

Operatori aggregati e target list

- un' interrogazione scorretta:

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

- di chi sarebbe il nome? La target list deve essere omogenea

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

Proviamo ...

- Si può fare
 - con i costrutti di SQL che conosciamo
 - con l'aiuto di una vista (concetto che non abbiamo ancora discusso – lo facciamo subito)
 - oppure con le "interrogazioni nidificate"

Proviamo ...

- Si può fare
 - **con i costrutti di SQL che conosciamo**
 - **con l'aiuto di una vista (concetto che non abbiamo ancora discusso – lo facciamo subito)**
 - oppure con le "interrogazioni nidificate"

Una soluzione

- "Le persone con reddito superiore alla media"
 - troviamo il reddito medio (vista)
 - Una tabella con un attributo (la media) ed una ennupla (il valore della media)
 - confrontiamo ciascun reddito con il reddito medio (prodotto cartesiano + selezione)

Viste

```
CREATE VIEW V AS  
SELECT ...
```

anche (non in tutti i sistemi)

```
CREATE VIEW V AS  
SELECT ...  
UNION  
SELECT ...
```

```
CREATE OR REPLACE VIEW V AS  
SELECT ...
```

I vari passi (persone con reddito superiore alla media)

```
create view mediaReddito  
as select avg(reddito) as redditoMedio  
from persone;
```

Mostrare su [sqliteonline](#)

```
select *  
from persone, mediaReddito;
```

Mostrare su [sqliteonline](#)

```
select nome, reddito, redditomedio  
from persone, mediaReddito  
where reddito > redditoMedio;
```

Proviamo ...

- Si può fare
 - con i costrutti di SQL che conosciamo
 - con l'aiuto di una vista (concetto che non abbiamo ancora discusso – lo facciamo subito)
 - **oppure con le "interrogazioni nidificate"**

Interrogazioni nidificate (nested query o subquery)

- Varie forme di nidificazione
 - nella WHERE
 - nella FROM
 - nella SELECT
- Coerente con i tipi
 - anche Booleano (EXISTS)
 - non lo vediamo quest'anno
 - attenzione al rapporto fra singoli valori e insiemi

Nella WHERE

- La persona che guadagna più di tutte le altre

```
select *  
from persone  
where reddito = ( select max(reddito)  
                  from persone)
```

Nella WHERE

- Le persone che guadagnano più della media

```
select *  
from persone  
where reddito >= ( select avg(reddito)  
                  from persone)
```

Correlated subquery

- Per ogni padre, il figlio che guadagna di più

```
select padre, figlio, reddito
from persone join paternita p on nome = figlio
where reddito = ( select max(reddito)
                  from persone join paternita
                        on nome = figlio
                  where padre = p.padre )
```

- L'interrogazione interna viene eseguita una volta per ciascuna ennupla della FROM esterna

Correlated subquery

- Per ogni padre, il figlio che guadagna di più

```
select padre, figlio, reddito
from persone join paternita p on nome = figlio
where reddito = ( select max(reddito)
                  from persone join paternita
                        on nome = figlio
                  where p.padre = padre )
```

**Mostrare su
sqliteonline**

- L'interrogazione interna viene eseguita una volta per ciascuna ennupla della FROM esterna

Per semplificare, usiamo una vista

```
CREATE VIEW PersoneConPadre  
AS      select *  
        from persone join paternita  
              on nome = figlio
```

Correlated subquery

- Per ogni padre, il figlio che guadagna di più

```
select *  
from personeconPadre p  
where reddito = (select max(reddito)  
                 from personeconPadre  
                 where p.padre = padre )
```

- L'interrogazione interna viene eseguita una volta per ciascuna ennupla della FROM esterna
- Per ogni ennupla **p** di **personeconPadre** viene eseguita

```
select max(reddito)  
from personeconPadre  
where p.padre = padre
```

dove **p.padre** è una costante e il valore ottenuto (chiamiamolo **m**) viene utilizzato nell'interrogazione "esterna"

```
select *  
from personeconPadre p  
where reddito = m
```

Altre nidificazioni nella WHERE

- La motivazione originaria per la nidificazione

- nome e reddito del padre di Franco

```
select Nome, Reddito  
from Persone join Paternita on Nome = Padre  
where Figlio = 'Franco'
```

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

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

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

```
select Nome, Reddito  
from Persone  
where Nome in (select Padre  
                from Paternita  
                where Figlio = any (select Nome  
                                     from Persone  
                                     where Reddito > 20))
```

notare la **distinct**

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

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

```
select Nome, Reddito  
from Persone  
where Nome in (select Padre  
from Paternita, Persone  
where Figlio = Nome  
and Reddito > 20)
```

Altre forme di nidificazione

- Le vediamo più per curiosità che per altro

Nidificazione nella FROM

- Per ogni padre, il figlio che guadagna di più

```
select p.*  
from personeconPadre p join  
    ( select padre, max(reddito) as maxreddito  
      from personeconPadre  
      group by padre  
    ) as m on p.padre = m.padre  
where reddito = maxreddito
```

Ancora nella FROM

- Tutte le persone, con il reddito massimo dei figli dello stesso padre

```
select p.*, maxreddito
from personeconPadre p ,
    ( select padre, max(reddito) as maxreddito
      from personeconPadre
     group by padre
    ) as m
where p.padre = m.padre
```

Nidificazione nella SELECT

- Calcolo di valori con la nidificazione
- Per ogni padre, tutti i dati e il reddito massimo dei suoi figli (correlazione)

```
select distinct padre, (select max(reddito)
                        from paternita join persone
                        on figlio = nome
                        where padre = p.padre)
from paternita p join persone on padre =nome
```

Operazioni di aggiornamento

- Vediamo solo aspetti base, per sapere operare, i dettagli non interessano e ci sono ben pochi concetti importanti

```
INSERT INTO Persone VALUES ('Mario',25,52)
```

```
INSERT INTO Persone(Nome, Reddito, Eta)  
VALUES('Pino',52,23)
```

```
INSERT INTO Persone(Nome, Reddito)  
VALUES('Lino',55)
```

```
INSERT INTO Persone ( Nome )  
SELECT Padre  
FROM Paternita  
WHERE Padre NOT IN (SELECT Nome  
FROM Persone)
```

Eliminazione di ennuple

DELETE FROM Tabella
[WHERE Condizione]

```
DELETE FROM Persone  
WHERE Eta < 35
```

```
DELETE FROM Paternita  
WHERE Figlio NOT in (
```

```
SELECT Nome  
FROM Persone)
```

```
DELETE FROM Paternita
```

Modifica di ennuple

```
UPDATE NomeTabella  
SET Attributo = < Espressione |  
                SELECT ... |  
                NULL |  
                DEFAULT >  
[ WHERE Condizione ]
```

```
UPDATE Persone SET Reddito = 45  
WHERE Nome = 'Piero'
```

```
UPDATE Persone  
SET Reddito = Reddito * 1.1  
WHERE Eta < 30
```

Altre operazioni DDL

- Aggiungere vincoli, con verifica (provate)

```
alter table persone  
  add constraint redditononnegativo  
    check (reddito >=0)
```

```
alter table persone  
  add constraint redditoesagerato  
    check (reddito >=100)
```

Esercizi con supporto alla correzione

- [prova parziale del 18/11/2019](#), compito C, esercizi 2, 3, 4, 5
 - [script SQL](#) per creare e popolare la base di dati per questi esercizi
 - [esercitazione su Moodle con correzione automatica](#)

Note su sqLiteonline e sulle soluzioni

```
select cod, nome  
from Partiti join Risultati on cod = partito join elezioni on elezione = id  
where tipoelezione = 'Europee'  
group by partito
```

```
SELECT ID, Anno, TipoElezione, count(Cod) as NumeroPartiti  
FROM Risultati JOIN Partiti ON (Partito = Cod) JOIN Elezioni ON Elezione = ID  
GROUP BY ID
```