

Interrogazioni nidificate (nested query o subquery)

- Varie forme di nidificazione
 - nella WHERE
 - nella FROM
 - nella SELECT
- Coerente con i tipi
 - anche Booleano (EXISTS)
 - 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 p.padre = padre )
```

- 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
```

in cui p.padre è una costante e il valore m risultante viene utilizzato in

```
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`

- In questo caso la nidificazione non aiuta molto

- 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)
```

- Nome e reddito dei padri di persone che guadagnano più di 20, **con indicazione del reddito del figlio**

```
select distinct P.Nome, P.Reddito, F.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))
```

EXISTS

- Quantificatore esistenziale
- Correlazione fra la sottointerrogazione e le variabili nel resto

- 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)
```

Disgiunzione (OR) e unione

```
select distinct Persone.*  
from Persone join Paternita on padre= nome  
union  
select distinct Persone.*  
from Persone join Maternita on madre= nome
```

- I padri i cui figli guadagnano **tutti** più di 20

```
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)
```

- I padri i cui figli guadagnano **tutti** più di 20

```
select distinct padre
```

```
from paternita
```

```
except
```

```
select distinct padre
```

```
from paternita join persone on figlio = nome
```

```
where reddito <= 20
```

- I padri i cui figli guadagnano tutti più di 20

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

NO!!! provare ad eseguire per vedere la differenza

- I padri i cui figli guadagnano tutti più di 20

```
select distinct Padre
from Paternita
where not exists (
    select *
    from Persone
    where Reddito <= 20)
```

NO!!! provare anche questa

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

```
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](#)