

Basi di dati II — Prova parziale — 16 aprile 2014 — Compito A

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%) Considerare un sistema con dischi con $N = 400$ blocchi per traccia

- tempo medio di posizionamento della testina (tempo di seek) $t_S = 5$ msec
- tempo medio di latenza (attesa dovuta alla rotazione) $t_L = 3$ msec
- tempo minimo di lettura di un blocco $t_B = 15 \mu\text{sec}$

Rispondere alle seguenti domande mostrando formula e valore numerico

1. Qual è il tempo medio necessario per leggere un blocco del quale sia dato l'indirizzo fisico?

2. Qual è il tempo medio necessario per la scansione sequenziale di un file costituito da $F = 100$ blocchi contigui, non letti di recente?

3. Qual è il tempo che si può ipotizzare necessario per eseguire un accesso diretto ad un record di un file attraverso un indice che abbia profondità $p = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 100$, usato di recente, ma in modo non molto intenso?

4. Qual è il tempo che si può ipotizzare necessario per eseguire $m = 1000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p = 4$, fan-out $f_I = 100$, con disponibilità di circa $P = 150$ pagine di buffer?

Basi di dati II — 16 aprile 2014 — Compito A

Domanda 2 (25%) Considerare le relazioni R1 ed R2 schematizzate sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Quindi R1 occupa $B_1 = 6$ blocchi e R2 ne occupa $B_2 = 8$.

Relazione R1

| | | | | | | | | | | | | | | | | | |
|----|-----|----|----|-----|----|----|-----|----|----|-----|----|----|-----|----|----|-----|----|
| 20 | X01 | AA | 21 | Y01 | DA | 22 | Z03 | AB | 23 | K03 | AB | 24 | Z03 | AB | 25 | Z03 | AB |
| | Y42 | CA | | X42 | CC | | W05 | EF | | W07 | EF | | W08 | EF | | W09 | EF |
| | W73 | CC | | W93 | CB | | X52 | HA | | X59 | HA | | X50 | HA | | X56 | HA |
| | Z55 | GC | | W54 | LB | | Y55 | EA | | Y54 | EA | | Y51 | EA | | Y57 | EA |

Relazione R2

| | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|---|----|----|---|----|----|---|----|----|---|----|----|---|----|----|---|----|----|---|----|----|---|
| 40 | AA | 3 | 41 | BC | 4 | 42 | LB | 7 | 43 | AA | 8 | 44 | AC | 3 | 45 | EA | 7 | 46 | BA | 5 | 47 | EF | 6 |
| | DA | 7 | | GB | 7 | | HB | 3 | | EC | 2 | | CB | 5 | | LB | 8 | | BB | 4 | | GA | 8 |

Si supponga di disporre di un buffer di $p = 4$ pagine.

Considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con il metodo nested loop senza utilizzo di indici. Supporre che non serva memorizzare il risultato e che quindi esso possa essere prodotto una ennupla alla volta (approccio "pipelining")

Indicare, nell'ordine, le prime quattro ennuple che vengono prodotte

Indicare gli indirizzi dei blocchi che si trovano nel buffer dopo che sono state prodotte le prime quattro ennuple.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime quattro ennuple del risultato.

Indicare il numero complessivo di accessi a memoria secondaria necessari per eseguire il join (indicare formula e numero)

Basi di dati II — 16 aprile 2014 — Compito A

Domanda 3 (15%) Si consideri 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 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, 21>
<SETSTRING, 21, R1, 33, 0, xxxx, Rossi>
<START, 22>
<SETSTRING, 22, R1, 44, 0, xxxx, Neri>
<START, 23>
<COMMIT, 21>
<SETSTRING, 23, R1, 33, 0, Rossi, Verdi>
<START, 24>
<SETSTRING, 24, R1, 55, 0, xxxx, Bruni>
<NQCKPT, 22, 23, 24>
<SETSTRING, 24, R1, 56, 0, xxxx, Neri>
<START, 25>
<SETSTRING, 25, R1, 66, 0, xxxx, Bianchi>
<COMMIT, 24>
```

1. Quali modifiche sulla base di dati debbono essere eseguite, nell’ordine, durante un recovery con strategia undo-redo in cui al checkpoint vengono salvate tutte le pagine sporche del buffer?

2. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery con strategia undo-redo in cui al checkpoint vengono salvate solo le pagine sporche del buffer modificate da transazioni andate in commit?

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

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

Basi di dati II — 16 aprile 2014 — Compito A

Domanda 4 (20%) Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** in ciascuna casella, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè generabile da uno scheduler multiversion con controllo di serializzabilità: “a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began”). Negli schedule, s_i indica l’inizio della transazione i e c_i il suo commit.

| | S | CSR | S2PL | MV |
|--|---|-----|------|----|
| $s_1, r_1(x), w_1(x), s_2, r_2(x), w_2(x), c_2, s_3, r_3(y), w_3(y), c_3, r_1(y), w_1(y), c_1$ | | | | |
| $s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$ | | | | |
| $s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$ | | | | |
| $s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$ | | | | |

Domanda 5 (20%) Si consideri una relazione $R(\text{CodiceImpiegato}, \text{Cognome}, \text{Nome}, \text{Categoria})$ con $L = 2.000.000$ ennuple, con fattore di blocco $f_R = 50$. Con riferimento alla ricerca di tutti gli impiegati con una certa mansione, indicare il costo dell’accesso sequenziale e di quello diretto con indice su $Categoria$ (con profondità $p = 4$) nei due casi seguenti (mostrare formule e valori numerici):

1. campo selettivo ($c_1 = 100.000$ valori diversi per $Categoria$)

| | |
|----------------------------|------------------------|
| costo accesso sequenziale: | costo accesso diretto: |
|----------------------------|------------------------|

2. campo poco selettivo ($c_2 = 20$ valori diversi per $Categoria$)

| | |
|----------------------------|------------------------|
| costo accesso sequenziale: | costo accesso diretto: |
|----------------------------|------------------------|

Basi di dati II — Prova parziale — 16 aprile 2014 — Compito A

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

Tempo a disposizione: un'ora e quindici minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%) Considerare un sistema con dischi con $N = 400$ blocchi per traccia

- tempo medio di posizionamento della testina (tempo di seek) $t_S = 5$ msec
- tempo medio di latenza (attesa dovuta alla rotazione) $t_L = 3$ msec
- tempo minimo di lettura di un blocco $t_B = 15$ μ sec

Rispondere alle seguenti domande mostrando formula e valore numerico

1. Qual è il tempo medio necessario per leggere un blocco del quale sia dato l'indirizzo fisico?

La somma del tempo medio di seek, del tempo medio di latenza e del tempo minimo di lettura di un blocco

$$t_{tot} = t_S + t_L + t_B = 5 + 3 + 0,015 \text{ msec} = \text{ca } 8 \text{ msec}$$

2. Qual è il tempo medio necessario per la scansione sequenziale di un file costituito da $F = 100$ blocchi contigui, non letti di recente?

Il tempo medio necessario per leggere un blocco (il primo) più il tempo minimo di lettura per ciascuno degli altri

$$t_{tot} + (F - 1) \times t_B = \text{ca } 9,5 \text{ msec}$$

3. Qual è il tempo che si può ipotizzare necessario per eseguire un accesso diretto ad un record di un file attraverso un indice che abbia profondità $p = 4$ e fan-out (fattore di blocco dell'indice) $f_I = 100$, usato di recente, ma in modo non molto intenso?

Si può immaginare che al massimo la radice dell'indice si trovi nel buffer, ma non gli altri nodi. Quindi, tre accessi per l'indice e uno per il record del file, in posizioni non prevedibili:

$$((p - 1) + 1) \times t_{tot} = \text{ca } 32 \text{ msec}$$

4. Qual è il tempo che si può ipotizzare necessario per eseguire $m = 1000$ accessi diretti in tempi ravvicinati a record di un file (molto grande) attraverso un indice che abbia profondità $p = 4$, fan-out $f_I = 100$, con disponibilità di circa $P = 150$ pagine di buffer?

Si può immaginare che la radice e un altro livello dell'indice restino nel buffer, dopo il primo caricamento, quindi, per ogni record, due accessi all'indice e uno al file, in posizioni non prevedibili (qualche accesso in più per il caricamento iniziale e qualcuno in meno per blocchi acceduti più volte):

$$(p - 2 + 1) \times m \times t_{tot} = \text{ca } 24 \text{ sec}$$

Basi di dati II — 16 aprile 2014 — Compito A

Domanda 2 (25%) Considerare le relazioni R1 ed R2 schematizzate sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Quindi R1 occupa $B_1 = 6$ blocchi e R2 ne occupa $B_2 = 8$.

Relazione R1

| | | | | | | | | | | | | | | | | | |
|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|
| 20 | X01 Y42 W73 Z55 | AA CA CC GC | 21 | Y01 X42 W93 W54 | DA CC CB LB | 22 | Z03 W05 X52 Y55 | AB EF HA EA | 23 | K03 W07 X59 Y54 | AB EF HA EA | 24 | Z03 W08 X50 Y51 | AB EF HA EA | 25 | Z03 W09 X56 Y57 | AB EF HA EA |
|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|----|--------------------------|----------------------|

Relazione R2

| | | | | | | | | | | | | | | | | | | | | | | | |
|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|
| 40 | AA DA | 3 7 | 41 | BC GB | 4 7 | 42 | LB HB | 7 3 | 43 | AA EC | 8 2 | 44 | AC CB | 3 5 | 45 | EA LB | 7 8 | 46 | BA BB | 5 4 | 47 | EF GA | 6 8 |
|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|----|----------|--------|

Si supponga di disporre di un buffer di $p = 4$ pagine.

Considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con il metodo nested loop senza utilizzo di indici. Supporre che non serva memorizzare il risultato e che quindi esso possa essere prodotto una ennupla alla volta (approccio "pipelining")

Indicare, nell'ordine, le prime quattro ennuple che vengono prodotte

(X01, AA, 3), (Y01, DA, 7), (W54, LB, 7), (X01, AA, 8)

Indicare gli indirizzi dei blocchi che si trovano nel buffer dopo che sono state prodotte le prime quattro ennuple.

20, 21, 22, 43

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime quattro ennuple del risultato.

20, 21, 22, 40, 41, 42, 43

Indicare il numero complessivo di accessi a memoria secondaria necessari per eseguire il join (indicare formula e numero)

$$B_1 + (B_1 / (p - 1)) \times B_2 = 22$$

Basi di dati II — 16 aprile 2014 — Compito A

Domanda 3 (15%) Si consideri 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 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, 21>
<SETSTRING, 21, R1, 33, 0, xxxx, Rossi>
<START, 22>
<SETSTRING, 22, R1, 44, 0, xxxx, Neri>
<START, 23>
<COMMIT, 21>
<SETSTRING, 23, R1, 33, 0, Rossi, Verdi>
<START, 24>
<SETSTRING, 24, R1, 55, 0, xxxx, Bruni>
<NQCKPT, 22, 23, 24>
<SETSTRING, 24, R1, 56, 0, xxxx, Neri>
<START, 25>
<SETSTRING, 25, R1, 66, 0, xxxx, Bianchi>
<COMMIT, 24>
```

1. Quali modifiche sulla base di dati debbono essere eseguite, nell’ordine, durante un recovery con strategia undo-redo in cui al checkpoint vengono salvate tutte le pagine sporche del buffer?

```
undo(SETSTRING,25,...),
undo(SETSTRING,23,...),
undo(SETSTRING,22,...),
redo(SETSTRING, 24, ...,56, Neri)
```

2. Quali modifiche sulla base di dati debbono essere eseguite durante un recovery con strategia undo-redo in cui al checkpoint vengono salvate solo le pagine sporche del buffer modificate da transazioni andate in commit?

```
undo(SETSTRING,25,...),
undo(SETSTRING, 23,...),
undo(SETSTRING, 22,...),
redo(SETSTRING, 24, ...,55, Bruni),
redo(SETSTRING, 24, ...,56, Neri)
```

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

```
undo(SETSTRING,25,...),
undo(SETSTRING,23,...),
undo(SETSTRING,22,...)
```

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

```
redo(SETSTRING, 24, ...,55, Bruni),
redo(SETSTRING, 24, ...,56, Neri)
```

Basi di dati II — 16 aprile 2014 — Compito A

Domanda 4 (20%) Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** in ciascuna casella, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè generabile da uno scheduler multiversion con controllo di serializzabilità: “a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began”). Negli schedule, s_i indica l’inizio della transazione i e c_i il suo commit.

Soluzioni per il compito A

| | S | CSR | S2PL | MV |
|--|----|-----|------|------|
| $s_1, r_1(x), w_1(x), s_2, r_2(x), w_2(x), c_2, s_3, r_3(y), w_3(y), c_3, r_1(y), w_1(y), c_1$ | no | sì | no | no |
| $s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$ | sì | sì | sì | sì |
| $s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$ | no | no | no | no |
| $s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$ | no | no | no | sì * |

(*) in questo caso $r_1(y)$ legge il valore di y all’inizio della transazione 1, cioè prima della scrittura $w_2(y)$

Domanda 5 (20%) Si consideri una relazione $R(\text{CodiceImpiegato}, \text{Cognome}, \text{Nome}, \text{Categoria})$ con $L = 2.000.000$ ennuple, con fattore di blocco $f_R = 50$. Con riferimento alla ricerca di tutti gli impiegati con una certa mansione, indicare il costo dell’accesso sequenziale e di quello diretto con indice su $Categoria$ (con profondità $p = 4$) nei due casi seguenti (mostrare formule e valori numerici):

1. campo selettivo ($c_1 = 100.000$ valori diversi per $Categoria$)

| | |
|---|---|
| costo accesso sequenziale: $L/f_R = 40.000$ | costo accesso diretto: $p + L/v_1 \approx 24$ |
|---|---|

2. campo poco selettivo ($c_2 = 20$ valori diversi per $Categoria$)

| | |
|---|--|
| costo accesso sequenziale: $L/f_R = 40.000$ | costo accesso diretto: $p + L/v_2 + \dots \approx 100.000$ |
|---|--|