

**Basi di dati II**  
**Prof. Stefano Ceri**  
*Compito del 7-2-2002*

- A. Specificare algoritmicamente il comportamento di un modulo di sistema che assegni i timestamp con il metodo degli "orologi di Lamport". Il modulo è permanentemente attivo e offre due procedure:

*gettimestamp(T)*

*settimestamp(T)*

La prima è richiamata per ottenere un timestamp (nel parametro  $T$ ), la seconda per segnalare l'arrivo di un messaggio con timestamp di invio  $T$ . Presentare due versioni del modulo, delle quali una utilizza l'orologio di sistema, l'altra una variabile globale che rappresenta il timestamp corrente. Indicare brevemente le applicazioni dei timestamp che vi sono note.

- B. Si ha il seguente schema relazionale che descrive le domande di iscrizione agli asili nido. Si suppone che si possa fare domanda di iscrizione al massimo a 2 asili, specificando un ordine di preferenza ( $NroOrdine$  vale 1 per l'asilo preferito e 2 per la seconda scelta).

ASILONIDO(Nome, NroPosti, Indirizzo, Città, Provincia)  
DOMANDA(NomeAsilo, CodBambino, Data, NroOrdine)  
BAMBINO(CodFisc, Nome, DataNascita, Punteggio)  
ISCRIZIONE(NomeAsilo, CodBambino, Data, RettaMensile)

Si supponga che sia già stata definita una vista PRIMODASASSEGNARE( $CodFisc$ ,  $NomeAsilo$ ) che estrae il codice fiscale del bambino con il valore di  $Punteggio$  più elevato tra tutti quelli non ancora iscritti, associato al nome del primo asilo cui è stata presentata domanda, a condizione che l'asilo abbia ancora disponibilità di posti. (i punteggi sono tutti diversi). Utilizzando la vista PRIMODASASSEGNARE, scrivere una regola attiva che reagisce all'inserimento in ISCRIZIONE. La regola crea una tupla alla volta in ISCRIZIONE, fino a che capita che un asilo che compare come prima scelta ha terminato la propria disponibilità. La computazione viene fatta partire da un inserimento manuale di una prima iscrizione. Si assegni a  $Data$  il risultato della funzione `today()` e a  $RettaMensile$  il valore di default per l'attributo. (6 punti)

- C. Dato il seguente schema a oggetti:

```
create class Persona
  attributes Nome: string,
             Madre: *Persona,
             Padre: *Persona,
             DataNascita: date;

create class Matrimonio
  attributes Moglie: *Persona,
             Marito: *Persona,
             Data: date;
```

1. Estrarre in OQL le coppie di persone sposatesi dopo la nascita di più di 3 figli. (3 punti)
2. Estrarre i matrimoni in cui entrambi i coniugi erano precedentemente sposati. (3 punti)

- D. Descrivere le motivazioni del modello a stella. (6 punti)

- E. Descrivere le caratteristiche fondamentali dei sistemi relazionali a oggetti (object-relational). (6 punti)