

RICORSIONE IN SQL:1999

- in SQL2 non é possibile definire interrogazioni che facciano uso della ricorsione
- ad esempio se consideriamo la relazione
`Voli(lineaAerea,da,a,parte,arriva)`
non é possibile esprimere l'interrogazione che ritrova tutte le città raggiungibili l'una dall'altra (con un numero arbitrario di tappe intermedie)
- per risolvere queste interrogazioni si utilizza SQL da programma
- in SQL:1999 é stata aggiunta la possibilità di esprimere interrogazioni ricorsive
- base teorica: basi di dati deduttive

1

RICORSIONE IN SQL:1999

- comando WITH per definire relazioni IDB
`WITH R AS <definizione di R>`
`<interrogazione che coinvolge R>`
- si definisce cioè una relazione temporanea chiamata R e si usa R in un'interrogazione
 - si possono definire più relazioni (separate da virgola)
 - ognuna di queste definizioni può essere ricorsiva e le relazioni possono essere mutuamente ricorsive
 - ogni relazione coinvolta in una ricorsione deve essere preceduta dalla parola chiave `RECURSIVE`
- solo ricorsione *lineare*: solo un sottogol ricorsivo in ogni regola

2

RICORSIONE IN SQL:1999

1. parola chiave WITH
2. una o più definizioni, separate da virgole
ogni definizione consiste in
 - (a) parola chiave opzionale `RECURSIVE`
(deve essere specificata se la relazione che si sta definendo è ricorsiva)
 - (b) il nome della relazione che si sta definendo
 - (c) la parola chiave `AS`
 - (d) l'interrogazione che definisce la relazione
3. interrogazione che può far riferimento alle precedenti definizioni e forma il risultato del comando WITH

3

RICORSIONE IN SQL:1999

Esempio `Voli(lineaAerea,da,a,parte,arriva)`

- predicato `Raggiunge` (non lineare, quindi non ammesso in SQL:1999)

```
WITH RECURSIVE Raggiunge(da,a) AS
  (SELECT da,a FROM Voli)
  UNION
  (SELECT R1.da, R2.a
   FROM Raggiunge AS R1, Raggiunge AS R2
   WHERE R1.a = R2.a)
SELECT * FROM Raggiunge;
```

- predicato `Raggiunge` (lineare)

```
WITH Coppie AS SELECT da,a FROM Voli,
  RECURSIVE Raggiunge(da,a) AS
  Coppie
  UNION
  (SELECT Coppie.da, Raggiunge.a
   FROM Coppie, Raggiunge
   WHERE Coppie.a = Raggiunge.da)
SELECT * FROM Raggiunge;
```

4

RICORSIONE IN SQL:1999

- le definizioni all'interno del comando **WITH** sono disponibili solo all'interno del comando e non possono essere usate al di fuori di questo
- nel comando **WITH** si possono definire viste invece che tabelle
 la differenza sintattica è che si usa la parola chiave **VIEW** nella definizione della relazione
- esempio:
`VIEW Coppie AS SELECT da, a FROM Voli`
- se si definisce *Coppie* come vista, tale relazione non viene effettivamente costruita, ma il suo uso nella costruzione di *Raggiunge* è sostituito dall'uso delle componenti dalle tuple di *Vol*i

5

RICORSIONE IN SQL:1999

- la semantica delle interrogazioni ricorsive è definita mediante la nozione di punto fisso
- si costruisce una sequenza R_i di relazioni tali che:
 - R_0 è la relazione vuota
 - $R_i, 1 \leq i$, viene ottenuta applicando la definizione della relazione a R_{i-1}
 - quando, per un certo i , si ha che $R_i = R_{i-1}$ ci si ferma e tale relazione è il risultato dell'interrogazione

6

RICORSIONE IN SQL:1999

esempio (determinare i cammini di un grafo dati gli archi):

```
WITH RECURSIVE Cammino(da,a) AS
  (SELECT da,a FROM Arco)
  UNION
  (SELECT R1.da, R2.a
   FROM Arco AS R1, Cammino AS R2
   WHERE R1.a = R2.a);
SELECT * FROM Cammino;
```

Arco	da	a
	a	b
	b	c
	c	d
	d	b

RICORSIONE IN SQL:1999

- $Cammino_0 = \emptyset$

- $Cammino_1 =$

da	a
a	b
b	c
c	d
d	b

- $Cammino_2 = Cammino_1 \cup$

da	a
a	c
b	d
c	b
d	c

- $Cammino_3 = Cammino_2 \cup$

da	a
a	d
b	b
c	c
d	d

- $Cammino_4 = Cammino_3$
 punto fisso e risultato

7

8

RICORSIONE E NEGAZIONE

RICORSIONE E NEGAZIONE

- in alcuni casi è necessario utilizzare la negazione nella definizione di un predicato ricorsivo
- questo può però dar luogo a problemi nel definire la semantica
- la nozione di *negazione stratificata* permette di limitare l'uso della ricorsione a interrogazioni la cui semantica è ben definita

9

esempio non ok

```
WITH
  RECURSIVE P(X) AS
    (SELECT * FROM R)
  EXCEPT
    (SELECT * FROM Q),
  RECURSIVE Q(X) AS
    (SELECT * FROM R)
  EXCEPT
    (SELECT * FROM P)
SELECT * FROM P;
```

se R contiene solo 0 l'interrogazione ha due possibili risposte:
 $\{R(0), P(0)\}$ e $\{R(0), Q(0)\}$
 ma non c'è modo di scegliere tra le due

10

RICORSIONE E NEGAZIONE

RICORSIONE E NEGAZIONE

esempio ok

relazione $U_{\text{solo}}(X,Y)$: UA vola da X a Y (anche attraverso altre città), ma AA no

```
WITH
  Triple AS SELECT lineaAerea, da, a FROM Voli,
  RECURSIVE Raggiunge(lineaAerea,da,a) AS
    Triple
  UNION
    (SELECT Triple.lineaAerea, Triple.da,
      Raggiunge.a
     FROM Triple, Raggiunge
     WHERE Triple.a = Raggiunge.da AND
           Triple.lineaAerea = Raggiunge.lineaAerea)
  (SELECT da,a FROM Raggiunge
   WHERE lineaAerea = 'UA')
EXCEPT (SELECT da,a FROM Raggiunge
  WHERE lineaAerea = 'AA');
```

differenza insiemistica di U_{aragg} e AA_{aragg} : la semantica è ben definita

11

RICORSIONE E NEGAZIONE

esempio (segue)

	ua	sf	den	930	1230
	aa	sf	dal	900	1430
	ua	den	chi	1500	1800
Voli	ua	den	dal	1400	1700
	aa	dal	chi	1530	1730
	aa	dal	ny	1500	1900
	aa	chi	ny	1900	2200
	ua	chi	ny	1830	2130

la prima sottointerrogazione restituisce le seguenti coppie:

(sf,den) , (sf,dal) , (sf,chi) , (sf,ny) ,
 (den,dal) , (den,chi) , (den,ny) , (chi,ny)

la seconda sottointerrogazione restituisce le seguenti coppie:

(sf,dal) , (sf,chi) , (sf,ny) , (dal,chi) , (dal,ny) ,
 (chi,ny)

il risultato è la differenza di questi insiemi di coppie: (sf,den) , (den,dal) , (den,chi) , (den,ny)

12

STRATIFICAZIONE

- per evitare i problemi dovuti all'uso della ricorsione attraverso la negazione, ci si restringe alla ricorsione in cui la negazione è stratificata
- quando la negazione è stratificata esiste un algoritmo per calcolare un particolare minimo punto fisso, che corrisponde al contenuto informativo "intuitivo"
- stratificazione:
 - grafo i cui nodi corrispondono alle relazioni
 - arco dal nodo A al nodo B etichettato da - se nella definizione di A compare B negato
 - arco dal nodo A al nodo B se nella definizione di A compare B non-negato

13

STRATIFICAZIONE

- se il grafo ha un ciclo che contiene uno o più archi negativi la ricorsione non è stratificata, altrimenti il grafo è stratificato
- i predicati IDB possono essere raggruppati in strati: lo strato di un predicato A è il più grande numero di archi negativi su un cammino che comincia da A
- se la ricorsione è stratificata è possibile calcolare il contenuto delle relazioni in accordo alla semantica di punto fisso nell'ordine dei loro strati, partendo dal più basso

14

RICORSIONE IN SQL:1999

- viene richiesta la stratificazione anche rispetto ad altri costrutti non monotoni, ad esempio aggregati

WITH

```
    RECURSIVE P(X) AS
      (SELECT * FROM R)
    UNION
      (SELECT * FROM Q),
    RECURSIVE Q(X) AS
      SELECT SUM(X) FROM P
SELECT * FROM P;
```

es. R contiene 12 e 34 Q passa da \emptyset a $\{(46)\}$ a $\{92\} \Rightarrow$ non monotono

15