

# **Basi di Dati**

## **Il linguaggio SQL (seconda parte)**

**Marco Maratea**  
**Laurea in Informatica, DeMaCS, UNICAL**

**27 Novembre 2025**



## Selezione, proiezione e join

- Istruzioni SELECT con una sola relazione nella clausola FROM permettono di realizzare:
  - selezioni, proiezioni, ridenominazioni
  - con più relazioni nella FROM si realizzano join (e prodotti cartesiani)



Paolo Atzeni  
Stefano Ceri  
Piero Fraternali  
Stefano Paraboschi  
Riccardo Torlone

## SQL e algebra relazionale

- $R1(A1,A2) \ R2(A3,A4)$

```
select distinct R1.A1, R2.A4  
from R1, R2  
where R1.A2 = R2.A3
```

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



## SQL e algebra relazionale, 2

- $R1(A1, A2) \ R2(A3, A4)$

```
select R1.A1, R2.A4  
from R1, R2  
where R1.A2 = R2.A3
```

$$\text{PROJ}_{A1, A4} (\text{SEL}_{A2=A3} (R1 \text{ JOIN } R2))$$



- possono essere necessarie ridenominazioni
  - nel prodotto cartesiano
  - nella target list

```
select X.A1 AS B1, ...
from   R1 X, R2 Y, R1 Z
where  X.A2 = Y.A3 AND ...
```



```
select X.A1 AS B1, Y.A4 AS B2
from R1 X, R2 Y, R1 Z
where X.A2 = Y.A3 AND Y.A4 = Z.A1
```

```
REN B1,B2←A1,A4 (
PROJ A1,A4 (SEL A2 = A3 AND A4 = C1(
R1 JOIN R2 JOIN REN C1,C2 ← A1,A2 (R1))))
```



Paolo Atzeni  
Stefano Ceri  
Piero Fraternali  
Stefano Paraboschi  
Riccardo Torlone

Basi di dati

VI edizione

McGraw-Hill Connect

McGraw-Hill

## 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 al più presto
  - se possibile, eseguono join e non prodotti cartesiani



## SQL: specifica delle interrogazioni

- La capacità dei DBMS di "ottimizzare" le interrogazioni, rende (di solito) non necessario preoccuparsi dell'efficienza quando si specifica un'interrogazione
- È perciò più importante preoccuparsi della chiarezza



| 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

- I padri di persone che guadagnano più di 20

$$\begin{array}{c} \text{PROJ}_{\text{Padre}}(\text{paternita} \\ \text{JOIN } \text{Figlio} = \text{Nome} \\ \text{SEL}_{\text{Reddito} > 20} (\text{persone})) \end{array}$$

```
select distinct Padre
from Persone, Paternita
where Figlio = Nome and Reddito > 20
```

, è prodotto cartesiano



- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

PROJ<sub>Nome, Reddito, RP</sub> (SEL<sub>Reddito > RP</sub>  
(REN<sub>NP,EP,RP ← Nome,Eta,Reddito</sub> (persone)  
JOIN<sub>NP=Padre</sub>  
(paternita JOIN<sub>Figlio =Nome</sub> persone)))

```
select f.Nome, f.Reddito, p.Reddito
from Persone p, Paternita, Persone f
where p.Nome = Padre and
      Figlio = f.Nome and
      f.Reddito > p.Reddito
```



# SELECT, con ridenominazione del risultato

```
select Figlio, f.Reddito as Reddito,  
       p.Reddito as RedditoPadre  
  from Persone p, Paternita, Persone f  
 where p.Nome = Padre and Figlio = f.Nome  
       and f.Reddito > p.Reddito
```



## Join esplicito

- Padre e madre di ogni persona

```
select Paternita.Figlio, Padre, Madre  
from Maternita, Paternita  
where Paternita.Figlio = Maternita.Figlio
```

```
select Madre, Paternita.Figlio, Padre  
from Maternita join Paternita on  
Paternita.Figlio = Maternita.Figlio
```



SELECT ...

FROM Tabella { ... JOIN Tabella ON CondDiJoin }, ...  
[ WHERE AltraCondizione ]



## Join vs prodotto cartesiano

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

```
select f.Nome, f.Reddito, p.Reddito
from (Persone p join Paternita on p.Nome = Padre)
      join Persone f on Figlio = f.Nome
where f.Reddito > p.Reddito
```

```
select f.Nome, f.Reddito, p.Reddito
from Persone p, Paternita, Persone f
where p.Nome = Padre and
      Figlio = f.Nome and
      f.Reddito > p.Reddito
```



## Ulteriore estensione: join naturale (meno diffuso)

$\text{PROJ}_{\text{Figlio}, \text{Padre}, \text{Madre}}($   
 paternita JOIN <sub>$\text{Figlio} = \text{Nome}$</sub>  REN <sub>$\text{Nome}=\text{Figlio}$</sub> (maternita))  
  
 paternita JOIN maternita

```
select Madre, Paternita.Figlio, Padre
from Maternita join Paternita on
Paternita.Figlio = Maternita.Figlio
```

```
select Madre, Figlio, Padre
from Maternita natural join Paternita
```

mimer OK  
DB2 no



## Join esterno: "outer join"

- Padre e, se nota, 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
```

- **outer** è 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 outer join Paternita  
on Maternita.Figlio = Paternita.Figlio
```

```
select Paternita.Figlio, Padre, Madre  
from Maternita full outer join Paternita  
on Maternita.Figlio = Paternita.Figlio
```



Paolo Atzeni  
Stefano Ceri  
Piero Fraternali  
Stefano Paraboschi  
Riccardo Torlone

Basi di dati

VI edizione

McGraw-Hill Connect

McGraw-Hill

## Ordinamento del risultato

- Nome e reddito delle persone con meno di trenta anni in ordine alfabetico

```
select Nome, Reddito  
from Persone  
where Eta < 30  
order by Nome
```



```
select Nome, Reddito  
from Persone  
where Eta < 30
```

Persone

| Nome    | Reddito |
|---------|---------|
| Andrea  | 21      |
| Aldo    | 15      |
| Filippo | 30      |

```
select Nome, Reddito  
from Persone  
where Eta < 30  
order by Nome
```

Persone

| Nome    | Reddito |
|---------|---------|
| Aldo    | 15      |
| Andrea  | 21      |
| Filippo | 30      |



## Unione, intersezione e differenza

- La `select` da sola non permette di fare unioni; serve un costrutto esplicito:

```
select ...
union [all]
select ...
```

- i duplicati vengono eliminati (a meno che si usi `all`); anche dalle proiezioni!



```
select A, B  
from R  
union  
select A , B  
from S
```

```
select A, B  
from R  
union all  
select A , B  
from S
```



Paolo Atzeni  
Stefano Ceri  
Piero Fraternali  
Stefano Paraboschi  
Riccardo Torlone

## Notazione posizionale!

```
select padre, figlio  
from paternita  
union  
select madre, figlio  
from maternita
```

- quali nomi per gli attributi del risultato?
    - nessuno
    - quelli del primo operando
    - ...
- (dipende da BD)



| Figlio |         |
|--------|---------|
| Sergio | Franco  |
| Luigi  | Olga    |
| Luigi  | Filippo |
| Franco | Andrea  |
| Franco | Aldo    |
| Luisa  | Maria   |
| Luisa  | Luigi   |
| Anna   | Olga    |
| Anna   | Filippo |
| Maria  | Andrea  |
| Maria  | Aldo    |

| Padre  | Figlio  |
|--------|---------|
| Sergio | Franco  |
| Luigi  | Olga    |
| Luigi  | Filippo |
| Franco | Andrea  |
| Franco | Aldo    |
| Luisa  | Maria   |
| Luisa  | Luigi   |
| Anna   | Olga    |
| Anna   | Filippo |
| Maria  | Andrea  |
| Maria  | Aldo    |



## Notazione posizionale, 2

```
select padre, figlio  
from paternita  
union  
select figlio, madre  
from maternita
```

```
select padre, figlio  
from paternita  
union  
select madre, figlio  
from maternita
```



## Notazione posizionale, 3

- Anche con le ridenominazioni non cambia niente:

```
select padre as genitore, figlio
from paternita
union
select figlio, madre as genitore
from maternita
```

- Corretta:

```
select padre as genitore, figlio
from paternita
union
select madre as genitore, figlio
from maternita
```



# Differenza

Basi di dati

VI edizione

McGraw-Hill connect

McGraw-Hill

```
select Nome  
from Impiegato  
except  
select Cognome as Nome  
from Impiegato
```

(solo DB2)

- vedremo che si può esprimere con **select** nidificate



# Intersezione

```
select Nome  
from Impiegato  
intersect  
select Cognome as Nome  
from Impiegato
```

(solo DB2)

- equivale a

```
select I.Nome  
from Impiegato I, Impiegato J  
where I.Nome = J.Cognome
```