

Operazioni su insiemi: unione, intersezione e differenza

Unione

- 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!

Unione: notazione posizionale 1

```
Select padre  
From paternita  
union  
select madre  
from maternita
```

- quali nomi per gli attributi del risultato?
 - nessuno
 - quelli del primo operando
 - Altro...

	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

Unione: notazione posizionale, 2

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

Scorretta!

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

Corretta!

Unione: notazione posizionale, 3

- Anche con le ridenominazioni non cambia niente,

scorretta:

```
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

I nomi di impiegati che non sono anche cognomi

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

(vedremo che si può esprimere con **select** nidificate)

Intersezione

I nomi di impiegati che sono anche cognomi

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

In alternativa, si può fare usando il join:

```
select I.Nome
from Impiegato as I JOIN Impiegato as J
on I.Nome = J.Cognome
```


Operazioni sui dati

Interrogazioni con **operatori aggregati**

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
```

```
select nome, reddito  
from persone  
where eta < 30  
order by nome  
(order by reddito)
```

Persone

Nome	Reddito
Andrea	21
Aldo	15
Filippo	30

Persone

Nome	Reddito
Aldo	15
Andrea	21
Filippo	30

Operatori aggregati

- Nelle espressioni della target list possiamo avere anche **espressioni che calcolano valori a partire da insiemi di ennuple**:
 - conteggio, minimo, massimo, media, totale
 - sintassi base (semplificata):

Funzione ([DISTINCT] *)

Funzione ([DISTINCT] Attributo)

Operatori aggregati: COUNT

- Il numero di figli di Franco

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

- l'operatore aggregato (**count**) viene applicato al risultato dell'interrogazione:

```
select *  
from Paternita  
where Padre = 'Franco'
```

Paternità

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

NumFigliDiFranco

2

COUNT e valori nulli

I valori nulli non vengono contati

```
select count(*) from persone
```

```
select count(reddito) from persone
```

```
select count(distinct reddito) from persone
```

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	NULL
Maria	55	21
Anna	50	35

Altri operatori aggregati

- **SUM, AVG, MAX, MIN**
- **Media dei redditi dei figli di Franco**

```
select  avg(reddito)
from    persone join paternita on nome=figlio
where   padre='Franco'
```


Operatori aggregati e valori nulli

```
select avg(reddito) as redditomedio  
from persone
```

viene calcolato $(30+36+36)/3$; non viene considerato il valore NULL

Persone

Nome	Età	Reddito
Andrea	27	30
Aldo	25	NULL
Maria	55	36
Anna	50	36

Operatori aggregati e target list

- un'interrogazione scorretta:

```
Select nome, max(reddito)  
from persone
```

scorretta !

di chi sarebbe il nome? (potrebbero esserci più persone con stesso nome e reddito massimo!)

- La target list deve essere omogenea:

```
select min(eta), avg(reddito)  
from persone
```

corretta !

Operatori aggregati e raggruppamenti

- Gli operatori aggregati possono essere applicati a **partizioni delle relazioni**
- La clausola **GROUP BY**, consente di partizionare le ennuple di una relazione; la sintassi:

GROUP BY listaAttributi

Operatori aggregati e raggruppamenti

- Il numero di figli di ciascun padre (con ordinamento)

```

Select  padre, count(distinct figlio) as NumeroFigli
from    paternita
group by padre
(order by padre)
  
```

paternita

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

Padre	NumFigli
Sergio	1
Luigi	2
Franco	2

Semantica delle interrogazioni con operatori aggregati e raggruppamenti

1. interrogazione senza **group by** e senza operatori aggregati
select *
from paternita
2. si raggruppa (o partiziona la relazione) e poi si applica l'operatore aggregato a ciascun gruppo

Raggruppamenti e target list

Confronta il reddito di ciascun padre con quello medio dei propri figli

Query corretta:

```
select    padre, avg(f.reddito), p.reddito
from      (persone f join paternita on figlio = f.nome)
          join persone p on padre =p.nome
group    by padre
```

oppure:

```
select    padre, avg(f.reddito), p.reddito
from      (persone f join paternita on figlio = f.nome
          join persone p on padre =p.nome)
group    by padre, p.reddito
```

Condizioni (having) sui gruppi

Seleziona i padri i cui figli hanno un reddito medio maggiore di 25

Query Corretta:

```
Select      padre, avg(f.reddito)
From        persone f join paternita on figlio = f.nome
group by    padre
having      avg(f.reddito) > 25
```

Query Scorretta:

```
Select      padre, avg(f.reddito)
From        persone f join paternita on figlio = f.nome
group by    padre
where       avg(f.reddito) > 25
```

WHERE o HAVING?

I padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 20

Query corretta:

```
Select      padre, avg(f.reddito)
from        persone f join paternita on figlio = f.nome
where       eta < 30
group       by padre
having      avg(f.reddito) > 25
```

Query corretta:

```
Select      padre, avg(f.reddito)
from        persone f join paternita on figlio = f.nome
group       by padre
having      avg(f.reddito) > 25 and eta < 30
```

NOTA: “having” è una condizione di selezione su gruppi, si usa quindi con attributi, della target list, su cui agiscono operatori aggregati

Sintassi, riassumiamo

Struttura generale Query SQL :

Select ListaAttributi_o_Espressioni
from ListaTabelle
[where CondizioniSemplici]
[group by ListaAttributi_di_Raggruppamento]
[having CondizioniAggregate]
[order by ListaAttributi_di_Ordinamento]