

TESTO

Si consideri la seguente base di dati relazionale che descrive la organizzazione di una biblioteca:

<p>TABLE Libri (Codice: char(10) PRIMARY KEY, Titolo: varchar(200) NOT NULL, Autore: char(6) REFERENCES Autori(Codice), Npagine: integer, Anno: integer, Genere: varchar(30), Lingua: varchar(20), SoloConsultazione: boolean)</p>	<p>TABLE Autori (Codice: char(6) PRIMARY KEY, Cognome: varchar(50) NOT NULL, Nome: varchar(50), AnnoNascita: integer, Nazionalità: varchar(20))</p>
<p>TABLE Lettori (Tessera: integer PRIMARY KEY, Cognome: varchar(50) NOT NULL, Nome: varchar(50), AnnoIscrizione: integer,)</p>	<p>TABLE Prestiti (Numero: integer PRIMARY KEY, Data: date, Lettore: integer REFERENCES Lettori(Tessera), Libro: char(10) REFERENCES Libri(Codice), Restituito: boolean)</p>

NOTE: la tabella Prestiti contiene sia i prestiti in essere che quelli passati. I primi avranno “Restituito=false”, gli altri avranno “Restituito=true”, a indicare che il prestito è terminato. L'attributo Data indica la data di inizio del prestito.

SOLUZIONE

A seguire si fornisce una possibile soluzione per ogni esercizio. Soluzioni alternative sono possibili, per quanto quelle indicate dovrebbero essere le più semplici (salvo errori). Le soluzioni fornite seguono il linguaggio SQL. La traduzione in algebra relazionale è tipicamente banale.

VERSIONE **A**

1. Fornire l'elenco dei lettori che hanno preso in prestito il libro “La cronaca di Travnik”, ordinando tale elenco per data di prestito. **(6 punti)**

```
SELECT Prestiti.Data AS Data, Lettori.Cognome AS CogLet, Lettori.Nome AS NomLet
FROM Lettori JOIN Prestiti ON Lettori.Tessera = Prestiti.Lettore
      JOIN Libri ON Prestiti.Libro = Libri.Codice
WHERE Libri.Titolo = “La cronaca di Travnik”
ORDER BY Prestiti.Data
```

2. Trovare i lettori che non prendono in prestito un libro da almeno un anno. **(6 punti)**

```
SELECT Lettori.Cognome AS CognomeLettore, Lettori.Nome AS NomeLettore
FROM Lettori
```

```
EXCEPT
```

```
SELECT Lettori.Cognome, Lettori.Nome
FROM Lettori JOIN Prestiti ON Lettori.Tessera = Prestiti.Lettore
WHERE Prestiti.Data > '13-12-2006' (Si assume che oggi sia il 13-12-2007)
```

3. Quali autori hanno scritto solo libri di almeno 500 pagine? **(6 punti)**

```
SELECT Autori.Cognome AS CognomeAutore, Autori.Nome AS NomeAutore
FROM Autori JOIN Libri ON Autori.Codice = Libri.Autore
WHERE Libri.Npagine >= 500
```

```
EXCEPT
```

```
SELECT Autori.Cognome, Autori.Nome
FROM Autori JOIN Libri ON Autori.Codice = Libri.Autore
WHERE Libri.Npagine < 500
```

(Ignoriamo il caso in cui Npagine sia NULL)

4. Quali lettori hanno precedentemente preso in prestito uno o più libri attualmente in prestito a Giuseppe Rossi? (7 punti)

```
SELECT DISTINCT Lettori.Cognome AS CogLet, Lettori.Nome AS NomLet
FROM Lettori JOIN Prestiti ON Lettori.Tessera = Prestiti.Lettore
      JOIN Prestiti AS PrestitiGR ON Prestiti.Libro = PrestitiGR.Libro
      JOIN Lettori AS LettoriGR ON PrestitiGR.Lettore = LettoriGR.Tessera
WHERE LettoriGR.Nome= "Giuseppe" AND LettoriGR.Cognome="Rossi"
      AND PrestitiGR.Restituito = false
```

Si noti come la tabella Libri non sia necessaria – la join avviene attraverso il loro “codice”.

5. Si estenda la base di dati (aggiungendo nuove tabelle e/o modificando quelle esistenti) in modo da consentire di avere libri con più di un autore (7 punti)

Aggiungere la seguente tabella:

```
TABLE LibroAutore (
    Libro: char(10) REFERENCES Libri(Codice),
    Autore: char(6) REFERENCES Autori(Codice)
)
```

ed eliminare l'attributo Autore dalla tabella Libri.

VERSIONE **B**

1. Quali autori non sono mai stati presi in prestito? (Ovvero: nessun loro libro è mai stato preso in prestito). (6 punti)

```
SELECT Autori.Cognome AS Cognome, Autori.Nome AS Nome
FROM Autori
```

EXCEPT

```
SELECT Autori.Cognome, Autori.Nome
FROM Autori JOIN Libri ON Autori.Codice = Libri.Autore
      JOIN Prestiti ON Libri.Codice = Prestiti.Libro
```

2. Trovare i lettori che hanno preso in prestito un libro di sola consultazione (evidentemente per un errore della biblioteca). **(6 punti)**

```
SELECT DISTINCT Lettori.Cognome AS Cognome, Lettori.Nome AS Nome
FROM Lettori JOIN Prestiti ON Lettori.Tessera = Prestiti.Lettore
      JOIN Libri ON Prestiti.Libro = Libri.Codice
WHERE Libri.SoloConsultazione = true
```

3. Trovare le coppie di lettori iscritti lo stesso anno e che hanno letto (almeno) uno stesso libro di fantascienza. **(7 punti)**

```
SELECT DISTINCT Lettori.Cognome AS CognomeA, Lettori.Nome AS NomeA,
      Lettori2.Cognome AS CognomeB, Lettori2.Nome AS NomeB
FROM Lettori JOIN Prestiti ON Lettori.Codice = Prestiti.Lettore
      JOIN Libri ON Prestiti.Libro = Libri.Codice
      JOIN Prestiti AS Prestiti2 ON Libri.Codice = Prestiti2.Libro
      JOIN Lettori AS Lettori2 ON Prestiti2.Lettore = Lettori2.Codice
WHERE Lettori.AnnoIscrizione = Lettori2.AnnoIscrizione AND
      Libri.Genere = "Fantascienza"
```

4. Quali libri sono stati presi in prestito sia ieri che oggi? **(6 punti)**

```
SELECT Libri.Titolo AS Titolo
FROM Libri JOIN Prestiti ON Libri.Codice = Prestiti.Libro
WHERE Prestiti.Data = '12-12-2007'
```

INTERSECT

```
SELECT Libri.Titolo AS Titolo
FROM Libri JOIN Prestiti ON Libri.Codice = Prestiti.Libro
WHERE Prestiti.Data = '13-12-2007'
```

5. Si estenda la base di dati (aggiungendo nuove tabelle e/o modificando quelle esistenti) in modo da poter associare ad ogni libro oltre all'autore anche un curatore ed un traduttore. **(7 punti)**

Manteniamo gli autori, i curatori e i traduttori in una unica tabella, uguale alla vecchia Autori. Quindi, aggiungiamo due attributi, Curatore e Traduttore, alla tabella Libri, del seguente tipo:

```
Curatore: char(6) REFERENCES Autori(Codice),
Traduttore: char(6) REFERENCES Autori(Codice)
```

In alternativa, si possono creare nuove tabelle per curatori e traduttori, aggiustando i riferimenti.