

SQL

*operazioni sui dati:
interrogazioni con
join e prodotti cartesiano*

Selezione, proiezione e join

Le istruzioni SELECT :

- **con una sola relazione nella clausola FROM** permettono di realizzare:
selezioni, proiezioni, ridenominazioni
(con l'uso eventualmente di variabili)
- **con più relazioni nella FROM,** permettono di realizzare:
join e prodotti cartesiani

JOIN

Query : Trovare tutte le “coppie” di genitori

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Maternità

Madre	Figlio
Luisa	Maria
Maria	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Query : Trovare tutte le “coppie” di genitori

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Maternità

Madre	Figlio
Luisa	Maria
Maria	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Select padre, madre
From paternita, maternita
Where paternita.Figlio = maternita.Figlio

JOIN

NOTA : senza la condizione di selezione avremmo solo un prodotto cartesiano

Query : Trovare tutte le “coppie” di genitori

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Maternità

Madre	Figlio
Luisa	Maria
Maria	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Soluzione “piu’ corta” con variabili

```
Select      P.padre [as] papa, M.madre [as] mamma
From        paternita [as] P, maternita [as] M
Where       P.figlio = M.figlio
```

NOTA: La ridenominazione non è necessaria!

Query : Trovare tutte le “coppie” di genitori

Paternità

genitore	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Maternità

genitore	Figlio
Luisa	Maria
Maria	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Problema: ambiguità dell'attributo “genitore”

Soluzione: ridenominazione!

```
Select P.genitore [as] padre, M.genitore [as] madre
From paternita [as] P, maternita [as] M
Where P.figlio = M.figlio
```

JOIN

Query : Trovare i genitori di Olga

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Select

padre, madre

From

paternita P , maternita M

Where

P.figlio = M.figlio AND P.figlio = 'Olga'

condizione di join

condizione di selezione

SQL e prodotto cartesiano

Tabella1(att1, att2, ...) Tabella2(att3, att4, ...)



Select Tab1.att1, Tab2.att4

From Tab1, Tab2

Where Tab1.att2 = Tab2.att3




- selezione (**WHERE**)
- prodotto cartesiano (**FROM**)
- proiezione (**SELECT**)

possono essere necessarie ridenominazioni

- nel prodotto cartesiano
- nella target list

```
select X.Attrib1 as B1, ...  
from Tabella1 as X, Tabella2 as Y  
where X.Attrib2 = Y.Attrib3
```



JOIN: possono essere necessarie ridenominazioni

QUERY: Trovare tutte le “coppie” di genitori

Paternità	Nome	Figlio
	Sergio	Franco
	Luigi	Olga
	Luigi	Filippo
	Franco	Andrea
	Franco	Aldo

Maternità	Nome	Figlio
	Luisa	Maria
	Maria	Luigi
	Anna	Olga
	Anna	Filippo
	Maria	Andrea
	Maria	Aldo

Select **P.nome as padre, M.nome as madre**
From **paternita as P, maternita as M**
Where **P.figlio = M.figlio**

Problema: senza ridenominazione c'e' ambiguità!

SQL: esecuzione delle interrogazioni

- Le **espressioni SQL sono dichiarative** e noi ne stiamo vedendo la semantica
- In pratica, i DBMS eseguono le operazioni in modo **efficiente**, ad esempio:
 - eseguono le **selezioni** (la condizione “where”) **al più presto**
 - se possibile, **eseguono *join*** e non *prodotti cartesiani* (più lunghi!)

SQL: specifica delle interrogazioni

- La “capacità dei DBMS di ottimizzare” le interrogazioni (l’algoritmo di ricerca), non rende (di solito) necessario la “nostra efficienza” quando si specifica un’interrogazione
- È perciò più importante preoccuparsi della **chiarezza della specifica!** (anche perché così è più difficile sbagliare ...)
- ... ma non e’ detto sia sempre così !

Proiezione: attenzione!

Query: cognome e filiale di tutti gli impiegati

nome	cognome	filiale
Mario	Neri	Napoli
Alessia	Neri	Milano
Enzo	Rossi	Roma
Nadia	Rossi	Roma

Query 1

```
select cognome, filiale  
from impiegati
```

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma
Rossi	Roma

Attenzione! *Ci sono ripetizioni, questo vuol dire che in generale le risposte di una query non costituiscono relazioni valide (tabelle senza ripetizioni)*

Query 2:

```
select distinct cognome, filiale  
from impiegati
```

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma

Attenzione! *Non ci sono ripetizioni di righe (n-uple), quindi è una **tabella valida** (l'intestazione è una superchiave)*

Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

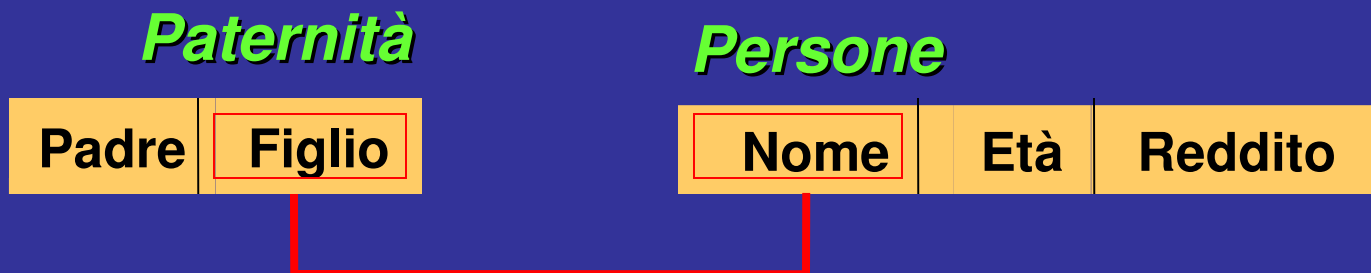
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Selezione, proiezione e join

Query : trovare i padri delle persone che guadagnano più di venti (mila euro)

```
select distinct padre  
from paternita, persone  
where figlio = nome  
and  
reddito > 20
```

NOTA: Un padre potrebbe avere più di un figlio che guadagna più di 20000E



Join naturale

Query : Padre e madre di ogni persona

paternita **NATURAL JOIN** maternita

select padre, madre, P.figlio
from maternita M, paternita P
where P.**figlio** = M.**figlio**

select padre, madre, paternita.figlio
from maternita **natural join** paternita

I nomi degli attributi di join devono essere uguali (...poco usato)

Join esplicito : „join...on...“

Query : trovare padre e madre di ogni persona

```
select madre, paternita.figlio, padre
from maternita, paternita
where paternita.figlio = maternita.figlio
```

... alternativa ...

```
select madre, paternita.figlio, padre
from maternita join paternita on
paternita.figlio = maternita.figlio
```

Nota: omettiamo la condizione di selezione nella clausola WHERE
In tal modo le queries sono più compatte e veloce nell'esecuzione

SELECT con *join esplicito*: sintassi

```
SELECT      { ... attributi ... }  
FROM        Tabella1 JOIN Tabella2 JOIN Tabella3 ...  
           ON   CondizioneDiJoin1  
           ON   CondizioneDiJoin2  
           ...  
[ WHERE     AltraCondizione di selezione ]
```

Join esterno: "outer join"

Query : *trova il padre e, se nota, la madre di ogni persona*

```
select paternita.figlio, padre, madre  
from    paternita left join maternita  
         on paternita.figlio = maternita.figlio
```

```
select paternita.figlio, padre, madre  
from    paternita left [outer] join maternita  
         on paternita.figlio = maternita.figlio
```

NOTA: il termine "outer" e' opzionale

Outer join

```
select paternita.figlio, padre, madre  
from maternita join paternita  
  on maternita.figlio = paternita.figlio
```

```
select paternita.figlio, padre, madre  
from maternita left/right [outer] join paternita  
  on maternita.figlio = paternita.figlio
```

```
select paternita.figlio, padre, madre  
from maternita full [outer] join paternita  
  on maternita.figlio = paternita.figlio
```

Uso di istanze diverse della stessa tabella

QUERY - Trovare le persone che sono nonni
con i relativi nipoti

Paternalità (Nonni)



Paternalità (Padri)



Valori uguali

join

Soluzione 1: con prodotto cartesiano e condizione di selezione

```
Select N.padre as nonno,  
       P.figlio as nipote  
From   paternita as N,  
       paternita as P  
Where  N.figlio = P.padre
```

Soluzione 2: con join esplicito

```
Select N.padre as nonno,  
       P.figlio as nipote  
From   paternita as N join  
       paternita as P  
on     N.figlio = P.padre
```