

*Corso di Laurea in Ingegneria Gestionale  
Sapienza Università di Roma*

# *Corso di Basi di Dati*

## *A.A. 2016/2017*

E3 – SQL : Esercitazione su  
Interrogazioni nidificate

Francesco Leotta

Ultimo aggiornamento : 16/03/2017

# Esercizio Proposto

---

## ***ESERCIZIO*** : *Siano date le seguenti relazioni*

- ▶ Fornitori(**fid:integer**, **fnome:string**, **indirizzo:string**)
- ▶ Pezzi(**pid:integer**, **pnome:string**, **colore:string**)
- ▶ Catalogo(**fid:integer**, **pid:integer**, **costo:real**)
  
- ▶ Per le relazioni sussistono i seguenti vincoli di integrità:
  - **Fornitori.fid** è **CHIAVE PRIMARIA** di *Fornitori*
  - **Pezzi.pid** è **CHIAVE PRIMARIA** di *Pezzi*
  - **Catalogo.fid** e **Catalogo.pid** sono **CHIAVE PRIMARIA** di *Catalogo*
  - **Catalogo.fid** è **CHIAVE ESTERNA** verso *Fornitori.fid*
  - **Catalogo.pid** è **CHIAVE ESTERNA** verso *Pezzi.pid*

# Esercizio Proposto

---

## **ESERCIZIO** : *Siano date le seguenti relazioni*

- ▶ Fornitori(**fid:string**, **fnome:string**, **indirizzo:string**)
- ▶ Pezzi(**pid:string**, **pnome:string**, **colore:string**)
- ▶ Catalogo(**fid:string**, **pid:string**, **costo:real**)

## **ESERCIZIO** : *Si calcolino le seguenti interrogazioni in SQL*

- ▶ 1. Trovare i *pnome* dei pezzi per cui esiste un qualche fornitore
- ▶ 2. Trovare gli *fnome* dei fornitori che forniscono ogni pezzo
- ▶ 3. Trovare gli *fnome* dei fornitori che forniscono tutti i pezzi rossi
- ▶ 4. Trovare i *pnome* dei pezzi forniti dalla Acme e da nessun altro
- ▶ 5. Trovare i *fid* dei fornitori che ricaricano su alcuni pezzi più del costo medio di quel pezzo
- ▶ 6. Per ciascun pezzo, trovare gli *fnome* dei fornitori che ricaricano di più su quel pezzo

# Esercizio Proposto

---

- ▶ 7. Trovare i *fid* dei fornitori che forniscono solo pezzi rossi
- ▶ 8. Trovare i *fid* dei fornitori che forniscono un pezzo rosso e un pezzo verde
- ▶ 9. Trovare i *fid* dei fornitori che forniscono un pezzo rosso o uno verde
- ▶ 10. Trovare i *pid* dei pezzi forniti da almeno due fornitori

# Esercizio Proposto – 1\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

1) *Trovare i pnome dei pezzi per cui esiste un qualche fornitore*

```
SELECT DISTINCT P.pnome
  FROM Pezzi P, Catalogo C
 WHERE P.pid = C.pid
```

Si effettua un *equi-join* tra il *pid* di **Catalogo** e il *pid* di **Pezzi**. Si ottengono così solo quei pezzi che hanno un fornitore.

Questo perché nella relazione **Pezzi** potrebbero esserci dei pezzi che non sono forniti da nessun fornitore.

# Esercizio Proposto – 2\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

2) Trovare gli *fnome* dei fornitori che forniscono ogni pezzo

```
SELECT DISTINCT F.fnome
FROM Fornitori F
WHERE NOT EXISTS (
```

```
(SELECT P.pid FROM Pezzi P)
EXCEPT
(SELECT C.Pid
FROM Catalogo C
WHERE C.fid = F.fid)
)
```

## A EXCEPT B

- Viene restituita una relazione che contiene tutti i *pid* dei pezzi non forniti dal fornitore i-esimo. Se tale relazione è vuota, significa che il fornitore i-esimo fornisce tutti i pezzi.

In quest'ultimo caso, **NOT EXISTS** restituisce **TRUE** e l'*fnome* dell'i-esimo fornitore considerato apparirà nella relazione risultato.

L'interrogazione interna va valutata e ri-calcolata per ogni tupla di **Fornitori** definita nell'interrogazione esterna.

**A.** (SELECT P.Pid FROM Pezzi P)  
- Calcola i *pid* di tutti i pezzi.

**B.** (SELECT C.Pid FROM Catalogo C WHERE C.fid=F.fid)  
- Calcola i *pid* dei pezzi venduti dal fornitore i-esimo (che stiamo considerando attraverso la variabile di range F definita nell'interrogazione esterna).

# Esercizio Proposto – 3\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

3) *Trovare gli fnome dei fornitori che forniscono tutti i pezzi rossi*

```
SELECT DISTINCT F.fnome
FROM Fornitori F
WHERE NOT EXISTS (
```

## A EXCEPT B

- Viene restituita una relazione che contiene tutti quei pezzi di colore rosso non forniti dal fornitore i-esimo. Se tale relazione è vuota, significa che il fornitore i-esimo fornisce tutti i pezzi esistenti di colore rosso (cioè non vuol dire che il fornitore i-esimo non possa fornire pezzi anche di altro colore...l'importante è che fornisca almeno tutti quelli di colore rosso) In quest'ultimo caso, **NOT EXISTS** restituisce **TRUE** e l'*fnome* dell'i-esimo fornitore considerato apparirà nella relazione risultato.

```
(SELECT * FROM Pezzi P
WHERE P.colore = 'Rosso')
EXCEPT
(SELECT P1.Pid, P1.colore,
P1.pnome
FROM Catalogo C, Pezzi P1
WHERE C.fid = F.fid AND
C.pid = P1.pid)
)
```

L'interrogazione interna va valutata e riccalcolata per ogni tupla di **Fornitori** definita nell'interrogazione esterna.

- A. (SELECT \* FROM Pezzi P)
- Calcola tutti i pezzi di colore rosso
- B. (SELECT P1.Pid, P1.colore, P1.pnome FROM Catalogo C, Pezzi P1 WHERE C.fid=F.fid AND C.pid = P1.pid)
- Calcola i pezzi venduti dal fornitore i-esimo (che stiamo considerando attraverso la variabile di range F definita nell'interrogazione esterna).

# Esercizio Proposto – 4\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

## 4) Trovare i pezzi forniti dalla ACME e da nessun'altra

```
SELECT P.pid, P.pnome, P.colore
FROM Pezzi P, Catalogo C, Fornitori F
WHERE P.pid=C.pid AND C.fid=F.fid AND F.fnome = 'ACME' AND
      NOT EXISTS (SELECT * FROM Catalogo C1, Fornitori F1
                  WHERE P.pid = C1.pid AND C1.fid=F1.fid AND
                        F1.fnome <> 'ACME'
                  )
```

Se la relazione restituita dall'interrogazione interna è **vuota**, NOT EXISTS restituisce TRUE.

A questo punto si verifica che il pezzo i-esimo sia almeno fornito dalla 'Acme'...in quest'ultimo caso, il pezzo considerato apparirà nella relazione risultato.

L'interrogazione interna calcola tutti quei fornitori (il cui nome è diverso da 'Acme') che forniscono il pezzo i-esimo (che stiamo considerando attraverso la variabile di range P definita nell'interrogazione esterna).

Se tale relazione è **vuota**, significa che il pezzo i-esimo non è fornito da alcun fornitore il cui nome è diverso da 'Acme'.

# Esercizio Proposto – 5\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

5) *Trovare i fid dei fornitori che ricaricano su alcuni pezzi più del costo medio di quel pezzo*

```
SELECT DISTINCT C.fid
FROM Catalogo C
WHERE C.costo > (SELECT AVG(C1.costo)
                  FROM Catalogo C1
                  WHERE C1.pid = C.pid
                  )
```

L'interrogazione interna calcola la media del prezzo di vendita del pezzo i-esimo considerato (C.pid).

L'interrogazione esterna, per ciascun pezzo fornito in catalogo, verifica che il prezzo del pezzo considerato sia superiore rispetto al prezzo medio di vendita di quel pezzo (calcolato con l'interrogazione interna).

In quest'ultimo caso, l'interrogazione esterna restituisce il **fid** del fornitore che vende il pezzo i-esimo ad un prezzo superiore alla media.

# Esercizio Proposto – 6\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

6) *Per ciascun pezzo, trovare gli fnome dei fornitori che ricaricano di più su quel pezzo*

```
SELECT C.pid, F.fnome
FROM Fornitori F, Catalogo C
WHERE C.fid = F.fid
      AND C.costo = (SELECT MAX(C1.costo)
                      FROM Catalogo C1
                      WHERE C1.pid = C.pid
                  )
```

L'interrogazione interna calcola il massimo prezzo di vendita del pezzo i-esimo considerato (*C.pid*).

L'interrogazione esterna, per ciascun pezzo fornito in catalogo, verifica che il prezzo del pezzo i-esimo considerato sia uguale al massimo prezzo di vendita di quel pezzo (calcolato con l'interrogazione interna). In quest'ultimo caso, l'interrogazione esterna restituisce il nome del fornitore (e il pid del pezzo-iesimo) che vende il pezzo i-esimo al prezzo massimo.

# Esercizio Proposto – 7\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

7) Trovare i fid dei fornitori che forniscono solo pezzi rossi

```
SELECT F.fid
FROM Fornitori F
WHERE NOT EXISTS (SELECT *
                   FROM Pezzi P, Catalogo C1
                   WHERE C1.fid = F.fid AND
                         C1.pid = P.pid AND
                         P.colore <> 'Rosso'
                  )
AND EXISTS (SELECT *
            FROM Catalogo C1
            WHERE C1.fid = F.fid)
```

La prima interrogazione interna verifica che l'i-esimo fornitore venga pezzi non rossi.

La seconda interrogazione interna verifica che l'i-esimo fornitore venga qualcosa (per evitare che eventuali fornitori che non vendono niente siano inclusi nel risultato).

L'interrogazione esterna inserisce il nome dell'i-esimo fornitore nella relazione risultato se la prima interrogazione interna restituisce una relazione vuota e la seconda interrogazione esterna restituisce una relazione non vuota.

# Esercizio Proposto – 8\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

8) Trovare i *fid* dei fornitori che forniscono almeno un pezzo rosso e un pezzo verde

```
SELECT DISTINCT C.fid
FROM Catalogo C, Pezzi P
WHERE C.pid = P.pid AND P.colore = 'rosso'
INTERSECT
SELECT DISTINCT C1.fid
FROM Catalogo C1, Pezzi P1
WHERE C1.pid = P1.pid AND P1.colore = 'verde'
```

La prima interrogazione calcola i *fid* dei fornitori (senza duplicati) che forniscono almeno un pezzo rosso.

La seconda interrogazione calcola i *fid* dei fornitori (senza duplicati) che forniscono almeno un pezzo verde.

L'INTERSEZIONE restituisce i *fid* dei fornitori che forniscono almeno un pezzo rosso e un pezzo verde.

# Esercizio Proposto – 9\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

9) Trovare i *fid* dei fornitori che forniscono almeno un pezzo rosso o uno verde

```
SELECT DISTINCT C.fid
FROM Catalogo C, Pezzi P
WHERE C.pid = P.pid AND P.colore = 'rosso'
UNION
SELECT DISTINCT C1.fid
FROM Catalogo C1, Pezzi P1
WHERE C1.pid = P1.pid AND P1.colore = 'verde'
```

La prima interrogazione calcola i *fid* dei fornitori (senza duplicati) che forniscono almeno un pezzo rosso.

La seconda interrogazione calcola i *fid* dei fornitori (senza duplicati) che forniscono almeno un pezzo verde.

L'UNIONE restituisce i *fid* dei fornitori che forniscono almeno un pezzo rosso oppure almeno un pezzo verde.

# Esercizio Proposto – 10\10

Fornitori

<u>fid</u>	fnome	indirizzo
------------	-------	-----------

Pezzi

<u>pid</u>	pnome	colore
------------	-------	--------

Catalogo

<u>fid</u>	<u>pid</u>	costo
------------	------------	-------

**10) Trovare i pid dei pezzi forniti da almeno due fornitori**

```
SELECT DISTINCT C.pid
FROM Catalogo C
WHERE EXISTS (SELECT DISTINCT C1.pid
               FROM Catalogo C1
              WHERE C1.pid = C.pid AND
                    C1.fid <> C.fid)
```

L' i-esimo pezzo compare nel risultato dell'interrogazione interna se esiste un altro fornitore che vende quel pezzo.