Informatica per le Scienze Umane - Corso di Laurea in Lettere
Appello del 22.02.2008 - Docente: Mirco Nanni

Parte 1: modello relazionale, SQL

Si consideri la seguente base di dati relazionale che descrive la organizzazione di una azienda di servizi, composta di reparti dislocati sul territorio:

TABLE Reparti (Codice: char(10) PRIMARY KEY, Nome: varchar(200) NOT NULL, Responsabile: char(6) REFERENCES Impiegati(Codice), Sede: integer REFERENCE Sedi(Codice))	TABLE Impiegati (Codice: char(6) PRIMARY KEY, Nome: varchar(50) NOT NULL, Reparto: char(10) REFERENCES Reparti(Codice), Progetto: char(4) REFERENCES Progetti(Codice), Stipendio: integer, Laureato: boolean)
TABLE Sedi (Codice: integer PRIMARY KEY, Regione: varchar(40) NOT NULL, Città: varchar(50), Indirizzo: varchar(100))	TABLE Progetti (Codice: char(4) PRIMARY KEY, Descrizione: varchar(80) NOT NULL, Budget: integer, Reparto: char(10) REFERENCES Reparti(Codice))

NOTE: Ogni progetto viene gestito da un solo reparto, mentre un reparto puo' gestire più progetti. Ogni impiegato, però, lavora ad un solo progetto.

Si formulino le seguenti interrogazioni tramite il linguaggio SQL oppure l'algebra relazionale.

1. Elencare gli impiegati che lavorano in progetti di almeno 1M€. **(4 punti)**

```
SELECT DISTINCT Impiegati.Nome
FROM Impiegati JOIN Progetti ON Impiegati.Progetto = Progetti.Codice
WHERE Progetti.Budget >= 1M€
```

2. Elencare gli impiegati responsabili di reparti che hanno progetti di almeno 1M€. **(5 punti)**

```
SELECT Impiegati.Nome
FROM Impiegati JOIN Reparti ON Reparti.Responsabile = Impiegati.Codice
        JOIN Progetti ON Reparto.Codice = Progetti.Reparto
WHERE Progetti.Budget >= 1M€
```

3. Elencare i reparti che lavorano solo a progetti di budget inferiore a 100.000€ **(6 punti)**

```
SELECT Reparti.Nome
FROM Reparti JOIN Progetti ON Reparti.Codice = Progetti.Reparto
WHERE Progetti.Budget < 100.000€

EXCEPT

SELECT  Reparti.Nome
FROM Reparti JOIN Progetti ON Reparti.Codice = Progetti.Reparto
WHERE  Progetti.Budget >= 100.000€
```

4. Elencari gli impiegati di Roma che guadagnano quanto il responsabile del proprio reparto. **(6 punti)**

```
SELECT  Impiegati.Nome
FROM  Impiegati    JOIN  Reparti  ON  Reparti.Codice = Impiegati.Reparto
                   JOIN  Sedi  ON  Reparti.Sede = Sedi.Codice
                   JOIN Impiegati AS Imp2 ON Reparti.Responsabile = Imp2.Codice
WHERE  Sedi.Città = "Roma"  AND Impiegati.Stipendio = Imp2.Stipendio
              AND  Impiegati.Codice <> Reparti.Responsabile
```

5. Trovare le sedi (con città e indirizzo) in cui almeno un impiegato risulta essere erroneamente allocato ad un progetto non gestito dal proprio reparto. **(6 punti)**

```
SELECT Sedi.Città, Sedi.Indirizzo
FROM Sedi  JOIN Reparti ON Sedi.Codice = Reparti.Sede
                   JOIN Impiegati ON Reparti.Codice = Impiegati.Reparto
                   JOIN Progetti ON Impiegati.Progetto = Progetti.Codice
WHERE  Progetti.Reparto <> Impiegati.Reparto
```

6. Inoltre, si estenda la base di dati (aggiungendo nuove tabelle e/o modificando quelle esistenti): in modo che un reparto possa avere più di una sede. **(5 punti)**

   E' sufficiente eliminare "Sede" a "Reparti" e aggiungere la seguenta tabella allo schema:

```
TABLE  SediReparti (
        Reparto: char(10) REFERENCES Reparti(Codice),
        Sede: char(4) REFERENCES Sedi(Codice),
        PRIMARY KEY (Reparto, Sede)
)
```

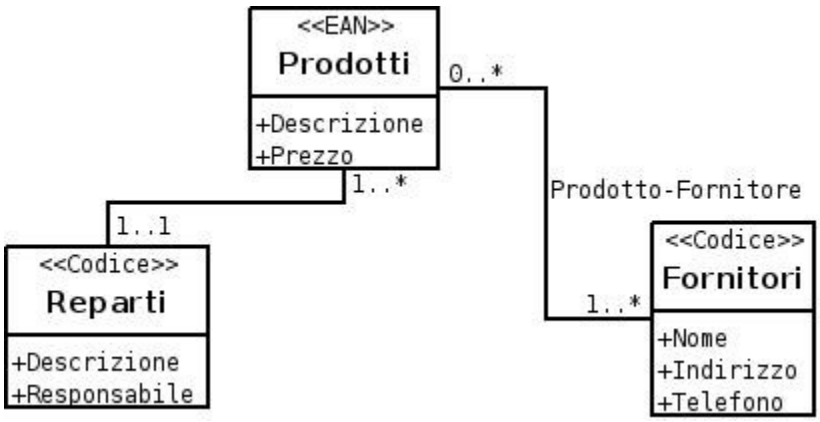
Informatica per le Scienze Umane - Corso di Laurea in Lettere
Appello del 22.02.2008 - Docente: Mirco Nanni

Parte 2: Progetto concettuale e logico, XML

Si considerino i seguenti fatti riguardanti un supermercato:

- ogni ***prodotto*** venduto è caratterizzato dal proprio codice EAN (numerico), la descrizione, il prezzo unitario, il reparto cui appartiene e i suoi fornitori, ovvero le aziende che lo forniscono al supermercato;
- ogni ***fornitore*** del supermercato è caratterizzato da un proprio codice identificativo, il nome, l'indirizzo, i prodotti forniti;
- ogni ***reparto*** è caratterizzato dal proprio codice, dalla descrizione e dal nome del responsabile di reparto, oltre che dai prodotti che vi fanno parte.

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)**

TABLE Prodotti (EAN: Integer PRIMARY KEY, Descrizione: varchar(200), Prezzo: Integer NOT NULL, Reparto: Integer REFERENCES Reparti(Codice))	TABLE Reparti (Codice: Integer PRIMARY KEY, Descrizione: varchar(50) NOT NULL, Responsabile: varchar(100))

TABLE Fornitori (Codice: Integer PRIMARY KEY, Nome: varchar(200) NOT NULL, Indirizzo: varchar (200) NOT NULL)	TABLE ProdottiFornitori (Prodotti: Integer REFERENCES Prodotti(Codice), Fornitori: Integer REFERENCES Fornitori(Codice), PRIMARY KEY (Prodotti, Fornitori))

3. Si costruisca un esempio di istanza della base di dati composta da tre prodotti, due fornitori e due reparti. **(4 punti)**

Prodotti

EAN	Descrizione	Prezzo	Reparto
8382222425	Pasta	0,86	00002
8382939229	Riso	1,85	00002
3453343343	Carne	3,50	00045

Reparto

Codice	Descrizione	Responsabile
000002	Pasta	Guido Rossi
000045	Carni	Carlo Pisani

Fornitori

Codice	Nome	Indirizzo
0992322239	Ingrosso Carni	V. Flaminia, 8 - Pistoia
2342342234	D tutto D+	V. Lami, 113 - Milano

ProdottiFornitori

Fornitori	Prodotti
0992322239	8886140754
2342342234	8382939229
2342342234	3453343343

4. Si costruisca un documento XML relativo ad uno dei prodotti del punto 3, che rappresenti cioè tutte le informazioni collegate al prodotto. **(8 punti)**

```
<Prodotto>
  <EAN>8886140754</EAN>
  <Descrizione>Carne</Descrizione>
  <Prezzo>1,80</Prezzo>
  <Reparto>
        <Codice>000045</Codice>
        <Descrizione>Carni</Descrizione>
        <NomeResponsabile>Carlo Pisani</NomeResponsabile>
  </Reparto>
  <Fornitore>
        <Codice>0992322239</Codice>
        <Nome>Ingrosso Carni</Nome>
        <Indirizzo>V. Flaminia, 8 - Pistoia</Indirizzo>
  </Fornitore>
  </Prodotto>
```

5. Si dia un DTD (Document Type Definition = definizione del tipo di documento) per il documento XML del punto 4. **(2 punti)**

```
<!DOCTYPE Prodotto [
    <!ELEMENT Prodotto (EAN, Descrizione, Prezzo, Fornitore+,Reparto)>
    <!ELEMENT EAN (#PCDATA)>
    <!ELEMENT Descrizione (#PCDATA)>
    <!ELEMENT Prezzo (#PCDATA)>
    <!ELEMENT Fornitore (Codice, Nome, Indirizzo)>
    <!ELEMENT Reparto (Codice, Nome, Indirizzo)>
    <!ELEMENT Codice (#PCDATA)>
    <!ELEMENT Nome (#PCDATA)>
    <!ELEMENT Indirizzo (#PCDATA)>
]>
```