

Corso di Laurea in Ingegneria
Informatica
Fondamenti di Informatica II
Modulo “*Basi di dati*”
a.a. 2014-2015

Docente: Gigliola Vaglini
Docente laboratorio: Francesco
Pistolesi

Lezione 4

Interrogazioni annidate

Interrogazioni annidate

- le condizioni atomiche permettono anche
 - il confronto fra un attributo (o più) e il risultato di una sottointerrogazione
 - quantificazioni esistenziali

- nome e reddito del padre di Franco

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

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

Interrogazioni annidate

- La forma annidata è “meno dichiarativa”, ma richiede meno variabili
- La forma piatta e quella annidata possono essere combinate
- Le sottointerrogazioni non possono contenere operatori insiemistici; la limitazione non è significativa

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

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

E 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
```

se un padre ha più di un figlio...

- 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))
```

```

select distinct p.Nome, p.Reddito, f.Reddito
from Persone p, Persone f
where p.Nome in (select Padre
                  from Paternita
                  where Figlio = f.Nome
                  and f.Reddito > 20))

```

Quantificazione esistenziale

- Ulteriore tipo di condizione
 - EXISTS (Sottoespressione)

Esecuzione

- L'interrogazione interna viene eseguita una volta per ciascuna tupla dell'interrogazione esterna

- 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)

```

– 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)
```

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)
```

E questo?

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

Interrogazioni annidate, 3

- Regole di visibilità generali
 - non è possibile fare riferimenti a variabili definite in blocchi più interni
 - se un nome di variabile è omesso, si assume un riferimento alla variabile più “vicina”
- in un blocco si può fare riferimento a variabili definite in blocchi più esterni

Disgiunzione e unione (non sempre)

Elencare le persone con reddito >30 oppure che hanno un padre con reddito >60

```
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 > 60

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)
```

Si consideri il seguente schema di base di dati

- Film(CodiceFilm, Titolo, CodiceRegista, Anno)
- Produzione (CasaProduzione, Nazionalità, CodiceFilm, Costo, Incasso1annoSala)
- Artista (CodiceAttore, Cognome, Nome, Sesso, DataDiNascita, Nazionalità)
- Interpretazione (CodiceFilm,CodiceAttore,Personaggio, SessoPersonaggio)
- Regista (CodiceRegista, Cognome, Nome, Sesso, DataDiNascita, Nazionalità)
- Noleggio (CodiceFilm, Incasso1annoVideo, Incasso1annoDVD)

Differenza e nidificazione

```
select Nome from Persone
except
select Figlio as Nome from Maternita
```

```
select Nome
from Persone p
where not exists (select *
                  from Maternita
                  where Figlio = p.Nome)
```

Formulare in SQL la seguente interrogazione (1)

1.nomi e cognomi dei registi che hanno diretto film che hanno incassato meno nelle sale che per il noleggio di DVD

Possibili interrogazioni SQL

```
select Nome,Cognome
from Regista as R, Film as F, Produzione as P, Noleggio
as N
where      (R.CR=F.CR) and (F.CF=P.CF) and
           (P.CF=N.CF) and (Inc1sala<Inc1DVD)
```

```
select Nome,Cognome
from Regista as R join Film as F on (R.CR=F.CR)
      join Produzione as P on (F.CF=P.CF)
      join Noleggio as N on (P.CF=N.CF)
where (Inc1sala<Inc1DVD)
```

Formulare in SQL la seguente interrogazione (3)

I titoli dei film i cui attori sono tutti dello stesso sesso

Interrogazione nidificata

```
select Nome,Cognome from Regista
where CR in
      (select CR from Film
       where CF in
         ( select CF from Produzione
          where CF in
            (select CF from Noleggio
             where (Inc1sala<Inc1DVD))
          )
        )
      )
```

```
select Titolo from Film as F
where not exists
  (select I1.CF
   from Interpretazione as I1 natural join Artista as A1
   where I1.CF in  (select A2.CA
                   from Interpretazione as I2 join Artista as A2
                   (I2.CF=F.CF) and (I2.CA=A2.CA)
                   where (A2.S<>A1.S)
                  )
  )
```

```

select Titolo from Film
except
select F.Titolo
from Film as F join
  (select CF from Artista as A join Interpretazione as I
   on A.CA=I.CA) on F.CF=I.CF
where Sesso='M'
union
select Titolo from Film
except
select Titolo
from Film as F join
  (select CF from Artista as A join Interpretazione as I
   on A.CA=I.CA)
  on F.CF=I.CF
where Sesso='F'

```

Formulare in SQL la seguente interrogazione (4)

**I titoli di film con solamente attori
donna che abbiano incassato in sala
più del proprio costo**

- La query precedente non è equivalente a

```

select Titolo from Film
except
select F.Titolo
from Film as F join
  (select CF from Artista as A join Interpretazione
   as I on A.CA=I.CA)
  on F.CF=I.CF
where (Sesso='M') or (Sesso='F')

```

```

select Titolo
  from Film as F join Produzione as P
    on F.CF=P.CF
  where (Inc1S>Costo)
except
select Titolo
From (Artista as A join Interpretazione as I
  on A.CA=I.CA) join Film as F on F.CF=I.CF
where Sesso='M'

```

```

select Titolo
  from Film as F join Produzione as P
    on F.CF=P.CF
  where (Inc1S>Costo) and Titolo not in
(select Titolo
 From (Artista as A join Interpretazione as I
    on A.CA=I.CA)
 where Sesso='M' and F.CF=I.CF

```

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

Funzione ([DISTINCT] *)
 Funzione ([DISTINCT] Attributo)

Operatori aggregati

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

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'
```

4

COUNT DISTINCT

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

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

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

COUNT e valori nulli

```
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 e valori nulli

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

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

Massimo e nidificazione

- La persona (o le persone) con il reddito massimo

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

Operatori aggregati e target list

- un'interrogazione scorretta:

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

- Cosa succede

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

Operatori aggregati e raggruppamenti

- Le funzioni possono essere applicate a partizioni delle relazioni

- Clausola GROUP BY:
GROUP BY listaAttributi

Operatori aggregati e raggruppamenti

- Il numero di figli di ogni padre

```
select Padre, count(*) as NumFigli
from paternita
group by Padre
```

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

Padre	NumFigli
Sergio	1
Luigi	2
Franco	2

Raggruppamenti e target list

scorretta

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

corretta

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

corretta

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

Esecuzione

1. interrogazione pura

```
select *
from paternita
```

2. si raggruppa e si applica l'operatore aggregato a ciascun gruppo

Condizioni sui gruppi

- I padri i cui figli hanno un reddito medio maggiore di 25; mostrare padre e reddito medio dei figli

```
select padre, avg(reddito)
from persone join paternita on figlio =
     nome
group by padre
having avg(reddito) > 25
```

WHERE o HAVING?

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

```
select padre, avg(reddito)
from persone join paternita on figlio = nome
where eta < 30
group by padre
having avg(reddito) > 20
```

Esercizio 1

Si consideri la seguente base di dati relativa all'Università di Pisa:

Facoltà (ID_fac, Nome_fac, Indirizzo, Preside)

CorsiLaurea (ID_corso, Nome_corso, ID_fac, Num_esami)

Esami (ID_es, ID_corso, Num_credits, Tipo)

Il campo Tipo della tabella Esami contiene "O" se l'esame è obbligatorio ed "F" se l'esame è facoltativo.

Scrivere una query SQL che produca come risultato il numero degli esami obbligatori di Ingegneria.

Group by e valori nulli

```
select B, count(*)
from R group by B
```

A	B
1	11
2	11
3	null
4	null

B	
11	2
null	2

```
select A, count (*)
from R group by A
```

A	
1	1
2	1
3	1
4	1

Soluzione

```
select count (*)
from ( Facoltà f join CorsiLaurea c
      on f.ID_fac = c.ID_fac ) join
Esami e on e.ID_corso = c.ID_corso
where Tipo = 'O' and Nome_Fac =
"Ingegneria"
```

Esercizio 2

- Film (ID_film, Titolo, ID_regista, CasaProd, Genere, Durata)
 - Attori (ID_attore, Nome, Cognome, Nazionalità)
 - Registi (ID_regista, Nome, Cognome, Nazionalità)
 - Interpreti (ID_film, ID_attore)
- Scrivere una query SQL che produca come risultato la lista dei TITOLI e degli ID dei film che sono stati diretti da un regista inglese e che sono stati interpretati da almeno 3 attori inglesi.

Soluzione

```
select  Titolo, ID_film
from    Film f1, Registi r
where   f1.ID_regista = r.ID_regista and
        r. Nazionalita = "EN" and
        f1.ID_film = any ( select i.ID_film
                           from   Attori a, Interpreti i
                           where  f1.ID_film = i.ID_film      and
                           a.ID_attore = i.ID_attore and
                           a.Nazionalità = "EN"
                           group by    i.ID_film
                           having      COUNT (*) >= 3 )
```