

Informatica per le Scienze Umane - Corso di Laurea in Lettere
 Appello del 15.02.2010 - Docente: *Coppola*
Parte I: Modello relazionale e SQL

Si consideri la seguente base dati relativa alla produzione di una fabbrica di liuteria, che deve tenere traccia degli ordinativi degli strumenti e della loro spedizione a vari negozi e privati.

TABLE Strumenti(codice: char(5) PRIMARY KEY, nome: char(20), numeroCorde: integer, materialeCorpo: char(30), materialeFondo: char(30), materialeManico: char(30), amplificato: char(10))	TABLE Modelli (codice: char(15) PRIMARY KEY, strumento: char(5) REFERENCES Strumenti(codice), codiceColore: char(6), finitura: char(8), codiceGrafica: char(10), fuoriCatalogo: boolean, note: varchar (180))
TABLE Destinatari (codice: char(12) PRIMARY KEY, denominazione: char(30), privato: boolean, partitaIVA: char(20), indirizzo: char (120))	TABLE Ordinativi (numero: integer PRIMARY KEY, modello: char (15) REFERENCES Modelli(codice) NOT NULL, destinatario: char(12) REFERENCES Destinatari(codice) NOT NULL, dataOrdine: date, dataPromessa: date, dataConsegna: date)

Nella descrizione adottata, assumiamo che:

- uno stesso modello di strumento può ovviamente far parte di più ordini diversi,
- le date possono essere confrontate tra loro con i consueti operatori < , > , = etc.
- può accadere che una spedizione avvenga dopo la data di consegna promessa
- si assume che una grafica su misura sia identificata da un numero di serie che identifica un particolare progetto grafico in un apposito archivio.

Si formulino le seguenti interrogazioni tramite il linguaggio SQL:

1. Elencare gli strumenti a cinque corde con manico in mogano. **(punti 4)**
2. Elencare i modelli di chitarra con fondo in palissandro e finitura laccata rossa che risultano fuori catalogo. **(punti 5)**
3. Elencare gli ordini di strumenti a 4 corde da privati che non sono stati consegnati in tempo. **(punti 6)**
4. Elencare i clienti che hanno ordinato dal 1/01/2009 un qualsiasi modello di chitarra classica amplificata con microfono o un qualsiasi modello di contrabbasso, ma nessun modello di violino, viola o violoncello. **(punti 7)**
5. Si modifichi la base di dati in modo da includere l'indicazione se per un qualsiasi ordine è stato richiesto un colore dello strumento diverso da quello standard per il modello, quale colore, e se è richiesta una grafica su misura. **(punti 4)**
6. Tenendo conto della risposta alla domanda 5, elencare i clienti che hanno ordinato chitarre a 6 corde ed a 12 corde nello stesso colore e finitura, eventualmente su richiesta. **(punti 7)**

Si considerino i seguenti fatti riguardanti una flotta di autobus e le linee che esse servono:

- Ogni *linea* è caratterizzata da un proprio codice numerico, dalle fermate che gli sono state assegnate e, per ognuna di esse, l'ora in cui la fermata è prevista.
- Ogni *autobus* è caratterizzato da un codice identificativo numerico, il proprio numero di posti a sedere, la linea che sta servendo, l'ultima fermata che ha fatto e l'orario in cui l'ha effettuata (non necessariamente uguale a quello previsto per quella linea).
- Ogni *fermata* è caratterizzata da un proprio codice identificativo, dal proprio nome e dalla presenza o meno di una pensilina, nonché dalle linee che vi si fermano e gli autobus che vi hanno fatto l'ultima sosta.

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)**
(Nota: si rappresentino gli orari con il tipo "time", che ammette valori del tipo "13:15")
3. Si costruisca un esempio di istanza della base di dati contenente 2 linee, tre autobus e 4 fermate. **(4 punti)**
4. Si costruisca un documento XML relativo ad una delle linee descritte nel punto 3, che rappresenti cioè tutte le informazioni ad essa collegate. **(8 punti)**
5. Si dia un DTD (Document Type Definition) per il documento XML del punto 4, coerentemente con il modello concettuale fornito al punto 1. **(2 punti)**

Soluzione Parte 1

1) È sufficiente una semplice interrogazione della tabella strumenti

```
SELECT codice, nome
FROM Strumenti
WHERE numeroCorde = 5 and materialeManico = "mogano";
```

2) In questo caso serve il Join di due tabelle, e fare un po' di assunzioni sulla codifica di alcuni dati nella base dati. In questo caso assumiamo che la finitura ed il colore nella tabella modelli abbiano una codifica compatibile con quella usata nella query. È ovviamente corretto anche riportare solo i modelli, purché si effettui il filtraggio dei risultati utilizzando gli attributi necessari della tabella Strumenti.

```
SELECT Strumenti.codice, Strumenti.nome, Modelli.codice
FROM Strumenti JOIN Modelli on codice = strumento
WHERE nome = "Chitarra" AND materialeFondo = "Palissandro"
      AND finitura="Lacca" AND codiceColore = "RED" AND fuoriCatalogo = True;
```

3) Qui occorre una join più complessa (gli ordini, più gli strumenti per sapere il numero di corde, a cui si arriva passando dai modelli, ed infine i destinatari per sapere se si tratta di un privato). Varie scelte dei campi mostrare sono possibili, l'importante è identificare in modo univoco gli ordini che ci interessano.

```
SELECT Ordinativi.numero, Destinatari.codice, Destinatari.denominazione
FROM Strumenti JOIN Modelli ON codice = strumento
      JOIN Ordinativi ON Modelli.codice = Ordinativi.modello
      JOIN Destinatari ON Ordinativi.destinatario = Destinatari.codice
WHERE Strumento.numeroCorde=4 AND Destinatari.privato = True AND
      Ordinativi.dataPromessa < Ordinativi.dataConsegna;
```

4) La condizione di selezione è più complessa, e richiede l'uso di sottoquery combinate tra loro. Attenzione alla corretta espressione delle condizioni logiche, è necessario usare bene le parentesi e conoscere la priorità degli operatori AND e OR.

```
SELECT Destinatari.codice, Destinatari.denominazione
FROM Destinatari JOIN Ordinativi ON Destinatari.codice = Ordinativi.destinatario
      JOIN Modelli ON Ordinativi.modello = Modelli.codice
      JOIN Strumenti ON Modelli.strumento = Strumenti.codice
WHERE dataOrdine > "01/01/2009" AND (Strumenti.nome = "Chitarra Classica" AND
Strumenti.amplificato = "microfono" OR Strumenti.nome = "Contrabasso")
EXCEPT
SELECT Destinatari.codice, Destinatari.denominazione
FROM Destinatari JOIN Ordinativi ON Destinatari.codice = Ordinativi.destinatario
      JOIN Modelli ON Ordinativi.modello = Modelli.codice
      JOIN Strumenti ON Modelli.strumento = Strumenti.codice
WHERE dataOrdine > "01/01/2009" AND (Strumenti.nome = "viola" OR Strumenti.nome =
```

“violino” OR Strumenti.nome = “Violoncello”);

- 5) La modifica richiesta può essere realizzata semplicemente aggiungendo alcuni campi alla tabella ordinativi. Potremmo assumere che se i campi aggiunti hanno valore nullo, ciò indica che non è stata richiesta alcuna modifica del modello base ordinato. Per semplicità di scrittura delle query, conviene seguire il suggerimento implicito nel testo, ed aggiungiamo un attributo booleano che indica esplicitamente la richiesta di modifiche. Riportiamo l'intera tabella ordinativi come modificata. Per i codici colore e grafica sono usate le stesse codifiche già presenti della base dati.

```
TABLE Ordinativi (  
  numero: integer PRIMARY KEY,  
  modello: char (15) REFERENCES Modelli(codice) NOT NULL,  
  destinatario: char(12) REFERENCES Destinatari(codice) NOT NULL,  
  dataOrdine: date,  
  dataPromessa: date,  
  dataConsegna: date,  
  custom: boolean,  
  codiceColoreCustom: char (6),  
  codiceGraficaCustom: char(10)  
)
```

- 6) In questo caso dobbiamo eseguire una join complessa, che per ogni potenziale cliente generi una coppia di ordinativi, che poi verificheremo essere relativi uno ad una chitarra 6 corde, ed uno ad una chitarra 12 corde. Tenendo presente la tabella precedentemente riportata, anche la clausola where risulta complessa, poiché per confrontare i codici di colore dobbiamo tenere presente che nessuno, uno dei due o tutti e due potrebbero essere stati personalizzati.

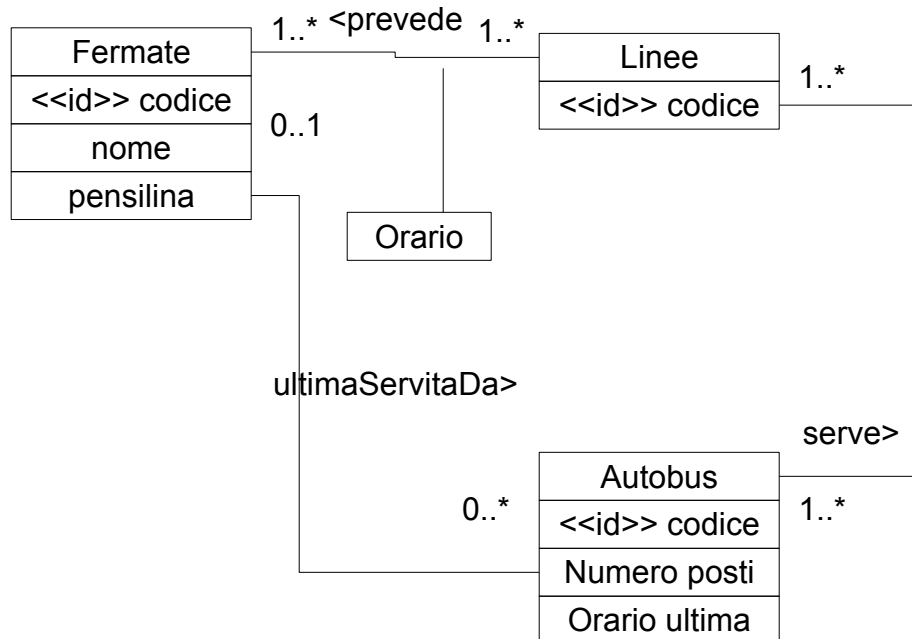
Diamo qui una soluzione che prima isola i clienti che hanno ordinato chitarre a 12 e 6 corde in una tabella temporanea, e quindi determina in quali casi le chitarre ordinate avessero

```
SELECT Destinatari.codice, Destinatari.denominazione  
FROM Destinatari JOIN Ordinativi ON Destinatari.codice = Ordinativi.destinatario  
  JOIN Modelli ON Ordinativi.modello = Modelli.codice  
  JOIN Strumenti ON Modelli.strumento = Strumenti.codice  
  JOIN Ordinativi AS Ordinativi2 ON Destinatari.codice = Ordinativi2.destinatario  
  JOIN Modelli AS Modelli2 ON Ordinativi2.modello = Modelli2.codice  
  JOIN Strumenti AS Strumenti2 ON Modelli2.strumento = Strumenti2.codice  
WHERE Strumenti.nome = “chitarra” AND Strumenti2.nome = ”chitarra”  
  AND Strumenti.numeroCorde = 6 AND Strumenti2.numeroCorde = 12  
  AND Modelli.finitura = Modelli2.finitura  
  AND ( (Ordinativi.custom=False AND Ordinativi2.custom=False AND  
Modelli.codiceColore = Modelli2.codiceColore) OR  
  (Ordinativi.custom=False AND Ordinativi2.custom=True AND  
Modelli.codiceColore = Ordinativi2.codiceColoreCustom) OR  
  (Ordinativi.custom=True AND Ordinativi2.custom=False AND  
Ordinativi.codiceColoreCustom = Modelli2.codiceColore) OR
```

(Ordinativi.custom=True AND Ordinativi2.custom=True AND Ordinativi.codiceColoreCustom = Ordinativi2.codiceColoreCustom));

Soluzione Parte 2

1) Dal testo va capito che una linea definita nella base dati è diversa da come la immaginiamo di solito, poiché passa da ogni fermata solo a un certo orario (e non ad un certo numero di orari stabiliti). Siamo in una situazione semplificata rispetto alla realtà. La relazione tra linee e fermate è comunque molti a molti: una stessa fermata può ospitare più linee (è detto nel testo) e ogni linea ha numerose fermate. La relazione tra autobus e linee è un po' più complicata; un autobus serve certamente una sola linea, è possibile che più autobus servano la stessa linea, quindi la relazione è uno a molti (si ha una situazione irrealistica dato che in questo modo solo uno degli autobus ha la possibilità di arrivare in orario, ma per l'esercizio possiamo ignorare questo dettaglio).



2) In base alle regole di conversione viste a lezione, lo schema UML sopra riportato si traduce nel seguente schema SQL, in cui la relazione “prevede” diventa la tabella ausiliaria FermataSuLinea

```

TABLE Linee {
    codice: integer PRIMARY KEY,
}

```

```

TABLE Autobus {
    codice: integer PRIMARY KEY,
    numeroPosti: integer,
    serve: integer REFERENCES Linee.codice,
    ultimaSosta char(4) REFERENCES Fermate.codice,
    orarioUltimaSosta: time
}

```

```

TABLE Fermate{
    codice: char(4) PRIMARY KEY,
    nome: char(40),
    pensilina: boolean
}

```

```

TABLE FermataSuLinea {
    linea: integer REFERENCES Linee.codice,
    fermata: char(4) REFERENCES Fermate.codice,
    orarioPrevisto: time,
    PRIMARY KEY (linea, fermata)
}

```

3) Esempio di istanza dello schema SQL dato

Linee

Codice
1
2

Autobus

Codice	NumeroPosti	Serve	UltimaSosta	orarioUltimaSosta
5	52	1	57	12:33:00 PM
13	52	2	58	01:40:00 PM
17	44	2	56	12:10:00 PM

Fermate

Codice	Nome	Pensilina
55	Lungarno Sud	F
56	Lungarno Nord	F
57	Cisanello	T
58	Aurelia	T

FermataSuLinea

Linea	Fermata	OrarioPrevisto
1	55	12:00:00 PM
1	56	12:55:00 PM
1	57	12:32:00 PM
2	55	01:05:00 AM
2	56	12:10:00 PM
2	58	12:40:00 PM

4) Esempio di descrizione della Linea 1 secondo l'istanza data.

```
<xml version="1.0"?>
<Linea>
  <codice> 1 </codice>
  <serve>
    <Autobus>
      <codice> 5 </codice>
      <numeroPosti> </numeroPosti>
      <UltimaSosta>
        <Fermata>
          <codice> 57 </codice>
          <nome> Cisanello </nome>
          <pensilina> TRUE </pensilina>
        </Fermata>
      </UltimaSosta>
      <orarioUltimaSosta> 12:33:00 PM </orarioUltimaSosta>
    </Autobus>
  </serve>
  <FermataSuLinea>
    <Fermata>
      <codice> 55 </codice>
      <nome> Lungarno Sud </nome>
      <pensilina> FALSE </pensilina>
    </Fermata>
    <OrarioPrevisto> 12:00:00 PM </OrarioPrevisto>
  </FermataSuLinea>
  <FermataSuLinea>
    <Fermata>
      <codice> 56 </codice>
      <nome> Lungarno Nord </nome>
      <pensilina> FALSE </pensilina>
    </Fermata>
    <OrarioPrevisto> 12:56:00 PM </OrarioPrevisto>
  </FermataSuLinea>
  <FermataSuLinea>
    <Fermata>
      <codice> 57 </codice>
      <nome> Cisanello </nome>
      <pensilina> TRUE </pensilina>
    </Fermata>
    <OrarioPrevisto> 12:32:00 PM </OrarioPrevisto>
  </FermataSuLinea>
</Linea>
```


5) DTD associata all'esempio XML

```
<!DOCTYPE Linea [  
<!ELEMENT Linea (codice,serve+,FermataSuLinea+)>  
<!ELEMENT codice (#PCDATA)>  
<!ELEMENT serve (Autobus)>  
<!ELEMENT FermataSuLinea (Fermata,OrarioPrevisto)>  
<!ELEMENT Autobus (codice,numeroPosti,UltimaSosta,orarioUltimaSosta)>  
<!ELEMENT numeroPosti (#PCDATA)>  
<!ELEMENT UltimaSosta (Fermata)>  
<!ELEMENT orarioUltimaSosta (#PCDATA)>  
<!ELEMENT Fermata (codice,nome,pensilina)>  
<!ELEMENT nome (#PCDATA)>  
<!ELEMENT pensilina (#PCDATA)>  
<!ELEMENT OrarioPrevisto (#PCDATA)>  
>
```