

INTERROGAZIONI

```
Libri(CodiceLibro, Titolo, Anno, LinguaLib, NomeEditore*)
AutoriLibri(CodiceLibro *, CodiceAutore*)
Autori(CodiceAutore, Nome, AnnoNascita)
Editori(NomeEditore, Indirizzo, Nazione, Telefono)
```

a) Per ogni autore che ha scritto solo libri in lingua italiana, si riportino il nome e l'anno di nascita dell'autore

```
SELECT a.Nome, a.AnnoNascita
FROM Autori a
JOIN AutoriLibri al1 ON a.CodiceAutore = al1.CodiceAutore
WHERE NOT EXISTS (SELECT *
                   FROM AutoriLibri al2
                   JOIN Libri l ON l.CodiceLibro = al2.CodiceLibro
                   WHERE al2.CodiceAutore = a.CodiceAutore
                   AND l.LinguaLib <> 'Italiano')
```

La JOIN nella query principale mi serve a escludere gli autori che non hanno scritto alcun libro (la quantificazione universale sull'insieme vuoto è sempre vera); se volessi includerla, la terza riga della query è eliminabile senza conseguenze.

b) Per ogni coppia di autori che hanno scritto un libro in comune si riporti il nome del primo ed il nome del secondo

```
SELECT a1.Nome, a2.Nome
FROM Autori a1, Autori a2
JOIN AutoriLibri al1 ON al1.CodiceAutore = a1.CodiceAutore
WHERE a1.CodiceAutore < a2.CodiceAutore
      AND EXISTS (SELECT *
                  FROM AutoriLibri al2
                  WHERE al2.CodiceLibro = al1.CodiceLibro
                  AND al2.CodiceAutore = a2.CodiceAutore)
```

Se due autori hanno scritto più libri in comune, appariranno più volte nel risultato; per evitarlo potrei fare SELECT DISTINCT, ma Nome non è chiave e potrei perdere coppie di omonimi. Le coppie diverse soltanto per ordine degli autori sono escluse invece dalla condizione a1.CodiceAutore < a2.CodiceAutore.

Schema

```
Libri(CodiceLibro, Titolo, Anno, LinguaLib, NomeEditore*)
AutoriLibri(CodiceLibro *, CodiceAutore*)
Autori(CodiceAutore, Nome, AnnoNascita)
Editori(NomeEditore, Indirizzo, Nazione, Telefono)
```

c) Per ogni libro con almeno due autori in cui tutti i suoi autori sono nati lo stesso anno si riportino il titolo e la lingua del libro

```
{l.Titolo, l.LinguaLib | l ∈ Libri, a1 ∈ AutoriLibri, a1.CodiceLibro =
l.CodiceLibro, a1 ∈ Autori, a1.CodiceAutore = a1.CodiceAutore .
  ∀ (a2 ∈ AutoriLibri, a2.CodiceLibro = l.CodiceLibro, a2 ∈ Autori,
a2.CodiceAutore = a2.CodiceAutore) :
    a2.AnnoNascita = a1.AnnoNascita} purché esistano almeno due autori
```

Trasformo il $\forall x.P$ in $\neg\exists.\neg P$ (nego solo la tesi, non l'ipotesi).

```
{l.Titolo, l.LinguaLib | l ∈ Libri, a1 ∈ AutoriLibri, a1.CodiceLibro =
l.CodiceLibro, a1 ∈ Autori, a1.CodiceAutore = a1.CodiceAutore .
  ¬∃ (a2 ∈ AutoriLibri, a2.CodiceLibro = l.CodiceLibro, a2 ∈ Autori,
a2.CodiceAutore = a2.CodiceAutore) :
    a2.AnnoNascita <> a1.AnnoNascita} purché esistano almeno due autori
```

Scrivo quindi la query:

```
SELECT l.Titolo, l.LinguaLib
FROM Libri l
JOIN AutoriLibri al1 ON al1.CodiceLibro = l.CodiceLibro
JOIN Autori a1 ON a1.CodiceAutore = al1.CodiceAutore
WHERE NOT EXISTS (SELECT *
                  FROM AutoriLibri al2
                  JOIN Autori a2 ON a2.CodiceAutore = al2.CodiceAutore
                  WHERE al2.CodiceLibro = l.CodiceLibro
                     AND a2.AnnoNascita <> a1.AnnoNascita)
GROUP BY l.CodiceLibro, l.Titolo, l.LinguaLib
HAVING COUNT(*) > 1
```

Se si ammettono libri omonimi nella stessa lingua (la coppia Titolo, LinguaLib non è dichiarata come chiave nello schema dato, anche se potrebbe esserlo volendo) conviene aggiungere CodiceLibro alle dimensioni di analisi.

Schema

```
Libri(CodiceLibro, Titolo, Anno, LinguaLib, NomeEditore*)
AutoriLibri(CodiceLibro *, CodiceAutore*)
Autori(CodiceAutore, Nome, AnnoNascita)
Editori(NomeEditore, Indirizzo, Nazione, Telefono)
```

d) Per ogni coppia di autori tali che il primo è coautore di tutti i libri del secondo, scrivere il nome del primo e il nome del secondo

```
{a1.Nome, a2.Nome | a1 ∈ Autori, a2 ∈ Autori, a1.CodiceAutore <
a2.CodiceAutore .
  ∀ (al1 ∈ AutoriLibri, al1.CodiceAutore = a1.CodiceAutore) :
    ∃ (al2 ∈ AutoriLibri, al2.CodiceAutore = a2.CodiceAutore) .
al2.CodiceLibro = al1.CodiceLibro}
```

Trasformo il $\forall x.P$ in $\neg\exists.\neg P$ (nego solo la tesi, non l'ipotesi).

```
{a1.Nome, a2.Nome | a1 ∈ Autori, a2 ∈ Autori, a1.CodiceAutore <
a2.CodiceAutore .
  ¬∃ (al1 ∈ AutoriLibri, al1.CodiceAutore = a1.CodiceAutore) :
    ¬∃ (al2 ∈ AutoriLibri, al2.CodiceAutore = a2.CodiceAutore) .
al2.CodiceLibro = al1.CodiceLibro}
```

```
SELECT a1.Nome, a2.Nome
FROM Autori a1, Autori a2
WHERE a1.CodiceAutore < a2.CodiceAutore
      AND NOT EXISTS (SELECT *
                      FROM AutoriLibri al1
                      WHERE al1.CodiceAutore = a1.CodiceAutore
                      AND NOT EXISTS (SELECT *
                                      FROM AutoriLibri a2
                                      WHERE al2.CodiceAutore = a2.CodiceAutore
                                      AND al2.CodiceLibro = al1.CodiceLibro))
```

e) Per ogni nazione, dare la nazione e il numero totale di libri il cui editore è di quella nazione

```
SELECT e.Nazione, COUNT(*) TotaleLibri
FROM Editori e
JOIN Libri l USING (NomeEditore)
GROUP BY e.Nazione
```