

# SQL

(quarta parte)

# Argomenti della lezione

- ▶ Interrogazioni complesse in SQL
  - ▶ join esplicito
  - ▶ outer join
  - ▶ operatori aggregati
  - ▶ interrogazioni nidificate

**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      |

# Join esplicito

- Padre e madre di ogni persona

paternita JOIN maternita

$\text{PROJ}_{\text{Figlio, Padre, Madre}}(\text{paternita}$   
 $\text{JOIN}_{\text{Figlio} = \text{Nome}}$   
 $\text{REN}_{\text{Nome} \leftarrow \text{Figlio}}(\text{maternita}))$

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

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

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

# Outer join

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

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

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

# 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**

| <b>Nome</b> | <b>Reddito</b> |
|-------------|----------------|
| Andrea      | 21             |
| Aldo        | 15             |
| Filippo     | 30             |

**select nome,  
                    reddito  
from persone  
where eta < 30  
order by nome**

**Persone**

| <b>Nome</b> | <b>Reddito</b> |
|-------------|----------------|
| Aldo        | 15             |
| Andrea      | 21             |
| Filippo     | 30             |



# Operatori aggregati

- Il numero di figli di Franco

```
select count(*)  
  AS NumFigliDiFranco  
from Paternita  
where Padre = 'Franco'
```

**Paternità**

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

**NumFigliDiFranco**

2

- ▶ Il numero di figli di ciascun padre

```
select Padre, count(*) AS NumFigli  
from Paternita  
group by Padre
```

# **Paternità**

| <b>Padre</b> | <b>Figlio</b> |
|--------------|---------------|
| Sergio       | Franco        |
| Luigi        | Olga          |
| Luigi        | Filippo       |
| Franco       | Andrea        |
| Franco       | Aldo          |

| <b>Padre</b> | <b>NumFigli</b> |
|--------------|-----------------|
| Sergio       | 1               |
| Luigi        | 2               |
| Franco       | 2               |

# Altri operatori aggregati

- ▶ Somma: **SUM**
- ▶ Media: **AVG**
- ▶ Massimo: **MAX**
- ▶ Minimo: **MIN**

# Unione

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

**SELECT ...**

**UNION**

**SELECT ...**

# Notazione posizionale!

```
select padre from paternita  
union
```

```
select madre from maternita
```

- ▶ quali nomi per gli attributi del risultato?

|        | 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

**select padre, figlio  
from paternita  
union**

**select figlio, madre  
from maternita**

**select padre, figlio  
from paternita  
union**

**select madre, figlio  
from maternita**

**select padre as genitore, figlio  
from paternita  
union**

**select madre as genitore, figlio  
from maternita**

# Interrogazioni nidificate

le condizioni atomiche permettono anche:

- ▶ il confronto fra un attributo e il risultato di una sottointerrogazione
- ▶ quantificazioni esistenziali

- ▶ nome e reddito del padre di Mario

```
select Nome, Reddito  
from Persone, Paternita  
where Nome = Padre and  
Figlio = 'Mario'
```

- ▶ nome e reddito del padre di Mario

```
select Nome, Reddito  
from Persone  
where Nome = (select Padre  
               from Paternita  
               where Figlio = 'Mario')
```

- ▶ Nome e reddito dei padri di persone che guadagnano più di 20 milioni

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



- Le persone che hanno almeno un figlio

```
select *  
from Persone  
where exists ( select *  
               from Paternita  
               where Padre = Nome) or  
exists ( select *  
         from Maternita  
         where Madre = Nome)
```



# Quantificazione esistenziale

- ▶ possiamo avere condizioni del tipo:
  - ▶ **EXISTS** ( Sottoespressione )

- I padri i cui figli guadagnano tutti più di venti milioni

```
select distinct Padre
from Paternita Z
where not exists (
    select *
    from Paternita W, Persone
    where W.Padre = Z.Padre
    and W.Figlio = Nome
    and Reddito <= 20)
```

# Sintesi

- ▶ Interrogazioni complesse in SQL
  - ▶ join esplicito
  - ▶ outer join
  - ▶ operatori aggregati
  - ▶ interrogazioni nidificate