

Informatica per le Scienze Umane - Corso di Laurea in Lettere  
Appello del 15.01.2009 - Docente: Massimo Coppola  
**Modello Relazionale e SQL**

Si consideri la seguente base di dati relativa alla redazione di una rivista, che deve tenere traccia degli articoli pubblicati, dei loro autori, delle foto utilizzate e dei fotografi.

TABLE Collaboratori ( codice: char(8) PRIMARY KEY, nome: char(30), cognome: char(30), freelance: boolean, lingua: char(2), indirizzo: char(100) telefono: char(20) )	TABLE Fotografie ( codice: char(12) PRIMARY KEY, autore: char(8) REFERENCES Collaboratori(codice), data_produzione: date, data_acquisto: date, anno_produzione: integer, tema: char(100), luogo: char(100) )
TABLE ImmaginiUsate( articolo: char(12) REFERENCES Articoli(codice) NOT NULL, foto: char(12) REFERENCES Fotografie(codice) NOT NULL, PRIMARY KEY(articolo, foto) )	TABLE Articoli ( codice: char(12) PRIMARY KEY, autore: char(8) REFERENCES Collaboratori(codice), anno_publicazione: integer, data_publicazione: date )

Nella descrizione adottata, assumiamo che:

- l'autore di un articolo è uno solo; un articolo può avere un numero qualsiasi di foto di autori diversi tra loro; le foto possono essere anche usate più volte per articoli differenti;
- alcuni articoli sono traduzioni su licenza di articoli di riviste straniere;
- alcuni collaboratori sono sia giornalisti che fotografi; possono risultare autori di articoli e di fotografie, sia in occasioni diverse, sia come fotoreporter che producono un articolo assieme alle foto relative.
- le date possono essere confrontate tra loro con i consueti operatori < , > , = etc.
- per semplicità, in qualche caso è riportato separatamente l'anno di una data come integer.

Si formulino le seguenti interrogazioni tramite il linguaggio SQL:

1. Elencare gli autori di articoli pubblicati nel 2008. **(punti 5)**
2. Elencare gli articoli che includono una foto scattata prima del 1950 ed una dopo il 2000. **(punti 5)**
3. Elencare i fotoreporter, ossia i collaboratori che hanno scritto articoli di cui sono anche autori di almeno una foto. **(punti 6)**
4. Elencare le foto dei freelance relative alla guerra in Libano che non sono usate in nessun articolo. **(punti 7)**
5. Si modifichi la base di dati in modo da includere per ogni articolo l'indicazione se si tratta di un articolo tradotto, e l'indicazione del traduttore dell'articolo (che farà parte dei collaboratori della rivista). **(punti 4)**
6. Si elenchino gli articoli per i quali sono state riutilizzate fotografie di articoli già pubblicati in precedenza. **(punti 5)**

Si considerino i seguenti fatti riguardanti la presenza di persone ed aziende sul territorio:

- Ogni *persona* è caratterizzata da un proprio codice fiscale, il proprio nome, la data di nascita, nonché dal quartiere in cui ha la residenza e l'azienda per cui lavora.
- Ogni *quartiere* è caratterizzato da un codice identificativo numerico, la propria denominazione, nonché dalle persone che vi risiedono e dalle aziende che vi hanno una filiale.
- Ogni *azienda* è caratterizzata da un proprio codice identificativo, dal proprio nome, dal settore in cui opera (alimentare, informatica, ecc.), nonché dalle persone che vi lavorano e dai quartieri (più di uno, in generale) in cui hanno una filiale.

1. Si rappresentino i fatti sopra descritti in uno schema concettuale UML **(9 punti)**
2. Si traduca lo schema concettuale in uno schema relazionale **(9 punti)**
3. Si fornisca un esempio di istanza dello schema contenente due aziende, due persone e due quartieri **(4 punti)**
4. Si rappresentino tutte le informazioni relative ad una delle persone rappresentate nell'istanza realizzata come un documento XML **(8 punti)**
5. Si fornisca il DTD per il documento XML di cui al punto precedente **(2 punti)**

## Traccia di Soluzione della parte I

### Risposta 1)

```
SELECT DISTINCT codice, nome, cognome
FROM Collaboratori JOIN Articoli ON codice=autore
WHERE anno_publicazione = 2008;
```

### Risposta 2)

Per completezza si forniscono autore e data degli articoli, sarebbe sufficiente il solo codice.

```
SELECT Articoli(codice), Articoli(autore), Articoli(data_publicazione)
FROM Articoli JOIN ImmaginiUsate ON codice = articolo
      JOIN Fotografie ON foto = Fotografie(codice)
WHERE anno_produzione < 1950
      INTERSECT
SELECT Articoli(codice), Articoli(autore), Articoli(data_publicazione)
FROM Articoli JOIN ImmaginiUsate ON codice = articolo
      JOIN Fotografie ON foto = Fotografie(codice)
WHERE anno_produzione > 2000;
```

### Risposta 3)

```
SELECT DISTINCT Collaboratori(codice), Collaboratori(nome), Collaboratori(Cognome)
FROM ImmaginiUsate JOIN Fotografie ON foto = codice
      JOIN Articoli ON articolo = Articoli(codice)
      JOIN Collaboratori ON Articoli(autore) = Collaboratori(codice)
WHERE Fotografie(autore)=Articoli(autore);
```

### Risposta 4)

```
SELECT Fotografie(codice), Fotografie(autore)
FROM Collaboratori JOIN Fotografie ON codice = autore
WHERE tema = "Guerra" AND luogo = "Libano" AND freelance = TRUE
      EXCEPT
SELECT Fotografie(codice), Fotografie(autore)
FROM ImmaginiUsate JOIN Fotografie ON foto = codice;
```

### Risposta 5)

Diamo due possibili soluzioni all'esercizio, una che distingue autori da traduttori, l'altra che assume che di un articolo tradotto ci interessi solo registrare il traduttore.

Nella prima soluzione, la tabella articoli viene estesa con un campo traduttore, che fa riferimento ad un collaboratore della rivista.

In questa soluzione si è scelto di usare come indicazione della traduzione dell'articolo il fatto che il campo traduttore sia non NULL (ovvero, un articolo che non è una traduzione deve avere il campo traduttore impostato a NULL). Va inoltre notato che nella tabella articoli il campo autore può essere impostato a NULL se, come probabile, l'autore dell'articolo originale non è un collaboratore della rivista.

Nota bene: per completezza aggiungiamo un campo che fornisce il riferimento all'originale, che non è strettamente necessario né richiesto per l'esercizio; necessariamente, tale campo è

un testo libero, non potendosi considerare un link interno alla nostra base dati.

```
TABLE Articoli {  
.....  
traduttore: char(8) REFERENCES Collaboratori(codice),  
articoloOriginale: varchar(120)  
}
```

Alternativamente, possiamo assumere che in caso di articolo tradotto il traduttore sia registrato nel campo autore; in questo caso, è necessario aggiungere un campo booleano che indichi se l'articolo è una traduzione, e di conseguenza come va interpretato il campo autore. Notiamo che (1) in uno schema UML questo corrisponderebbe ad aggiungere una sottoclasse di collaboratore; (2) questo secondo tipo di estensione non consente più di distinguere tra giornalista e fotoreporter come richiesto dalla domanda successiva, dato che le sottoclassi di collaboratore aumentano e non sono disgiunte.

```
TABLE Articoli {  
.....  
articolo_tradotto: boolean  
}
```

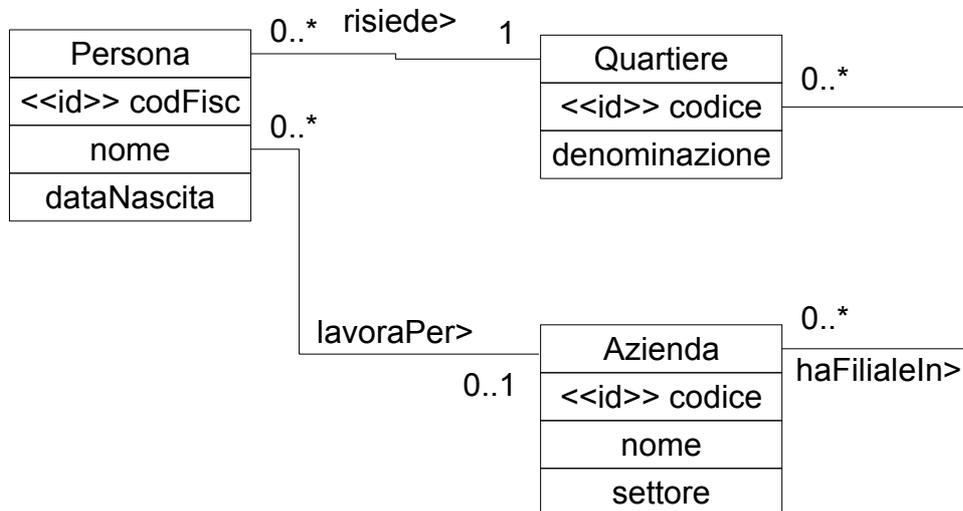
### **Risposta 6)**

Per semplicità definiamo una tabella temporanea che collega gli articoli pubblicati alle foto in essi usate, scartando le informazioni non essenziali. Tramite un JOIN della tabella temporanea con se stessa possiamo ottenere tutte le foto pubblicate in due articoli (più un certo numero di duplicati, ovviamente). La clausola WHERE ci restituisce solo le foto pubblicate in un articolo ed in uno precedente (e quindi diverso). Notare che (1) con questa soluzione non vengono restituite le foto pubblicate in due diversi articoli nella stessa data; (2) la clausola DISTINCT nella seconda SELECT elimina i duplicati dalla lista degli articoli, incluso il caso in cui un articolo faccia uso di più di una foto di repertorio.

```
CREATE TABLE ArticoliFoto =  
SELECT Articoli(codice) AS artcodice, Articoli(data_publicazione), Fotografie(codice)  
AS fotocodice  
FROM Articoli JOIN ImmaginiUsate ON codice = articolo  
JOIN Fotografie ON foto=Fotografie(codice);  
  
SELECT DISTINCT ArticoliFoto(artcodice), ArticoliFoto(data_publicazione)  
FROM ArticoliFoto JOIN ArticoliFoto AS ArticoliFoto2 ON fotocodice = fotocodice  
WHERE ArticoliFoto(data_publicazione) > ArticoliFoto2(data_publicazione);
```

## Traccia di Soluzione della parte II Risposta 1)

Schema concettuale UML



## Risposta 2)

Si traducono i link 1-m come semplici riferimenti nella tabella Persone, che possono essere NULL solo se la cardinalità minima dal lato opposto è 0. L'associazione tra aziende e quartieri è più generale, e richiede una tabella ausiliaria. Nella traccia di soluzione assumiamo di tenere traccia delle persone che vivono nei quartieri presenti nella base dati, e che quindi una persona possa non lavorare per alcuna azienda contenuta nella base dati.

```
TABLE Persone {
    codFisc: char(16) PRIMARY KEY,
    nome: char (100) NOT NULL,
    dataNascita: date,
    risiedeIn: integer REFERENCES Quartieri(codice) NOT NULL,
    lavoraPer: char(8) REFERENCES Aziende(codice)
}
```

```
TABLE Quartieri {
    codice: integer PRIMARY KEY,
    denominazione: char (50)
}
```

```
TABLE Aziende {
    codice: char(8) PRIMARY KEY,
    nome: char (80),
    settore: char(30)
}
```

```

TABLE haFilialeIn{
  azienda: char(8) REFERENCES Aziende(codice),
  quartiere: integer REFERENCES Quartieri(codice),
  PRIMARY KEY (azienda, quartiere)
}

```

### Risposta 3)

Esempio di istanza della base dati.

#### Persone

codFisc	nome	dataNascita	risiedeIn	lavoraPer
MRIRSS00E11Z123Z	Mario Rossi	01/01/75	2	AZA00000
PLORSS11D22X123X	Paolo Rossi	02/02/85	3	AZB00000

#### Quartieri

codice	denominazione
2	Borgata Aurelia
3	Porta a Mare

#### Aziende

codice	nome	settore
AZA00000	Autofficine Bianchi	autofficine
AZB00000	Bottega del gelato di Verdi	ristorazione

#### HaFilialeIn

azienda	quartiere
AZA00000	3
AZA00000	2
AZB00000	2

### Risposta 4)

In questa traccia si è scelto di memorizzare i codici degli elementi per mezzo di tag XML e non come attributi. Notare che in questa soluzione si sceglie di codificare la relazione molti a molti dall'azienda al quartiere, seguendo il verso della relazione in UML.

```

<?xml version="1.0"?>
<Persona>
  <codFisc>PLORSS11D22X123X</codFisc>
  <nome>Paolo Rossi</nome>
  <dataNascita>02/02/85</dataNascita>
  <risiedeIn>
    <Quartiere>
      <codice>3</codice>
      <denominazione>Porta a Mare</denominazione>
    </Quartiere>
  </risiedeIn>

```

```

<lavoraPer>
  <Azienda>
    <codice>AZB00000</codice>
    <nome>Bottega del gelato di Verdi</nome>
    <settore>ristorazione</settore>
    <haFilialeIn>
      <Quartiere>
        <codice>2</codice>
        <denominazione>Borgata Aurelia</denominazione>
      </Quartiere>
    </haFilialeIn>
  </Azienda>
</lavoraPer>
</Persona>

```

### Risposta 5)

Coerentemente con la rappresentazione XML, si codifica la relazione molti a molti dall'azienda al quartiere, seguendo il verso della relazione in UML.

```

<!DOCTYPE Persona [
  <!ELEMENT codFisc      (#PCDATA)>
  <!ELEMENT nome        (#PCDATA)>
  <!ELEMENT dataNascita  (#PCDATA)>
  <!ELEMENT risiedeIn    (Quartiere)>
  <!ELEMENT lavoraPer    (Azienda)>
  <!ELEMENT Quartiere    (Codice,Denominazione)>
  <!ELEMENT Codice       (#PCDATA)>
  <!ELEMENT Denominazione (#PCDATA)>
  <!ELEMENT Azienda      (Codice,Nome,Settore,haFilialeIn+)>
  <!ELEMENT Nome         (#PCDATA)>
  <!ELEMENT Settore      (#PCDATA)>
  <!ELEMENT haFilialeIn  (Quartiere)>
]

```