

# Basi di Dati e Sistemi Informativi

## Prova scritta del 25 settembre 2001

**Avvertenze:** e' severamente vietato consultare libri e appunti; chiunque verrà trovato in possesso di materiale attinente al corso vedrà annullata la propria prova.

**Durata 2h 15m**

1. Si vuole progettare un sistema informativo per gestire il magazzino di un grossista di alimentari.  
Il sistema deve gestire l'entrata, l'uscita delle merci e tutte le informazioni relative alle spedizioni verso i dettaglianti. Per ogni tipo di merce in magazzino il sistema memorizza: il codice, la descrizione, la quantità disponibile, la scorta minima (quantità al di sotto della quale viene ordinata una nuova fornitura di quel tipo di merce). In magazzino sono presenti inoltre una serie di celle frigorifere. Per la merce nelle celle viene memorizzata anche la temperatura di conservazione.  
Il sistema memorizza per ogni entrata di merce: il codice della merce in entrata, la data, il fornitore, la quantità e la data di scadenza intesa come data entro la quale la merce deve essere consumata. La merce che entra viene suddivisa in blocchi e ogni blocco viene allocato su uno scaffale in una posizione libera. Ogni scaffale è identificato da una lettera univoca e ogni posizione in uno scaffale da un numero progressivo unico per scaffale. Per ogni blocco di merce si conosce la quantità e la data di scadenza della merce.  
Se il blocco di merce va in cella frigorifera la sua posizione è individuata da una lettera greca per la cella e da un numero per la posizione nella cella.  
In ogni istante il sistema deve poter fornire la situazione di occupazione degli scaffali e delle celle del magazzino e deve essere in grado di evidenziare le posizioni libere.  
Per ogni uscita di merce il sistema memorizza: la data, il dettagliante, la quantità. La merce che esce viene prelevata da uno o più posizioni sugli scaffali o nelle celle: ciò implica l'aggiornamento delle quantità dei blocchi sugli scaffali/celle e l'eventuale creazione di posizioni libere sugli scaffali/celle se un blocco viene esaurito.  
Il sistema memorizza infine i dati sui dettaglianti: codice univoco, ragione sociale, indirizzo, città.  
Il sistema mantiene l'archivio storico dell'allocazione della merce in magazzino. E' quindi possibile ricostruire la situazione del magazzino in ogni istante del passato.

*Progettare lo schema concettuale utilizzando il modello entità-relazione e lo schema relazionale della base di dati (indicare esplicitamente per ogni relazione dello schema relazionale: le chiavi primarie, gli attributi che possono contenere valori nulli e i vincoli di integrità referenziale).*

2. Dato lo schema relazionale dell'esercizio 1, esprimere in algebra relazionale ottimizzata le seguenti interrogazioni:
  - 2.a *Trovare le posizioni libere nella cella "alfa", riportando il numero identificativo delle posizioni libere.*
  - 2.b *Trovare la data e il fornitore della merce uscita oggi dal magazzino riportando anche il codice e la descrizione della merce.*
  - 2.c *Trovare il codice e la descrizione della merce della quale ieri si sono avute entrate ma non uscite.*
3. Dato il seguente schema di base di dati (chiavi sottolineate) relativo alle strade comunali, provinciali e statali e ai comuni d'Italia (STRADA.Tipo  $\in$  {S, P, C} dove S=statale, P=provinciale, C=comunale):  
STRADA(Codice, Tipo, Lunghezza); COMUNE(Nome, Abitanti, Superficie, Regione)  
ATTRAVERSA(Comune, Strada)  
Vincoli di integrità referenziale:  
ATTRAVERSA.Comune  $\rightarrow$  COMUNE, ATTRAVERSA.Strada  $\rightarrow$  STRADA  
Formulare in SQL le seguenti interrogazioni (definire viste dove è necessario):
  - 3.a *Trovare il nome, gli abitanti e la superficie dei comuni attraversati da strade statali.*
  - 3.c *Trovare per ogni comune il numero di strade statali, provinciali, comunali che lo attraversano.*
  - 3.b *Trovare il codice e la lunghezza delle strade statali che non attraversano comuni della Lombardia.*
4. Definire formalmente l'operatore di theta join dell'algebra relazionale.
5. Illustrare la struttura di un DBMS indicando i vari moduli presenti, le funzionalità di ogni modulo e le proprietà delle transazioni che vengono garantite da ogni modulo.