

Universita' di Milano Bicocca  
Corso di Basi di dati 1 in eLearning

C. Batini

7. SQL DML

7.11 Interrogazioni nidificate

Regole di visibilita'

# Regole di visibilita' nelle espressioni nidificate

# Interrogazioni nidificate, regole di visibilità - 1

1. non è possibile fare riferimenti da blocchi esterni a variabili (cioe' nomi definiti con la clausola as) definite in blocchi più interni  
→ perche' si assume che se e' stata dichiarata in una SELECT interna sia utilizzabile solo lì

```
select Nome.P1, Reddito.P1, Reddito.P2  
from Persone P1  
where Nome.P2 in (select Padre  
                  from Paternita  
                  where Figlio = any (select Nome  
                                     from Persone P2  
                                     where Reddito > 20))
```

# Altro esempio di visibilità scorretta

```
select *  
from Impiegato  
where Dipart in (select Nome  
                 from Dipartimento D1  
                 where Nome = 'Produzione')  
  
or  
Dipart in (select Nome  
          from Dipartimento D2  
          where D2.Citta = D1.Citta)
```

D1 definita nel blocco sopra

## Esercizio 7.11.1

Nell'esempio precedente chi ha scritto la interrogazione voleva probabilmente trovare gli impiegati del Dipartimento Produzione e dei Dipartimenti localizzati nella stessa città' del Dipartimento Produzione.

Scrivi una interrogazione SQL che ha lo stesso significato, ma rispetta le regole di visibilita'

## Interrogazioni nidificate, regole di visibilità - 2

2. in un blocco si può invece fare riferimento a  
variabili definite in blocchi più esterni

```
select Nome.P1, Reddito.P1, Reddito.P2
from Persone P1, Persone P2
where Nome.P1 in (select Padre
                  from Paternita
                  where Figlio = any (select Nome
                                     from P2
                                     where Reddito > 20))
```

## Interrogazioni nidificate, regole di visibilità - 3

3. se un nome di variabile è omesso, si assume come riferimento la variabile più "vicina" dichiarata all'esterno.

## Interrogazioni nidificate, regole di visibilità - 4 Attenzione!

- A seguito della regola 1, che afferma non è possibile fare riferimento a variabili definite in blocchi più interni

la semantica "bottom up" che recita

- esegui prima l'interrogazione interna e poi nella esecuzione della where esterna utilizza per tutte le n-ple il risultato intermedio

- non puo' essere piu' applicata. Infatti →

- Ricordiamo la semantica "bottom up"  
Esempio: Data la Relazione Persona (Codice Fiscale, Nome, Cognome) trovare le persone che hanno degli omonimi (cioe' persone con stesso nome e cognome ma diverso codice fiscale)

Come faccio a eseguire la interrogazione interna senza assegnare un valore alla variabile P?

~~Select \*  
from Persona P  
where exists (select \*  
from Persona P1  
where P1. Nome = P.Nome and  
P1. Cognome = P.Cognome and  
P1. CodFiscale not = P.CodFiscale)~~

## Interrogazioni nidificate, regole di visibilità - 4 Attenzione!

- Quindi non e' possibile "eseguire" la interrogazione dai blocchi interni verso i blocchi esterni

## Vale ancora e solo la semantica top-down

- Quando in un blocco interno si riferimento a variabili definite in blocchi più esterni, l'interrogazione nidificata viene valutata separatamente per ogni n-pla prodotta nella valutazione della query esterna.
- Quindi per ogni n-pla di una query esterna si esegue il blocco piu' interno e poi si confronta. → Semantica top down

# Esempio di applicazione della regola 4 nella costruzione della interrogazione

- I padri i cui figli guadagnano tutti più di venti mila €

## Maternità

Madre	<u>Figlio</u>
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Legami  
logici

## Paternità

<u>Padre</u>	<u>Figlio</u>
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

## Persone

<u>Nome</u>	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

## Esempio di applicazione della regola 4

- I padri i cui figli guadagnano tutti più di venti mila €

```
select distinct Padre  
from Paternita Z  
where not exists (figli_che_guadagnano_  
                    meno_di_20mila)
```

Per ogni padre, cerchiamo con una query interna se non esistono, per quel padre, figli che guadagnano meno di 20 mila. Se non esistono lo selezioniamo → Clausola not exist

•I padri i cui figli guadagnano tutti più di venti mila

```
select distinct Padre  
from Paternita  
where not exists (  
    figli_che_guadagnano_  
    meno_di_20mila  
)
```

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

```
select distinct Padre  
from Paternita  
where not exists (
```

```
)
```

•I padri i cui figli guadagnano tutti più di venti mila

```
select distinct Padre
from Paternita
where not exists (
    select *
    from Persone
)
```

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

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

•I padri i cui figli guadagnano tutti più di venti mila

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

## Meno compatta ma ugualmente corretta



```
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)
```

Cerchiamo i Figli:

1. Accoppiando la copia Z di Paternita' con la copia "interna" di Paternita' e poi

2 accoppiando tale relazione

con Persone, per trovare i padri con figli con reddito <=20

Altro esempio: nella relazione Persona (CodFiscale, Nome,  Cognome), trovare le persone che non hanno omonimi

Prima la Select \*

```
from Persona P
where exists (select *
              from Persona P1
              where P1. Nome = P.Nome and
                    P1. Cognome = P.Cognome and
                    P1. Codfiscale not = P.Cod.Fiscale)
```

Trovava le persone che hanno omonimi.

Ora?

```
Select *
from Persona P
where not exists (select *
                  from Persona P1
                  where P1. Nome = P.Nome and
                        P1. Cognome = P.Cognome and
                        P1. Codfiscale not = P.Cod.Fiscale)
```



## Disgiunzione e unione

```
select * from Persone where Reddito > 30
union
select F.*
from Persone F, Paternita, Persone P
where F.Nome = Figlio and Padre = P.Nome
and P.Reddito > 30
```



### Risultato

Le persone che hanno un reddito > di 30  
insieme alle persone i cui padri hanno un reddito >  
di 30



## Disgiunzione e unione

```
select * from Persone where Reddito > 30  
union
```

```
select F.*
```

```
from Persone F, Paternita, Persone P
```

```
where F.Nome = Figlio and Padre = P.Nome  
and P.Reddito > 30
```

- Formulazione equivalente con exists e query nidificata

```
select *
```

```
from Persone F
```

```
where Reddito > 30 or
```

```
exists (select *
```

```
from Paternita, Persone P
```

```
where F.Nome = Figlio and Padre = P.Nome  
and P.Reddito > 30)
```

## Approfondimento 7.11.1

- Leggi e studia la sezione 4.8 del libro di riferimento, dove sono proposte alcune interrogazioni in SQL e Algebra relazionale (ignora le parti degli esercizi sul calcolo relazionale e sul Prolog, che non fanno parte del programma)

## Esercizio 7.11.2

- Esegui l'esercizio 4.7 del testo di riferimento, domande 10-12.

## Esercizio 7.11.3

- Esegui l'esercizio 4.8 del testo di riferimento.

# Concetti introdotti

- Regole di visibilita' per le interrogazioni nidificate.