

Basi di Dati

Corso di Laurea in “Informatica”

13 giugno 2005

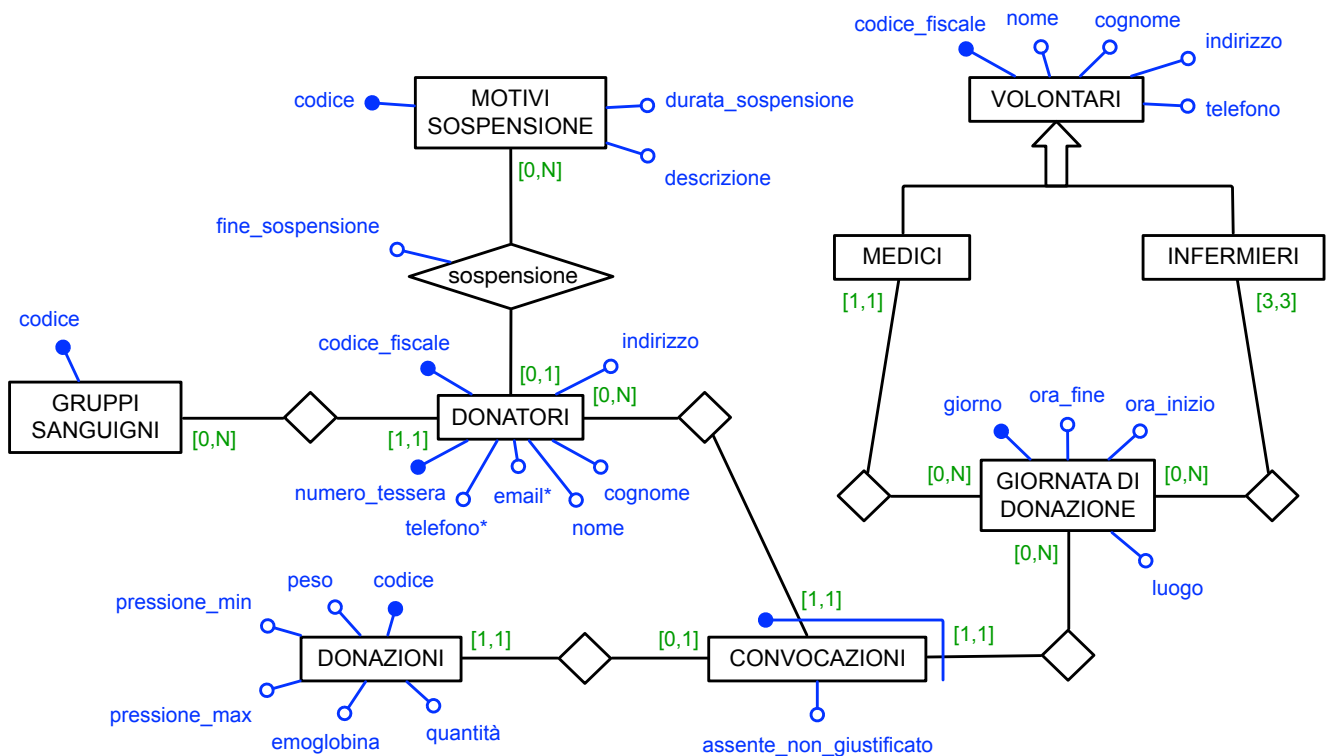
Note

1. Su tutti i fogli contenenti le soluzioni indicare, IN STAMPATELLO, la data dell'appello ed il proprio cognome, nome e numero di matricola
2. Non è consentita la consultazione di alcunché.
3. L'orario di consegna scritto alla lavagna è tassativo.
4. Il testo del compito va consegnato insieme ai fogli con le soluzioni.

Esercizi

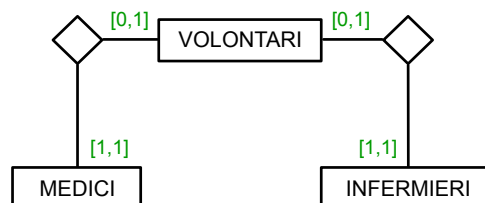
1. Mostrare lo schema concettuale per la base di dati di una sezione di una associazione di donatori di sangue, secondo le seguenti specifiche.
 - (a) Alla sezione afferiscono un certo numero di soci donatori; per essi si registrano il numero della tessera, il gruppo sanguigno (A+, 0+, B-, AB+, ecc.), la data delle ultime analisi di controllo, il codice fiscale, nome e cognome, data di nascita, indirizzo, nonché telefono ed email, se disponibili.
 - (b) Un donatore può essere sospeso per un periodo determinato di tempo stabilito in base al motivo della sospensione (intervento chirurgico, terapia antibiotica, gravidanza, allattamento, ecc.).
 - (c) Alla sezione afferiscono un certo numero di medici, infermieri ed altro personale volontario: per essi si devono registrare il codice fiscale, nome e cognome, indirizzo e telefono.
 - (d) L'attività di donazione della sezione è organizzata in “giornate di donazione”. Per ogni giornata, identificata dalla data, si registrano gli orari di inizio e fine attività ed il luogo.
 - (e) Per regolamento, ad ogni giornata debbono garantire la presenza un medico e tre infermieri: di questi va obbligatoriamente registrata la partecipazione (non interessa registrare la partecipazione di altro personale volontario).
 - (f) Per ogni giornata sono convocati (con un certo anticipo) un certo numero di donatori; di questi, a posteriori, interesserà distinguere coloro che hanno risposto alla convocazione: a tal fine, si considerano “assenti non giustificati” i convocati che non si sono presentati e neppure hanno comunicato l'impossibilità di presentarsi. In generale, un sottoinsieme di coloro che si sono presentati ha effettuato la donazione vera e propria (perché alcuni presenti sono considerati temporaneamente inidonei alla donazione dopo la visita medica).
 - (g) Le donazioni sono identificate da un codice; per ogni donazione si debbono registrare il peso in Kg, la pressione arteriosa (massima e minima, entrambe intere positive) e l'emoglobina del donatore (un valore tra 10.0 e 20.0), la quantità di sangue donata (di norma 400 cc).
2. Tradurre lo schema concettuale dell'esercizio precedente in uno schema logico relazionale, codificando opportunamente i vincoli dello schema.

Diagramma ER



- Non tengo lo storico delle sospensioni.
- Uso il “codice_fiscale” per sapere quali volontari sono anche donatori, come se fosse una “uno a uno”. Ho rindondanza (codice_fiscale, nome, cognome, telefono) ma non è un gran problema quindi lo lascio.
- L’associazione [0,N] a [3,3] la si dovrebbe riscrivere come tre associazioni [0,N] a [1,1]; resta un problema: potrei avere lo stesso infermiere in tutte e 3 le associazioni (serve un vincolo di enunupla in “giornata di donazione”).
- “assenza_non_justificata” all’inizio è false.
- emoglobina è un numeric(3,1).

Ristrutturo schema:



Scelgo le primary key dove ho entità con più identificatori.

Scelgo: numero_tessera, codice.

Schema Logico (e creo tabelle in SQL)

-- gruppi_sanguigni(_codice_)

```
create table gruppi_sanguigni (  
    codice varchar(10) primary key not null  
);
```

-- motivi_sospensione(_codice_, descrizione, durata)

```
create table motivi_sospensione(  
    codice integer primary key not null,  
    descrizione varchar(100) not null,  
    durata_sospensione integer not null,  
    check (durata>0)  
);
```

-- donatori (_numero_tessera_, codice fiscale(uk), nome, cognome, indirizzo, tel*, email*,
 gruppo_canguigno(fk), motivo_sospensione(fk)*, fine_sospensione*)

```
create table donatori(  
    numero_tessera integer not null primary key,  
    codice_fiscale char(16) not null unique,  
    nome varchar(50) not null,  
    cognome varchar(50) not null,  
    indirizzo varchar(250) not null,  
    telefono varchar(12) null,  
    email varchar(50) null,  
    gruppo_sanguigno varchar(10) not null references gruppi_sanguigni(codice),  
    motivo_sospensione integer null references motivi_sospensione(codice),  
    fine_sospensione date null,  
    check( motivo_sospensione is null se e solo se fine_sospensione is null )  
    check( (motivo_sospensione is null and fine_sospensione is null) or  
        (motivo_sospensione is not null and fine_sospensione is not null)  
    )  
);
```

-- volontari(_codice_fiscale_, nome, cognome, telefono, indirizzo)

```
create table volontari(  
    codice_fiscale char(16) not null primary key,  
    nome varchar(50) not null,  
    cognome varchar(50) not null,  
    indirizzo varchar(250) not null,  
    telefono varchar(12) not null,  
);
```

```
create table medici(  
    codice_fiscale char(16) not null primary key references volontari(codice_fiscale)  
);
```

```
create table infermieri(  
    codice_fiscale char(16) not null primary key references volontari(codice_fiscale)  
);
```

```
-- giornate_di_donazione (_giorno_, ora_inizio, ora_fine, luogo, medico(fk),  
                           inf1(fk), inf2(fk), inf3(fk))
```

```
create table giornate_di_donazione(  
    giorno date not null primary key,  
    ora_inizio time not null,  
    ora_fine time not null,  
    check (ora_inizio < ora_fine),  
    luogo varchar(200) not null,  
    medico char(16) not null references medici(codice_fiscale),  
    inf1 char(16) not null references infermieri(codice_fiscale),  
    inf2 char(16) not null references infermieri(codice_fiscale),  
    inf3 char(16) not null references infermieri(codice_fiscale),  
    check( inf1 != inf2 and inf2 != inf3 and inf1 != inf3 ),  
    check( medico!= inf1 and medico != inf2 and medico != inf3 )  
);
```

```
-- convocazioni (_giornata_(fk), _donatore_(fk), assente_ng bool)
```

```
create table convocazione(  
    giornata date not null references giornate_donazione(giorno),  
    donatore integer not null references donatori(numero_tessera),  
    primary key( giornata, donatore),  
    assente_non_giustificato boolean not null default false  
);
```

```
-- donazioni(_codice_, peso, pressione_min, pressione_max, emoglobina, quantità,  
             [giornata, donatore](fk)(uk) )
```

```
create table donazioni(  
    codice integer not null primary key,  
    peso integer not null,  
    pressione_min integer not null,  
    pressione_max integer not null,  
    quantità integer not null default 400,  
    emoglobina numeric(3,1) not null,  
    giornata date not null,  
    donatore integer not null,  
    foreignkey(giornata ,donatore),  
    references convocazione (giornata, donatore),  
    unique(giornata, donatore),  
    check(presmin>0 and press_min<press_max),  
    check(peso>30),  
    check(emoglobina between 100 and 200)  
);
```