

A. Si vuole costruire la base di dati per la gestione di un porto per barche a vela e motore. Il porto è organizzato a banchine e ciascuna banchina ha degli spazi disponibili per l'affitto; alcune banchine sono riservate alle barche di passaggio. Le barche vengono identificate tramite un numero di immatricolazione e hanno un responsabile a terra di cui è nota l'identità e un recapito telefonico. Gli affitti degli spazi permanenti vengono pagati a trimestre, in base alla dimensione dello spazio occupato; il sistema di gestione conserva gli avvenuti pagamenti per tutte le barche che affittano uno spazio, con importo e data di pagamento. È possibile rinunciare al rinnovo dell'affitto di uno spazio, che si rende disponibile nel mese successivo. Gli affitti delle barche di passaggio vengono pagati all'atto dell'affitto, in base alle notti previste e alla dimensione dello spazio occupato. È possibile prenotare un pernottamento e assicurarsi così la disponibilità di spazio; la prenotazione deve essere coperta dal pagamento della prima notte.

1. Svolgere il progetto concettuale; si ricorda di specificare un identificatore per ogni entità e cardinalità minima e massima di ogni relazione. (6 punti)
2. Svolgere il progetto logico, descrivendo le chiavi di ogni tabella e i "cammini di join". (3 punti)

B. Si ha il seguente schema relazionale:

LIBRO(CodLibro,Anno,CasaEditrice)
AUTORELIBRO(CodLibro,Autore)
CURATORELIBRO(CodLibro,Editore)

1. Formulare una query SQL che permetta di trovare le persone che sono state allo stesso tempo autori di due libri e curatori di due libri (non necessariamente gli stessi libri). (3 punti)
2. Formulare una query SQL che restituisca l'autore più prolifico. (3 punti)
3. Formulare in algebra relazionale ottimizzata, in Datalog, o in calcolo relazionale l'interrogazione che trova le persone che sono state coinvolte, come autori o curatori, nella preparazione di libri nel 1997, ma non nel 1999. (3 punti)

C. (4 punti) Si consideri la base di dati a oggetti:

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

class Persona
( Nome: string,
  Affiliazione: Ente )

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.

D. (4 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.

E. (3 punti) Descrivere brevemente le ottimizzazioni del protocollo di commit a due fasi.