

Basi di dati, primo modulo (vecchio ordinamento)

Prova parziale — 24 maggio 2002 — Compito A

Tempo a disposizione: un'ora e quarantacinque minuti. Libri aperti.

Domanda 1 (25%) Valutare il costo, in termini di numero di accessi a memoria secondaria (senza considerare gli eventuali benefici derivanti dalla gestione dei buffer), dell'esecuzione di una operazione di join con il metodo merge scan di due relazioni R_1 e R_2 , assumendo:

- blocchi di dimensione $B = 1000$
- puntatori ai blocchi dimensione $P = 2$
- record di dimensione rispettivamente $L_1 = 40$ e $L_2 = 50$
- attributo di join di dimensione $A = 4$
- cardinalità rispettivamente $N_1 = 200.000$ e $N_2 = 1.000.000$

in ciascuno seguenti casi (gli indici citati sono relativi all'attributo di join)

1. indice primario per entrambe le relazioni
2. indice primario per R_1 e secondario per R_2
3. indice secondario per entrambe le relazioni

Domanda 2 (25%) Considerare la seguente interrogazione in SQL:

```
SELECT DISTINCT A, K
FROM R1, R2, R3
WHERE R1.C = R2.D AND R2.E = R3.F AND R1.B = 3
```

Mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni), giustificando brevemente le scelte più significative, con riferimento alle seguenti informazioni sulla base di dati:

- la relazione R_1 (ABC) ha 800.000 ennuple e 100.000 valori diversi per l'attributo B, distribuiti uniformemente; ha una struttura heap e nessun indice
- la relazione R_2 (DE) ha 500.000 ennuple; è definito un vincolo di riferimento fra l'attributo E e la chiave F della relazione R_3 ; ha una struttura heap e un indice secondario sulla chiave D
- la relazione R_3 (FK) ha 1.000 ennuple e ha una struttura hash sulla chiave F

Domanda 3 (25%) Si supponga di dover eseguire una interrogazione che calcola statistiche su una base di dati (ad esempio: trovare per ciascun dipartimento la media degli stipendi degli impiegati). Indicare (con un breve commento) quali delle seguenti affermazioni sono vere (si ricorda che i livelli di isolamento, in ordine crescente di severità, sono: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ e SERIALIZABLE; inoltre, per "tempo morto" si intende qui un periodo in cui non sono eseguiti aggiornamenti)

1. poiché la transazione è di sola lettura, può essere definita come READ-ONLY; diventa così più efficiente, qualunque sia il livello di isolamento scelto
2. eseguendo la transazione in un "tempo morto", essa ha la stessa efficienza qualunque sia livello di isolamento scelto
3. eseguendo la transazione in un "tempo morto", si ha un miglioramento dell'efficienza con ciascuno dei livelli di isolamento; l'efficienza è di norma maggiore se la transazione è definita come READ-ONLY e READ UNCOMMITTED
4. eseguendo la transazione concorrentemente ad aggiornamenti e desiderando dati "coerenti," è necessario definire la transazione come SERIALIZABLE
5. eseguendo la transazione concorrentemente ad aggiornamenti e accettando dati "incoerenti," si può definire la transazione come READ UNCOMMITTED
6. eseguendo la transazione concorrentemente ad aggiornamenti che consistano solo in modifiche di stipendi ma non in inserimenti di nuovi impiegati, si può definire la transazione come REPEATABLE READ

Domanda 4 (25%) Considerare i seguenti schedule:

1. $r_1(x)w_1(x)r_2(x)w_2(x)r_0(y)w_1(y)$
2. $r_1(x)w_1(x)r_2(x)w_2(x)r_3(y)w_1(y)$
3. $r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$
4. $r_2(x)w_2(x)r_1(x)w_1(x)$

Specificare, con una breve giustificazione, a quali delle seguenti classi ciascuno di essi appartiene: S (seriale), VSR (view-serializzabile), CSR (conflict-serializzabile), 2PL (generabile da uno scheduler basato sul lock a due fasi) e TS (generabile da uno scheduler che utilizzi il metodo dei timestamp; si assuma che l'ordinamento degli identificatori delle transazioni corrisponda a quello dei timestamp).

Basi di dati, primo modulo (vecchio ordinamento)

Prova parziale – 24 maggio 2002 – Compito B

Tempo a disposizione: un'ora e quarantacinque minuti. Libri aperti.

Domanda 1 (25%) Valutare il costo, in termini di numero di accessi a memoria secondaria (senza considerare gli eventuali benefici derivanti dalla gestione dei buffer), dell'esecuzione di una operazione di join con il metodo merge scan di due relazioni R_1 e R_2 , assumendo:

- blocchi di dimensione $B = 1000$
- puntatori ai blocchi dimensione $P = 2$
- record di dimensione rispettivamente $L_1 = 40$ e $L_2 = 50$
- attributo di join di dimensione $A = 4$
- cardinalità rispettivamente $n_1 = 1.000.000$ e $n_2 = 500.000$

in ciascuno seguenti casi (gli indici citati sono relativi all'attributo di join)

1. indice secondario per entrambe le relazioni
2. indice primario per R_1 e secondario per R_2
3. indice primario per entrambe le relazioni

Domanda 2 (25%) Considerare la seguente interrogazione in SQL:

```
SELECT DISTINCT A, L
FROM T1, T2, T3
WHERE T1.C = T2.D AND T2.E = T3.F AND T1.B = 3
```

Mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni), giustificando brevemente le scelte più significative, con riferimento alle seguenti informazioni sulla base di dati:

- la relazione $T_1(\underline{A}BC)$ ha 800.000 ennuple e 100.000 valori diversi per l'attributo B, distribuiti uniformemente; ha una struttura heap e un indice secondario sulla chiave A
- la relazione $T_2(\underline{D}E)$ ha 500.000 ennuple; è definito un vincolo di riferimento fra l'attributo E e la chiave F della relazione T_3 ; ha una struttura heap e un indice secondario sulla chiave D
- la relazione $T_3(\underline{F}L)$ ha 1.000 ennuple e ha una struttura hash sulla chiave F

Domanda 3 (25%) Si supponga di dover eseguire una interrogazione che calcola statistiche su una base di dati (ad esempio: trovare per ciascun dipartimento la media degli stipendi degli impiegati). Indicare (con un breve commento) quali delle seguenti affermazioni sono vere (si ricorda che i livelli di isolamento, in ordine crescente di severità, sono: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ e SERIALIZABLE; inoltre, per "tempo morto" si intende qui un periodo in cui non sono eseguiti aggiornamenti)

1. poiché la transazione è di sola lettura, può essere definita come READ-ONLY; diventa così più efficiente, qualunque sia il livello di isolamento scelto
2. poiché la transazione è di sola lettura, può essere definita come READ-ONLY e READ UNCOMMITTED; diventa così più efficiente, fornendo risultati corretti nel caso in cui sia eseguita in un "tempo morto"
3. eseguendo la transazione in un "tempo morto", essa ha la stessa efficienza qualunque sia livello di isolamento scelto
4. eseguendo la transazione in un "tempo morto", si ha un miglioramento dell'efficienza con ciascuno dei livelli di isolamento; l'efficienza è di norma maggiore se la transazione è definita come READ-ONLY e READ UNCOMMITTED
5. eseguendo la transazione concorrentemente ad aggiornamenti e accettando dati "incoerenti," si può definire la transazione come READ UNCOMMITTED
6. eseguendo la transazione concorrentemente ad aggiornamenti che consistano solo in modifiche di stipendi ma non in inserimenti di nuovi impiegati, si può definire la transazione come REPEATABLE READ

Domanda 4 (25%) Considerare i seguenti schedule:

1. $w_0(z)r_1(z)r_2(z)w_2(z)w_2(y)$
2. $r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$
3. $r_2(x)w_2(x)r_1(x)w_1(x)$
4. $w_0(x)r_2(x)r_1(x)w_2(x)w_2(z)$

Specificare, con una breve giustificazione, a quali delle seguenti classi ciascuno di essi appartiene: S (seriale), VSR (view-serializzabile), CSR (conflict-serializzabile), 2PL (generabile da uno scheduler basato sul lock a due fasi) e TS (generabile da uno scheduler che utilizzi il metodo dei timestamp; si assuma che l'ordinamento degli identificatori delle transazioni corrisponda a quello dei timestamp).

Basi di dati, primo modulo (vecchio ordinamento)

Prova parziale – 24 maggio 2002 – Compito C

Tempo a disposizione: un'ora e quarantacinque minuti. Libri aperti.

Domanda 1 (25%) Valutare il costo, in termini di numero di accessi a memoria secondaria (senza considerare gli eventuali benefici derivanti dalla gestione dei buffer), dell'esecuzione di una operazione di join con il metodo merge scan di due relazioni R_1 e R_2 , assumendo:

- blocchi di dimensione $B = 1000$
- puntatori ai blocchi dimensione $P = 2$
- record di dimensione rispettivamente $L_1 = 40$ e $L_2 = 50$
- attributo di join di dimensione $A = 4$
- cardinalità rispettivamente $c_1 = 400.000$ e $c_2 = 500.000$

in ciascuno seguenti casi (gli indici citati sono relativi all'attributo di join)

1. indice primario per entrambe le relazioni
2. indice primario per R_1 e secondario per R_2
3. indice secondario per entrambe le relazioni

Domanda 2 (25%) Considerare la seguente interrogazione in SQL:

```
SELECT DISTINCT A, K
FROM X1, X2, X3
WHERE X1.C = X2.D AND X2.E = X3.F AND X1.B = 3
```

Mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni), giustificando brevemente le scelte più significative, con riferimento alle seguenti informazioni sulla base di dati:

- la relazione X_1 (\underline{ABC}) ha 800.000 ennuple e 100.000 valori diversi per l'attributo B, distribuiti uniformemente; ha una struttura heap e nessun indice
- la relazione X_2 (\underline{DE}) ha 500.000 ennuple; è definito un vincolo di riferimento fra l'attributo E e la chiave F della relazione X_3 ; ha una struttura heap e un indice secondario sulla chiave D
- la relazione X_3 (\underline{FK}) ha 1.000 ennuple e ha una struttura hash sulla chiave F

Domanda 3 (25%) Si supponga di dover eseguire una interrogazione che calcola statistiche su una base di dati (ad esempio: trovare per ciascun dipartimento la media degli stipendi degli impiegati). Indicare (con un breve commento) quali delle seguenti affermazioni sono vere (si ricorda che i livelli di isolamento, in ordine crescente di severità, sono: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ e SERIALIZABLE; inoltre, per "tempo morto" si intende qui un periodo in cui non sono eseguiti aggiornamenti)

1. poiché la transazione è di sola lettura, può essere definita come READ-ONLY; diventa così più efficiente, qualunque sia il livello di isolamento scelto
2. poiché la transazione è di sola lettura, può essere definita come READ-ONLY e READ UNCOMMITTED; diventa così più efficiente, fornendo risultati comunque corretti
3. eseguendo la transazione in un "tempo morto", essa ha la stessa efficienza qualunque sia livello di isolamento scelto
4. eseguendo la transazione in un "tempo morto", si ha un miglioramento dell'efficienza con ciascuno dei livelli di isolamento; l'efficienza è di norma maggiore se la transazione è definita come READ-ONLY e READ UNCOMMITTED
5. eseguendo la transazione concorrentemente ad aggiornamenti e desiderando dati "coerenti," è necessario definire la transazione come SERIALIZABLE
6. eseguendo la transazione concorrentemente ad aggiornamenti che consistano solo in modifiche di stipendi ma non in inserimenti di nuovi impiegati, si può definire la transazione come REPEATABLE READ

Domanda 4 (25%) Considerare i seguenti schedule:

1. $r_1(x)w_1(x)r_2(x)w_2(x)r_3(y)w_1(y)$
2. $w_0(z)r_1(z)r_2(z)w_2(z)w_2(y)$
3. $r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$
4. $r_2(x)w_2(x)r_1(x)w_1(x)$

Specificare, con una breve giustificazione, a quali delle seguenti classi ciascuno di essi appartiene: S (seriale), VSR (view-serializzabile), CSR (conflict-serializzabile), 2PL (generabile da uno scheduler basato sul lock a due fasi) e TS (generabile da uno scheduler che utilizzi il metodo dei timestamp; si assuma che l'ordinamento degli identificatori delle transazioni corrisponda a quello dei timestamp).

Basi di dati, primo modulo (vecchio ordinamento)

Prova parziale — 24 maggio 2002 — Compito D

Tempo a disposizione: un'ora e quarantacinque minuti. Libri aperti.

Domanda 1 (25%) Valutare il costo, in termini di numero di accessi a memoria secondaria (senza considerare gli eventuali benefici derivanti dalla gestione dei buffer), dell'esecuzione di una operazione di join con il metodo merge scan di due relazioni R_1 e R_2 , assumendo:

- blocchi di dimensione $B = 1000$
- puntatori ai blocchi dimensione $P = 2$
- record di dimensione rispettivamente $L_1 = 40$ e $L_2 = 50$
- attributo di join di dimensione $A = 4$
- cardinalità rispettivamente $C_1 = 1.000.000$ e $C_2 = 200.000$

in ciascuno seguenti casi (gli indici citati sono relativi all'attributo di join)

1. indice secondario per entrambe le relazioni
2. indice primario per R_1 e secondario per R_2
3. indice primario per entrambe le relazioni

Domanda 2 (25%) Considerare la seguente interrogazione in SQL:

```
SELECT DISTINCT A, M
FROM R1, R2, R3
WHERE R1.C = R2.D AND R2.E = R3.F AND R1.B = 3
```

Mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni), giustificando brevemente le scelte più significative, con riferimento alle seguenti informazioni sulla base di dati:

- la relazione R_1 (\underline{ABC}) ha 800.000 ennuple e 100.000 valori diversi per l'attributo B, distribuiti uniformemente; ha una struttura heap e un indice secondario sulla chiave A
- la relazione R_2 (\underline{DE}) ha 500.000 ennuple; è definito un vincolo di riferimento fra l'attributo E e la chiave F della relazione R_3 ; ha una struttura heap e un indice secondario sulla chiave D
- la relazione R_3 (\underline{FM}) ha 1.000 ennuple e ha una struttura hash sulla chiave F

Domanda 3 (25%) Si supponga di dover eseguire una interrogazione che calcola statistiche su una base di dati (ad esempio: trovare per ciascun dipartimento la media degli stipendi degli impiegati). Indicare (con un breve commento) quali delle seguenti affermazioni sono vere (si ricorda che i livelli di isolamento, in ordine crescente di severità, sono: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ e SERIALIZABLE; inoltre, per "tempo morto" si intende qui un periodo in cui non sono eseguiti aggiornamenti)

1. poiché la transazione è di sola lettura, può essere definita come READ-ONLY e READ UNCOMMITTED; diventa così più efficiente, fornendo risultati comunque corretti
2. poiché la transazione è di sola lettura, può essere definita come READ-ONLY e READ UNCOMMITTED; diventa così più efficiente, fornendo risultati corretti nel caso in cui sia eseguita in un "tempo morto"
3. eseguendo la transazione in un "tempo morto", essa ha la stessa efficienza qualunque sia livello di isolamento scelto
4. eseguendo la transazione in un "tempo morto", si ha un miglioramento dell'efficienza con ciascuno dei livelli di isolamento; l'efficienza è di norma maggiore se la transazione è definita come READ-ONLY e READ UNCOMMITTED
5. eseguendo la transazione concorrentemente ad aggiornamenti e accettando dati "incoerenti," si può definire la transazione come READ UNCOMMITTED
6. eseguendo la transazione concorrentemente ad aggiornamenti che consistano solo in modifiche di stipendi ma non in inserimenti di nuovi impiegati, si può definire la transazione come REPEATABLE READ

Domanda 4 (25%) Considerare i seguenti schedule:

1. $r_1(x)w_1(x)r_2(x)w_2(x)r_0(y)w_1(y)$
2. $r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$
3. $r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$
4. $r_2(x)w_2(x)r_1(x)w_1(x)$

Specificare, con una breve giustificazione, a quali delle seguenti classi ciascuno di essi appartiene: S (seriale), VSR (view-serializzabile), CSR (conflict-serializzabile), 2PL (generabile da uno scheduler basato sul lock a due fasi) e TS (generabile da uno scheduler che utilizzi il metodo dei timestamp; si assuma che l'ordinamento degli identificatori delle transazioni corrisponda a quello dei timestamp).