

A. (6 punti)

Si consideri la base di dati relazionale:

ARTICOLO(CodArt,Tipo,PrezzoUnit)  
DETTAGLIO(CodArt,CodVen,Quan)  
VENDITA(CodVen,CodCli,Data,PrezzoTot)  
CLIENTE(CodCli,Nome,Indirizzo,Nazione)

Costruire un insieme di regole attive che genera e mantiene consistenti i dati globali di vendita in modo automatico. Si supponga che il codice cliente e la data corrente siano disponibili nelle variabili globali `$client` e `$sysdate`. Si supponga inoltre che ciascun cliente effettui un solo acquisto al giorno. Le regole attive devono reagire all'inserimento di un dettaglio o alla modifica del prezzo unitario di un articolo.

B. (6 punti)

Classificare il seguente schedule:

$r_1(x) w_1(x) r_2(x) r_2(t) r_3(t) r_3(z) w_2(z) r_1(y) r_4(y) w_4(y) w_5(x) w_5(z)$

Se è serializzabile, indicare in base a quale criterio e indicare tutte le sue possibili serializzazioni.

C. (6 punti)

Si consideri la base di dati a oggetti:

```
create class Parte
  attributes CodOrd: Ordine,
             Tipo: string,
             DataRic,DataRip: date;

create class Lavorazione
  attributes Part: Parte
             OreLavorate, Materiale: string;

create class Cliente
  attributes Nome,Indirizzo,PIva,CodFisc: string;

create class Ordine
  attributes CodCli: Cliente
             Importo, NumFatt: integer
             DataInvio, DataPag: date;
```

1. Esprimere in OQL una query che permette di estrarre i nomi e gli indirizzi dei clienti che hanno ordinato delle parti di tipi diversi che richiedono il materiale "Mastice".
2. Trovare gli ordini relativi a parti che richiedono almeno due lavorazioni distinte.

D. (6 punti)

Illustrare brevemente le strutture con accesso calcolato (*hash*).

E. (6 punti)

Spiegare quali sono i criteri di efficienza che hanno influenzato la definizione dell'algoritmo di commit a due fasi e delle sue ottimizzazioni di read-only e di commit o abort presunto.