

Basi di Dati

Docente: Prof. Pierangela Samarati

Seconda prova in itinere - 8 Maggio 2015

Tempo a disposizione 2:00h

Domanda 1)

Con riferimento alle basi di dati distribuite, illustrare il funzionamento del protocollo *two-phase commit*.

Illustrare dapprima il funzionamento in *assenza di guasti* e poi descrivere cosa succede in caso di *guasti* e le *azioni di ripristino* necessarie.

Domanda 2)

Rispondere brevemente, ma in modo preciso e completo, alle seguenti domande.

1. Discutere il problema del *deadlock* in un sistema *distribuito* illustrando un possibile esempio di algoritmo per la sua rilevazione unitamente ad un esempio di esecuzione.
2. Illustrare il concetto di *trigger* nelle basi di dati attive, indicandone le componenti, i possibili livelli di granularità e le modalità di attivazione.
Illustrate inoltre un esempio di insieme di trigger (composto da un trigger soltanto) *non terminante*.

Esercizio 1)

Avendo le seguenti informazioni riguardo una **ripresa a caldo**, si indichi per ciascuna affermazione riportata nella tabella allegata se è sicuramente vera (*Vero*), sicuramente falsa (*Falso*), o se *non è possibile determinarlo* (potrebbe essere vera oppure falsa).

- record di checkpoint: CK(T4, T6);
- insieme di UNDO: {T4, T5, T6};
- insieme di REDO: {T2, T3};
- operazioni di UNDO:
 - I(T4, O4, A4) → DELETE(O4)
 - D(T6, O6, B6) → INSERT(O6), O6 := B6
 - D(T4, O7, B7) → INSERT(O7), O7 := B2
 - U(T4, O8, B8, A8) → O8 := B8
- operazioni di REDO:
 - D(T2, O2, B2) → DELETE(O2)
 - U(T2, O1, B1, A1) → O1 := A1
 - I(T3, O3, A3) → INSERT(O3), O3 := A3

Esercizio 2)

Dati i seguenti schedule:

- $w_1(x) w_2(y) r_2(z) r_1(x) w_2(z) r_3(x) w_1(y) r_1(z) w_3(z)$
- $r_1(x) w_2(y) w_2(z) r_1(y) w_3(z) r_1(z) w_1(y) r_2(x) w_1(z)$

Si dica se gli schedule sono *VSR* e/o *CSR*, indicando (qualora esistano) tutti gli schedule seriali equivalenti. Si svolga l'esercizio illustrando dettagliatamente il processo/ragionamento seguito.

Per il primo schedule indicare se è possibile *rimuovere* o *aggiungere* **una** operazione (specificando quale operazione andrebbe rimossa o aggiunta e in quale posizione) per rendere lo schedule:

- VSR, CSR
- non VSR, CSR
- VSR, non CSR
- non VSR, non CSR

Esercizio 3)

Si consideri lo schedule:

$$r_1(x) r_4(x) w_4(x) r_1(y) r_4(z) w_4(z) w_3(y) w_3(z) w_1(t) w_2(z) w_2(t)$$

Dire se può essere stato generato da uno scheduler basato su *2PL* base, motivando opportunamente la risposta. Si consideri, per la soluzione, un lock a due stati.

Esercizio 4)

Data la seguente base di dati:

PAZIENTE(Codice, Nome, Cognome, DataNascita, Telefono)

APPUNTAMENTO(Id, CodPaziente, Medico, Reparto, Esame, Data, Ora)

- Progettare e scrivere la frammentazione *orizzontale* in base ai seguenti criteri:
 - PAZIENTE diviso in 3 frammenti in base alla *DataNascita* (che si distingue a seconda che sia prima del 1950, fra il 1950 e il 1980 e dopo il 1980) allocati nei 3 centri di gestione dei dati medici dell'ospedale allocati rispettivamente al "Centro 1", "Centro 2", e "Centro 3".
 - APPUNTAMENTO diviso in 3 frammenti in base al *Reparto* dove è stato fissato l'*Appuntamento* stesso (a seconda che sia *Chirurgia*, *Pediatria*, o un qualsiasi altro reparto) allocati nei 3 centri di gestione dei dati medici dell'ospedale allocati rispettivamente al "Centro 1", "Centro 2", e "Centro 3".
- Scrivere poi una query che ritorni tutti i dati degli *Appuntamenti* presi per il reparto *Pediatria* per una data successiva al 1 giugno 2015 considerando tutti i 3 diversi livelli di trasparenza (*frammentazione, allocazione e linguaggio*).

Basi di Dati

Docente: Prof. Pierangela Samarati

Seconda prova in itinere - 8 Maggio 2015

Matricola:

Cognome, Nome:

Voto:

Domanda 1

Domanda 2.1

Domanda 2.2

Esercizio 1

Esercizio 2.1

	VSR	CSR
S1		
S2		

Esercizio 2.2

Esercizio 3

Esercizio 4

Posto:

Si indichi per ciascuna affermazione riportata nella tabella allegata se è sicuramente vera (*Vero*), sicuramente falsa (*Falso*), o se *non è possibile determinarlo* (potrebbe essere vera oppure falsa).

1. la transazione T5 è iniziata prima del *checkpoint*
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
2. nel log esistono almeno due operazioni da parte della transazione T4
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
3. il log contiene un record di *commit* per la transazione T6
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
4. il log contiene un record di *abort* per la transazione T4
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
5. la transazione T2 è iniziata (*begin transaction*) e ha fatto *commit* prima del *checkpoint*
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
6. la transazione T3 è iniziata (*begin transaction*) dopo il *checkpoint*
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
7. supponendo che nel log esista una transazione T1, la transazione T1 è iniziata (*begin transaction*) ed ha fatto *abort* o *commit* prima del *checkpoint*
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
8. l'operazione di *delete* di O2 da parte di T2 è l'ultima nel log
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
9. l'operazione di *delete* di O2 segue nella transazione T2 l'operazione di *update* di O1
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo
10. la transazione T5 è iniziata (*begin transaction*) dopo che la transazione T4 è iniziata (*begin transaction*)
 - ☐ Vero
 - ☐ Falso
 - ☐ Non è possibile determinarlo