

Tecnologie delle Basi di Dati M

Appello del 19/10/2012

Esercizio 1 (4 punti)

Si consideri un sistema che gestisca le transazioni secondo la procedura *Undo-Redo*. Si supponga che il log del sistema abbia il seguente contenuto al verificarsi di un *system crash*:

1. BEGIN(T1)
2. UPDATE(T1, PA, VA1, VA2)
3. BEGIN(T2)
4. UPDATE(T2, PA, VA2, VA3)
5. COMMIT(T1)
6. UPDATE(T2, PB, VB1, VB2)
7. CHECKPOINT
8. END(T1)
9. UPDATE(T2, PC, VC1, VC2)
10. BEGIN(T3)
11. UPDATE(T3, PD, VD1, VD2)
12. COMMIT(T2)

si risponda alle seguenti domande, giustificando le risposte:

1. Il log può essere generato da un sistema che utilizzi il protocollo Strict-2PL?
 - a. Se sì, mostrare la sequenza di richieste da parte delle transazioni.
 - b. Se no, si indichi quali record andrebbero spostati (e in che posizione) per ottenere un log "corretto".
2. Quali pagine verrebbero incluse nella tabella delle pagine sporche dall'algoritmo ARIES? Quale valore avrebbe ciascuna di tali pagine all'inizio della procedura di Redo?

Esercizio 2 (3 punti)

Data la relazione con schema:

```
Impiegati(matricola, nome, residenza, mansione)
```

si consideri la seguente interrogazione SQL:

```
SELECT I.nome
FROM Impiegati I
WHERE I.mansione = 'tecnico'
AND I.nome LIKE 'P%'
```

tenendo conto che entrambi i predicati hanno un fattore di selettività del 10% e che la relazione Impiegati contiene 10000 tuple ed occupa 200 pagine. Si indichi il costo del miglior piano di accesso nei seguenti casi:

1. Presenza di un solo indice clustered su mansione (10 foglie).
2. Presenza di un solo indice clustered su nome (15 foglie).
3. Presenza di un solo indice unclustered (TID ordinate) su mansione (10 foglie).
4. Presenza di un solo indice unclustered (TID ordinate) su nome (15 foglie).
5. Presenza di un solo indice clustered su <mansione, nome> (50 foglie).
6. Presenza di un solo indice unclustered (TID ordinate) su <mansione, nome> (50 foglie).

Suggerimento: per la formula di Cardenas si utilizzino i seguenti valori, validi per $P = 200$:

R	$\Phi(R, 200)$
0	0
50	44.34
100	78.85
150	105.70
200	126.61
250	142.88
300	155.54
350	165.40
400	173.07
450	179.04

R	$\Phi(R, 200)$
500	183.69
550	187.30
600	190.12
650	192.31
700	194.01
750	195.34
800	196.37
850	197.18
900	197.80
950	198.29

R	$\Phi(R, 200)$
1000	198.67
1050	198.96
1100	199.19
1150	199.37
1200	199.51
1250	199.62
1300	199.70
1350	199.77
1400	199.82
1450	199.86

Esercizio 3 (5 punti)

Si confrontino i protocolli 2PL e Strict 2PL, specificando in particolare i motivi per cui il secondo è da preferirsi rispetto al primo.

Esercizio 4 (3 punti)

Gli algoritmi presentati per la valutazione efficiente di join sono adatti ad essere utilizzati nel caso in cui la condizione di join sia una condizione di uguaglianza. Si discuta sulla possibile estensione degli algoritmi *sort-merge join* e *hash join* per la valutazione efficiente di outer join (destro, sinistro e full).

Soluzione Esercizio 1

1. No: la transazione T2 non può scrivere la pagina PA fino al termine di T1. Il record 4. andrebbe quindi spostato dopo il commit di T1.
2. La tabella delle pagine sporche comprenderebbe PA (con valore VA3), PB (con valore VB2), PC (con valore VC1) e PD (con valore VD1).

Soluzione Esercizio 2

Costo scan sequenziale = **200**

Costo indice clustered su *mansione*: $NL \times 0.1 + NP \times 0.1 = 10 \times 0.1 + 200 \times 0.1 = 1 + 20 = \mathbf{21}$

Costo indice clustered su *nome*: $NL \times 0.1 + NP \times 0.1 = 15 \times 0.1 + 200 \times 0.1 = 2 + 20 = \mathbf{22}$

Costo indice unclustered su *mansione*: $NL \times 0.1 + \Phi(NT/10, NP) = 10 \times 0.1 + \Phi(1000, 200) = 1 + 199 = \mathbf{200}$

Costo indice unclustered su *nome*: $NL \times 0.1 + \Phi(NT/10, NP) = 15 \times 0.1 + \Phi(1000, 200) = 2 + 199 = \mathbf{201}$

Costo indice clustered su *<mansione, nome>*: $NL \times 0.01 = 50 \times 0.01 = \mathbf{1}$

Costo indice unclustered su *<mansione, nome>*: $NL \times 0.01 = 50 \times 0.01 = \mathbf{1}$