

**Ricerca di anomalie** : Indicare se le sequenze possono produrre anomalie; i simboli  $c_i$  e  $a_i$  indicano l'esito (commit o abort) della transazione.

1.  $r_1(x), w_1(x), r_2(x), w_2(y), a_1, c_2$
2.  $r_1(x), w_1(x), r_2(y), w_2(y), a_1, c_2$
3.  $r_1(x), r_2(x), r_2(y), w_2(y), r_1(z), a_1, c_2$
4.  $r_1(x), r_2(x), w_2(x), w_1(x), c_1, c_2$
5.  $r_1(x), r_2(x), w_2(x), r_1(y), c_1, c_2$
6.  $r_1(x), w_1(x), r_2(x), w_2(x), c_1, c_2$

**Classificazione di schedule** :

1. Classificare il seguente schedule come non-VSR, VSR o CSR:  
 $r_1(x), r_2(y), w_3(y), r_5(x), w_5(u), w_3(s), w_2(u), w_3(x), w_1(u), r_4(y), w_5(z), r_5(z)$   
 Supponendo di aggiungere lo schedule  $r_2(u), w_2(s)$  come prefisso oppure come suffisso dello schedule precedente, classificare i due schedule risultanti.
2. Analizzare e classificare il seguente schedule:  
 $r_1(x), r_4(x), w_4(x), r_1(y), r_4(z), w_4(z), w_3(y), w_3(z), w_2(t), w_2(z), w_1(t), w_5(t)$   
 Se possibile, aggiungere e togliere una azione in modo da far cambiare classe allo schedule.
3. Si consideri il seguente schedule, in cui l'identificativo di una transazione rappresenta il timestamp della stessa:  
 $r_4(x), r_2(x), w_4(x), w_2(y), w_4(y), r_3(y), w_3(x), w_4(z), r_3(z), r_6(z), r_8(z), w_6(z), w_9(z), r_5(z), r_{10}(z)$   
 Si assuma inizialmente  $RTM = WTM = 0$  per ogni oggetto. Discutere il comportamento del sistema con controllo di concorrenza basato su timestamp mono-versione e multi-versione.  
 Infine, si classifichi lo schedule.

**Protocollo di locking a due fasi** :

1. Verificare se la seguente sequenza di operazioni svolte da tre transazioni su tre risorse  $A, B$  e  $C$  è compatibile con il protocollo di locking a due fasi:  
 $r_1(A), r_2(B), w_1(C), r_2(A), r_1(B), w_2(C), r_3(C), w_2(B), r_3(B), w_1(A), w_3(A)$
2. Verificare se la seguente sequenza di operazioni svolte da tre transazioni su tre risorse  $A, B$  e  $C$  è compatibile con il protocollo di locking a due fasi:  
 $r_1(A), r_2(B), w_1(C), r_2(A), r_1(B), w_2(C), r_3(C), w_2(B), r_3(B), w_2(A), w_3(A)$

**Blocco critico** :

1. Si assuma che  $l_i^{r_j}$  ( $l_i^{w_j}$ ) rappresenti una richiesta di lock in lettura (scrittura) da parte della transazione  $i$  sulla risorsa  $j$ . Sia data la sequenza di richieste di lock:

$$l_1^{rA}, l_2^{rA}, l_3^{wB}, l_4^{rC}, l_2^{wD}, l_3^{wA}, l_1^{wC}, l_5^{wE}, l_5^{rD}, l_4^{rE}$$

Si verifichi se la sequenza di richieste di lock porta ad una situazione di blocco critico.

2. Si verifichi che la sequenza di richieste di lock porta ad una situazione di blocco critico. Si individui inoltre quale transazione può essere fatta terminare per rimuovere la situazione di blocco.

$$l_1^{rA}, l_2^{rA}, l_3^{wB}, l_3^{wA}, l_4^{rC}, l_2^{wA}, l_1^{rB}, l_5^{wE}, l_5^{rB}, l_4^{rE}$$

**Blocco critico distribuito** :

1. Siano date le seguenti condizioni di attesa:

Nodo 1:  $E_4 \rightarrow t_1, t_1 \rightarrow t_2, t_2 \rightarrow E_2$

Nodo 2:  $E_1 \rightarrow t_2, t_2 \rightarrow t_4, t_4 \rightarrow E_3$

Nodo 3:  $E_2 \rightarrow t_4, t_4 \rightarrow t_3, t_3 \rightarrow E_4$

Nodo 4:  $E_3 \rightarrow t_3, t_3 \rightarrow t_1, t_1 \rightarrow E_1$

Determinare se si è in presenza di una situazione di blocco critico.

- Dire se le seguenti condizioni di attesa determinano una situazione di blocco critico:
  - Nodo 1:  $E_2 \rightarrow t_1, t_1 \rightarrow t_2, E_3 \rightarrow t_2, t_2 \rightarrow t_3, t_3 \rightarrow E_2, E_2 \rightarrow t_4, t_4 \rightarrow t_3$
  - Nodo 2:  $E_1 \rightarrow t_3, t_3 \rightarrow t_5, t_5 \rightarrow t_6, t_6 \rightarrow E_3, E_3 \rightarrow t_7, t_7 \rightarrow t_6, t_9 \rightarrow t_4, t_4 \rightarrow E_1, t_1 \rightarrow E_1$
  - Nodo 3:  $E_2 \rightarrow t_6, t_6 \rightarrow t_8, t_8 \rightarrow t_2, t_2 \rightarrow E_1, t_7 \rightarrow E_2$

### Ripristino di un sistema transazionale a partire dal log :

- Si abbia nel log di un sistema di gestione di basi di dati la sequenza di record:  
 Dump,  $b(t_1), b(t_2), b(t_3), i(t_1, o1, a1), d(t_2, o2, b2), b(t_4), u(t_4, o3, b3, a3), u(t_1, o4, b4, a4), c(t_2),$   
 $ckpt(t_1, t_3, t_4), b(t_5), b(t_6), u(t_5, o5, b5, a5), a(t_3), Ckpt(t_1, t_4, t_5, t_6), b(t_7), a(t_4), u(t_7, o6, b6, a6),$   
 $u(t_6, o3, b7, a7), b(t_8), a(t_7),$  guasto  
 Illustrare i passi da compiere per la ripresa a caldo del sistema.
- Individuare gli insiemi di *Undo* e *Redo* per il ripristino di un sistema caratterizzato dal seguente log:  
 Dump,  $b(t_1), u(t_1, o1, b1, a1), b(t_2), b(t_3), u(t_3, o3, a3, b3), i(t_2, o2, a2), c(t_2), Ckpt(t_1, t_3), c(t_3), b(t_4),$   
 $u(t_4, o2, b4, a4), u(t_4, o3, b5, a5), b(t_5), i(t_5, o6, a6), a(t_1), c(t_4), u(t_5, o7, b7, a7), d(t_5, o1, b8),$  guasto

### Ripristino per sistemi distribuiti :

- Applicare il protocollo di ripresa a caldo dopo la caduta di un nodo assumendo un algoritmo di commit a due fasi, a fronte del seguente input (ove  $r(t_i)$  indica la presenza di un record *ready*):  
 $b(t_1), b(t_2), b(t_3), i(t_1, o1, a1), d(t_2, o2, b2), b(t_4), r(t_1), u(t_4, o3, b3, a3), c(t_1), Ckpt(t_2, t_3, t_4), b(t_5),$   
 $b(t_6), u(t_5, o5, b5, a5), r(t_5), b(t_7), u(t_7, o6, b6, a6), b(t_8), u(t_6, o1, b7, a7), a(t_7), r(t_6),$  guasto
- Si consideri il seguente log, trovato su di un nodo di una base di dati distribuita dopo un guasto, in cui  $r(t_1)$  indica che  $t_1$  è in stato *ready* e  $lc(t_1)$ ,  $la(t_1)$  indicano gli stati di *local commit* e di *local abort*.  
 $b(t_1), b(t_2), u(t_1, o1, b1, a1), u(t_2, o2, b2, a2), b(t_3), c(t_2), ckpt(t_1, t_3), b(t_4), r(t_1), b(t_5), i(t_3, o3, a3),$   
 $lc(t_1), d(t_4, o4, b4), d(t_5, o1, b5), r(t_5), u(t_4, o6, b6, a6), a(t_3), b(t_6), u(t_6, o7, b7, a7), c(t_6), r(t_4)$

### Basi di dati distribuite :

Si consideri la base di dati:

PRODUZIONE(NumeroSerie, TipoParte, Modello, Qta, Macchina)  
 PRELIEVO(NumeroSerie, Lotto, Cliente, Venditore, Ammontare)  
 CLIENTE(Nome, Città, Indirizzo)  
 VENDITORE(Nome, Città, Indirizzo)

Progettare la frammentazione orizzontale delle tabelle PRODUZIONE e PRELIEVO in base al tipo di parte (che assume quattro valori: “Keyboard”, “Screen”, “CPU-box” e “Cable”), prevedendo quattro stabilimenti di produzione disposti a Milano, Torino, Roma e Napoli, e delle tabelle CLIENTE e VENDITORE in base a tre bacini di vendita, centrati su Torino, Milano e Roma; si supponga che le vendite siano distribuite per bacini geografici (quindi, clienti di Milano sono serviti solo da venditori di Milano; si assuma che il bacino di vendita di Roma comprenda anche Napoli) e che ciascuna area geografica abbia una propria base di dati (cioè che sia disponibile una base di dati a Milano, Torino, Roma e Napoli). Esprimere a livello di trasparenza di frammentazione, di allocazione e di linguaggio le interrogazioni:

- Determinare la quantità totale prodotta del componente con numero di serie 77Y6878.
- Determinare i clienti che hanno comprato qualche lotto dal rivenditore Bianchi, che ha ufficio a Roma.
- Determinare le macchine utilizzate per la produzione dei pezzi di tipo “Keyboard” venduti al cliente Rossi.
- Modificare l’indirizzo del rivenditore Rossi, che si trasferisce da “Via Po 45” di Milano a “Viale Trastevere 150” di Roma.
- Calcolare la somma degli ammontari degli ordini ricevuti a Milano, Torino e Roma (si noti che anche le funzioni aggregate sono distribuibili).