

A. (6 punti) Si consideri la base di dati a oggetti:

```
class Progetto                                class Persona
{ Nome: string,                               ( Nome: string,
  Partecipante: Ente,                       Affiliazione: Ente )
  Membro: set-of(Persona),
  Budget: integer,
  Revisore: set-of(Persona); )
class Ente
( Nome: string,
  Citta: string )

class Ingegnere inherits Persona
{ AnnoLaurea: integer }
```

1. Estrarre nome e budget dei progetti in cui figurano degli ingegneri sia tra i membri che tra i revisori.
2. Estrarre i nomi dei revisori di progetti che sono affiliati ad enti che non compaiono come affiliazioni di membri di nessun progetto.

B. (6 punti) Dato il seguente schema relazionale:

VENDITA(Codice, PuntoVendita, Prodotto, Ammontare)
CONSUNTIVO(PuntoVendita, Prodotto, Totale)

1. Scrivere una o più regole attive che tengano aggiornato, in modo efficiente, il totale delle vendite per ciascun prodotto e punto vendita a seguito di inserimenti di tuple in VENDITA.
2. Scrivere una o più regole attive che permettano di gestire la presenza di un valore predefinito ALL per l'attributo PuntoVendita nella tabella CONSUNTIVO.

C. (6 punti) Date le seguenti condizioni d'attesa su tre nodi, mostrare l'applicazione dell'algoritmo di deadlock detection per sistemi distribuiti:

Nodo 1 $t_3 \rightarrow E_3; t_4 \rightarrow E_2; t_5 \rightarrow t_4; E_3 \rightarrow t_4;$

Nodo 2 $t_1 \rightarrow E_3; t_2 \rightarrow t_1; t_4 \rightarrow t_2; t_6 \rightarrow t_2; E_1 \rightarrow t_4;$

Nodo 3 $E_1 \rightarrow t_3; E_2 \rightarrow t_1; t_1 \rightarrow t_5; t_5 \rightarrow E_1;$

D. (6 punti) Descrivere le primitive per la gestione del buffer.

E. (6 punti) Descrivere brevemente le ottimizzazioni del protocollo di commit a due fasi.