f indica il fattore di blocco $f = D/L = 200$)

Come nel caso precedente, ma con riferimento ad un programma che, per realizzare i 100.000 inserimenti, utilizzi complessivamente $k = 10$ transazioni, ognuna con 10.000 inserimenti (e supponendo che non vi siano altre transazioni attive)

- numero di scritture di pagine di log:

Risposta: per ogni transazione si debbono scrivere le pagine di log corrispondenti. Ogni transazione scrive M/k ennuple e quindi le relative scritture su log occupano $M/k \times L \times 3 = 600\text{KB}$ e cioè $M/k \times L \times 3 \times 1/D = \text{ca.}150$ blocchi. In totale, per $k = 10$ transazioni, circa $M \times L \times 3 \times 1/D = \text{ca.}1500$ scritture

- numero di scritture di pagine della relazione, sempre assumendo una strategia undo-redo senza vincoli particolari

Risposta: non si può dire con precisione, anche solo $M/f = 500$, una per blocco