

Compito di Basi di dati e sistemi informativi - II

27 novembre 2000

Esercizio 5:

Si vuole progettare una base di dati per la gestione di un supermercato, contenente le seguenti informazioni:

- per ogni dipendente, il codice identificativo, il nome e il cognome, le eventuali persone a carico, l'indirizzo e il reparto di appartenenza;
- per ogni reparto, il nome, i dipendenti, il responsabile del reparto e gli articoli in vendita;
- per ogni articolo in vendita, il nome, il fornitore, il prezzo di vendita e due codici identificativi (uno assegnatogli dal fornitore, che identifica univocamente l'articolo nell'insieme degli articoli da lui forniti, l'altro dal supermercato, che identifica univocamente l'articolo all'interno del reparto cui è stato assegnato);
- per ogni fornitore, il nome, l'indirizzo e gli articoli che esso fornisce al supermercato (con i relativi prezzi).

Si assuma che, in ogni istante, ogni articolo venga fornito da un solo fornitore e che tale fornitore possa variare nel tempo.

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte. Lo schema dovrà essere completato con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione.

Esercizio 6:

Sia dato il seguente schema relazionale:

COMPONENTI(*Parte#*, *Componente#*);

PRODUTTORI(*Parte#*, *Produttore#*).

La relazione *COMPONENTI* descrive le *Componenti* di ogni *Parte*. Ogni *Componente* di una data *Parte* può essere, a sua volta, scomposta in più *Componenti*. Assumiamo che la relazione *COMPONENTI* non sia transitiva, ossia che da “*y* parte di *x*” e “*z* parte di *y*” non segua “*z* parte di *x*”. La relazione *PRODUTTORI* indica i *Produttori* delle varie *Parti*.

Definire preliminarmente le chiavi delle relazioni date. Successivamente, formulare opportune interrogazioni in SQL che permettano di determinare (senza usare l'operatore *CONTAINS* e usando solo se necessario le funzioni aggregate):

- tutte le parti prodotte dal produttore Liberi, le quali contengano come parte componente la parte X20;
- tutte le parti componenti delle parti prodotte dal produttore Liberi o dal produttore Allineati;
- i produttori che producono tutte le parti componenti della parte X20, ma non producono alcuna parte componente della parte X10;

- (d) i produttori che non producono alcun componente della parte X20, ma che producono almeno una componente di almeno una delle componenti della parte X20;
- (e) le coppie (x, y) di produttori tali che esiste almeno una parte prodotta da x , ma non da y , e viceversa.

Esercizio 7:

Si considerino i seguenti schedule:

$s_1 : r_1(x)r_2(x)w_2(z)w_1(z)r_3(z)w_2(y)r_1(y)w_3(z)w_1(y);$

$s_2 : r_1(z)r_2(x)w_3(x)w_1(z)r_3(z)r_2(z)r_3(y)w_2(z)w_3(y)r_2(y).$

Stabilire se gli schedule dati sono (o meno) view-serializzabili e se sono (o meno) conflict-serializzabili. In caso di risposta positiva, fornire uno schedule seriale equivalente.

Esercizio 8:

Si illustri l'algoritmo per la rilevazione del deadlock distribuito.