

**Basi di dati II**  
**Prof. Stefano Paraboschi**  
*I prova - 30/10/2014*

A. Si ha il seguente schedule:

$r_1(X), r_1(Y), r_3(Z), r_3(U), w_2(X), w_1(X), r_2(V), w_1(Z), w_4(V), w_4(X), w_4(U), w_4(Y)$

Classificare lo schedule rispetto alle classi VSR, CSR, 2PL, 2PL-strict, TS-mono e TS-multi. Discutere se é possibile garantire o escludere che l'esecuzione produrrá uno stato finale pari a quello che verrebbe prodotto da un sistema non-concorrente.

B. Si ha un lock manager che, per ridurre al minimo il rischio di starvation, applica una politica per cui un lock non viene concesso a una transazione se questo é in conflitto con il lock richiesto da transazioni che sono in coda in attesa del rilascio del lock corrente. Discutere i vantaggi e gli svantaggi di una soluzione di questo tipo e mostrare quindi la struttura della tabella dei conflitti per un sistema con lock gerarchico che segue questo approccio.

C. Si ha una base di dati distribuita su 3 sedi "Dalmine", "Caniana" e "Salvecchio". Lo schema globale é il seguente:

CORSO(CodCorso, Sede, Titolo, Docente, Cfu)  
PIANOSTUDI(Matr, CodCorso)  
STUDENTE(Matr, Sede, Nome, CorsoLaurea)

Le tabelle CORSO e STUDENTE sono frammentate in base all'attributo Sede. Gli studenti possono inserire nel loro piano degli studi insegnamenti di altre sedi. La tabella PIANOSTUDI é frammentata in base alla frammentazione di STUDENTE.

Scrivere ai livelli di trasparenza di frammentazione e di linguaggio i comandi SQL che:

1. spostano lo studente Mario Rossi dalla sede di "Caniana" alla sede di "Dalmine";
2. cambiano la sede del corso di "Basi di dati 2" da Dalmine a Salvecchio.