

Basi di dati II — Prova parziale — 25 maggio 2020
Compito A

Basi di dati II — 25 maggio 2020 — Compito A

Domanda 1 (25%)

Considerare una relazione `conti` (`numero`, `saldo`) e il seguente scenario in cui due client 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.

T1	T2
<pre>start transaction isolation level ...; select * from conti where numero = 3; update conti set saldo = 20 where numero = 3; commit;</pre>	<pre>start transaction isolation level ...; select * from conti where numero = 3; update conti set saldo = 10 where numero = 3; commit;</pre>

Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** (come in Postgres). Per ciascuno dei tre possibili valori per il livello di isolamento (`read committed`, `repeatable read`, `serializable`), indicare: l'esito (**commit** o **abort**) di ciascuna delle due transazioni e l'ordine in cui i due eventi si verificano. Rispondere su tre righe, in ogni riga indicare il livello di isolamento cui ci si riferisce e poi le due azioni, nell'ordine in cui si verificano (ad esempio *livello* **commit(T2) abort(T1)**). Si possono aggiugnere commenti, ma non è necessario.

Basi di dati II — 25 maggio 2020 — Compito A

Domanda 2 (25%)

Si supponga di dover eseguire, sulla base di dati delle carriere degli studenti, l'interrogazione che calcola, per ogni corso, la media dei voti assegnati. Indicare quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) è opportuno scegliere in ciascuno dei seguenti casi, assumendo che siano comunque già presenti molti esami per ciascun corso e che, nel corso degli inserimenti e delle modifiche, vengano spesso inizialmente inseriti valori sbagliati (anche in modo significativo), che vengono poi corretti prima del commit:

1. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); i risultati dell'interrogazione debbono essere **accurati**
2. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); i risultati dell'interrogazione possono essere **approssimati**
3. l'interrogazione è eseguita mentre non viene eseguito alcun aggiornamento; i risultati dell'interrogazione debbono essere **accurati**
4. l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; i risultati dell'interrogazione debbono essere **accurati**
5. l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; i risultati dell'interrogazione possono essere **approssimati**

Indicare la risposta su cinque righe, con il numero e il livello di isolamento.

Basi di dati II — 25 maggio 2020 — Compito A

Domanda 3 (25%)

Considerare un sistema distribuito su cui viene eseguita una transazione T che coinvolge un coordinatore N1 e due partecipanti N2 e N3. Come schematizzato sotto, coordinatore N1 invia il messaggio di **prepare** e subito dopo va in crash mentre i due partecipanti ricevono e rispondono correttamente, ma uno dei due, N2, va in crash poco dopo. Supporre che, dopo il crash, il coordinatore N1 segua la strategia più semplice, abortendo la transazione, e che N1 e N2 siano ripristinati abbastanza presto. Supporre che i timeout per le varie richieste del coordinatore scattino all'incirca negli istanti indicati a sinistra della tabella e che il coordinatore non scriva nel log quali messaggi di ack ha ricevuto.

Nodo N1		Nodo N2		Nodo N3	
Log	Messaggi	Log	Messaggi	Log	Messaggi
prepare(T,N2,N3)	prepare(T)→N2,N3				
	<i>crash</i>		<i>crash</i>		
t_1	<i>restart</i>				
t_2					
t_3			<i>restart</i>		

Indicare quali record scrive nel proprio log il coordinatore e quali messaggi invia, dopo il restart e fino alla conclusione della transazione.

Basi di dati II — 25 maggio 2020 — Compito A

Domanda 4 (25%)

Considerare un sistema che utilizzi blocchi di lunghezza $D = 1$ KB (approssimabili a 1000 byte) e una tabella R con una struttura fisica heap con record a lunghezza fissa che occupano $L = 100$ byte ciascuno, in cui vengono inserite $N = 30.000$ 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).

Considerare due casi:

1. il programma utilizza $N=30.000$ transazioni, una per ciascun inserimento
2. il programma utilizza $k=100$ transazioni, ciascuna con $N/k=300$ inserimenti (supporre per semplicità che non vi siano altre transazioni attive)

Supponendo che i record del log abbiano una lunghezza pari a circa il triplo di quella dei record del file, indicare per ciascuno dei due casi:

- numero di scritture di pagine di log
- il numero minimo di scritture di pagine dati, assumendo che il sistema utilizzi una strategia undo-redo senza vincoli particolari

Rispondere in modo sintetico, su quattro righe separate (indicando numeri e possibilmente anche formule):

“1, pagine di log: ... ”;

“1. pagine dati: ...”,

“2, pagine di log: ... ”;

“2. pagine dati: ...”,

Basi di dati II — Prova parziale — 25 maggio 2020
Compito B

Basi di dati II — 25 maggio 2020 — Compito B

Domanda 1 (25%)

Considerare una relazione `conti` (`numero`, `saldo`) e il seguente scenario in cui due client 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.

T1	T2
<pre>start transaction isolation level ...; select * from conti where numero = 3; update conti set saldo = 20 where numero = 3; commit;</pre>	<pre>start transaction isolation level ...; select * from conti where numero = 3; update conti set saldo = 10 where numero = 3; commit;</pre>

Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** (come in Postgres). Per ciascuno dei tre possibili valori per il livello di isolamento (`read committed`, `repeatable read`, `serializable`), indicare: l'esito (**commit** o **abort**) di ciascuna delle due transazioni e l'ordine in cui i due eventi si verificano. Rispondere su tre righe, in ogni riga indicare il livello di isolamento cui ci si riferisce e poi le due azioni, nell'ordine in cui si verificano (ad esempio *livello* **commit(T2) abort(T1)**). Si possono aggiugnere commenti, ma non è necessario.

Basi di dati II — 25 maggio 2020 — Compito B

Domanda 2 (25%)

Si supponga di dover eseguire, sulla base di dati delle carriere degli studenti, l'interrogazione che calcola, per ogni corso, la media dei voti assegnati. Indicare quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) è opportuno scegliere in ciascuno dei seguenti casi, assumendo che siano comunque già presenti molti esami per ciascun corso e che, nel corso degli inserimenti e delle modifiche, vengano spesso inizialmente inseriti valori sbagliati (anche in modo significativo), che vengono poi corretti prima del commit:

1. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); i risultati dell'interrogazione possono essere **approssimati**
2. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); i risultati dell'interrogazione debbono essere **accurati**
3. l'interrogazione è eseguita mentre non viene eseguito alcun aggiornamento; i risultati dell'interrogazione debbono essere **accurati**
4. l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; risultati dell'interrogazione possono essere **approssimati**
5. l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; i risultati dell'interrogazione debbono essere **accurati**

Indicare la risposta su cinque righe, con il numero e il livello di isolamento.

Basi di dati II — 25 maggio 2020 — Compito B

Domanda 3 (25%)

Considerare un sistema distribuito su cui viene eseguita una transazione T che coinvolge un coordinatore C e due partecipanti R1 e R2. Come schematizzato sotto, coordinatore C invia il messaggio di **prepare** e subito dopo va in crash mentre i due partecipanti ricevono e rispondono correttamente, ma uno dei due, R1, va in crash poco dopo. Supporre che, dopo il crash, il coordinatore C segua la strategia più semplice, abortendo la transazione, e che C e R1 siano ripristinati abbastanza presto. Supporre che i timeout per le varie richieste del coordinatore scattino all'incirca negli istanti indicati a sinistra della tabella e che il coordinatore non scriva nel log quali messaggi di ack ha ricevuto.

Nodo C		Nodo R1		Nodo R2	
Log	Messaggi	Log	Messaggi	Log	Messaggi
prepare(T,R1,R2)	prepare(T)→R1,R2				
	<i>crash</i>		<i>crash</i>		
t_1	<i>restart</i>				
t_2					
t_3			<i>restart</i>		

Indicare quali record scrive nel proprio log il coordinatore e quali messaggi invia, dopo il restart e fino alla conclusione della transazione.

Basi di dati II — 25 maggio 2020 — Compito B

Domanda 4 (25%)

Considerare un sistema che utilizzi blocchi di lunghezza $D = 1$ KB (approssimabili a 1000 byte) e una tabella R con una struttura fisica heap con record a lunghezza fissa che occupano $L = 100$ byte ciascuno, in cui vengono inserite $R = 20.000$ 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).

Considerare due casi:

1. il programma utilizza $R=20.000$ transazioni, una per ciascun inserimento
2. il programma utilizza $k=100$ transazioni, ciascuna con $N/k=200$ inserimenti (supporre per semplicità che non vi siano altre transazioni attive)

Supponendo che i record del log abbiano una lunghezza pari a circa il triplo di quella dei record del file, indicare per ciascuno dei due casi:

- numero di scritture di pagine di log
- il numero minimo di scritture di pagine dati, assumendo che il sistema utilizzi una strategia undo-redo senza vincoli particolari

Rispondere in modo sintetico, su quattro righe separate (indicando numeri e possibilmente anche formule):

“1, pagine di log: ... ”;

“1. pagine dati: ...”,

“2, pagine di log: ... ”;

“2. pagine dati: ...”,

Basi di dati II — Prova parziale — 25 maggio 2020

Compito A

Cenni sulle soluzioni

Basi di dati II — 25 maggio 2020 — Compito A

Domanda 1 (25%)

Considerare una relazione `conti` (`numero`, `saldo`) e il seguente scenario in cui due client 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.

T1	T2
<pre>start transaction isolation level ...; select * from conti where numero = 3; update conti set saldo = 20 where numero = 3; commit;</pre>	<pre>start transaction isolation level ...; select * from conti where numero = 3; update conti set saldo = 10 where numero = 3; commit;</pre>

Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** (come in Postgres). Per ciascuno dei tre possibili valori per il livello di isolamento (`read committed`, `repeatable read`, `serializable`), indicare: l'esito (**commit** o **abort**) di ciascuna delle due transazioni e l'ordine in cui i due eventi si verificano. Rispondere su tre righe, in ogni riga indicare il livello di isolamento cui ci si riferisce e poi le due azioni, nell'ordine in cui si verificano (ad esempio *livello* **commit(T2) abort(T1)**). Si possono aggiugnere commenti, ma non è necessario.

Soluzione, per entrambi i compiti:

```
serializable, commit(T1), abort(T2)
repeatable read, commit(T1), abort(T2)
read committed, commit(T1), commit(T2)
```

Basi di dati II — 25 maggio 2020 — Compito A

Domanda 2 (25%)

Si supponga di dover eseguire, sulla base di dati delle carriere degli studenti, l'interrogazione che calcola, per ogni corso, la media dei voti assegnati. Indicare quale livello di isolamento (**READ UNCOMMITTED**, **READ COMMITTED**, **REPEATABLE READ** o **SERIALIZABLE**) è opportuno scegliere in ciascuno dei seguenti casi, assumendo che siano comunque già presenti molti esami per ciascun corso e che, nel corso degli inserimenti e delle modifiche, vengano spesso inizialmente inseriti valori sbagliati (anche in modo significativo), che vengono poi corretti prima del commit:

1. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); i risultati dell'interrogazione debbono essere **accurati**
2. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); i risultati dell'interrogazione possono essere **approssimati**
3. l'interrogazione è eseguita mentre non viene eseguito alcun aggiornamento; i risultati dell'interrogazione debbono essere **accurati**
4. l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; i risultati dell'interrogazione debbono essere **accurati**
5. l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; risultati dell'interrogazione possono essere **approssimati**

Indicare la risposta su cinque righe, con il numero e il livello di isolamento.

Soluzione, compito A:

- 1 S
- 2 RC
- 3 RU
- 4 RR
- 5 RC

Soluzione, compito B:

- 1 RC
- 2 S
- 3 RU
- 4 RC
- 5 RR

Basi di dati II — 25 maggio 2020 — Compito A

Domanda 3 (25%)

Considerare un sistema distribuito su cui viene eseguita una transazione T che coinvolge un coordinatore N1 e due partecipanti N2 e N3. Come schematizzato sotto, coordinatore N1 invia il messaggio di **prepare** e subito dopo va in crash mentre i due partecipanti ricevono e rispondono correttamente, ma uno dei due, N2, va in crash poco dopo. Supporre che, dopo il crash, il coordinatore N1 segua la strategia più semplice, abortendo la transazione, e che N1 e N2 siano ripristinati abbastanza presto. Supporre che i timeout per le varie richieste del coordinatore scattino all'incirca negli istanti indicati a sinistra della tabella e che il coordinatore non scriva nel log quali messaggi di ack ha ricevuto.

Nodo N1		Nodo N2		Nodo N3	
Log	Messaggi	Log	Messaggi	Log	Messaggi
prepare(T,N2,N3)	prepare(T)→N2,N3				
	<i>crash</i>	ready	<i>crash</i>	ready	
t_1	<i>restart</i>				
t_2					
t_3			<i>restart</i>		

Indicare quali record scrive nel proprio log il coordinatore e quali messaggi invia, dopo il restart e fino alla conclusione della transazione.

nel log: abort(T)
 messaggio: abort(T) a N2 e N3
 messaggio: abort(T) a N2
 messaggio: abort(T) a N2
 nel log: complete(T)

Domanda 4 (25%)

Considerare un sistema che utilizzi blocchi di lunghezza $D = 1$ KB (approssimabili a 1000 byte) e una tabella R con una struttura fisica heap con record a lunghezza fissa che occupano $L = 100$ byte ciascuno, in cui vengono inserite $N = 30.000$ tuple, 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).

Considerare due casi:

1. il programma utilizza $N=30.000$ transazioni, una per ciascun inserimento
2. il programma utilizza $k=100$ transazioni, ciascuna con $N/k=300$ inserimenti (supporre per semplicità che non vi siano altre transazioni attive)

Supponendo che i record del log abbiano una lunghezza pari a circa il triplo di quella dei record del file, indicare per ciascuno dei due casi:

- numero di scritture di pagine di log
- il numero minimo di scritture di pagine dati, assumendo che il sistema utilizzi una strategia undo-redo senza vincoli particolari

Rispondere in modo sintetico, su quattro righe separate (indicando numeri e possibilmente anche formule):

“1, pagine di log: ... ”;

“1. pagine dati: ...”,

“2, pagine di log: ... ”;

“2. pagine dati: ...”,

Per il compito A:

caso 1. pagine di log tante quante sono le transazioni, $N = 30.000$

caso 1. pagine dati (minimo, supponendo la scrittura più efficiente): pari al numero di blocchi occupato, $N \times L/D = 3000$

caso 2. pagine di log ogni transazione scrive più di un blocco, quindi possiamo considerare semplicemente il numero di blocchi necessari: $3 \times (N \times L/D) = 9000$ (cui andrebbe aggiunto lo spazio per record di commit, ma trascuriamo)

caso 2. pagine dati analogo al caso 1

Per il compito B, 30.000 diventa 20.000, 3000 diventa 2000 e 9000 diventa 6000