

Basi di dati
Prof. Stefano Ceri e Prof. Stefano Paraboschi
Compito del 22-2-1999

A. Si vuole costruire una base di dati per conservare le informazioni relative ai personaggi di fantasia apparsi in fumetti e film di animazione (che chiamiamo collettivamente storie). I personaggi si distinguono in umani e animali. Per i personaggi animali bisogna descrivere la specie e se il personaggio è fortemente antropomorfizzato. I personaggi sono legati tra loro da legami di parentela e di diretta amicizia/rivalità. I personaggi possono inoltre essere legati tra di loro per il fatto di essere apparsi nelle stesse storie. Le storie sono prodotte da autori (disegnatori, scrittori, soggettisti, etc.), che possono avere ruoli diversi in storie diverse e anche più ruoli nella stessa storia. I personaggi vivono in luoghi che possono essere reali, completamente immaginari, o immaginari ma con un corrispondente evidente in luoghi reali. Bisogna anche tener traccia dei luoghi in cui sono ambientate le storie.

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 che descrive la serie di rilevazioni di temperatura per vari luoghi:

LUOGO(Nome, Latitudine, Longitudine, Altezza)
MISURA(NomeLuogo, Ora, Giorno, Mese, Anno, Gradi)

1. Costruire in SQL una vista che presenta la media sui vari anni dei minimi e massimi mensili, per ogni luogo. (3 punti)
2. Formulare in SQL l'interrogazione che restituisce i luoghi posti sotto il livello del mare che hanno presentato qualche giorno una media delle temperature inferiore a 0 gradi centigradi. (3 punti)
3. Formulare in algebra relazionale ottimizzata, calcolo relazionale o Datalog (a vostra scelta), l'interrogazione che restituisce i luoghi posti ad una altitudine superiore ai 1500 metri in cui nel 1998 non si sono mai rilevate temperature inferiori a -10 gradi. (3 punti)

C. Considerare un sistema distribuito su tre nodi con le seguenti condizioni d'attesa:

Nodo 1: $E_3 \rightarrow t_2; t_2 \rightarrow t_1; t_1 \rightarrow E_2$
Nodo 2: $E_1 \rightarrow t_1; t_1 \rightarrow t_3; t_3 \rightarrow E_3; t_4 \rightarrow E_3$
Nodo 3: $E_2 \rightarrow t_3; t_3 \rightarrow t_2; E_2 \rightarrow t_4; t_4 \rightarrow t_2; t_2 \rightarrow E_1$

Determinare la presenza di un deadlock applicando l'algoritmo di rilevamento distribuito, utilizzando obbligatoriamente la convenzione di propagare in avanti le condizioni di attesa in cui la transazione che possiede la risorsa ha il timestamp minore. (4 punti)

```
D. create class Persona                                create class Pilota inherits Persona
  attributes Nome: string;                             attributes CodLicenza: string;

create class Passeggero inherits Persona
  attributes Voli: setof(record(Su: *Volo, Posto: string));

create class PianoVolo                                create class Volo
  attributes CodVolo: string,                           attributes Data: date,
  AeroportoPart: string,                               PrimoPilota: *Pilota,
  AeroportoArr: string,                               SecondoPilota: *Pilota,
  OraPart: time, OraArr: time;                         Piano: *PianoVolo;
```

Dato lo schema di base di dati ad oggetti sopra illustrato, formulare in OQL l'interrogazione che permette di trovare le persone che sono state sia passeggeri che piloti (indifferentemente primo o secondo) in voli caratterizzati dallo stesso piano. (4 punti)

E. Illustrare la specifica e l'uso delle regole di associazione nell'ambito del data mining. (4 punti)