

Atzeni, Ceri, Paraboschi, Torlone
Basi di dati
McGraw-Hill, 1996-2002

Capitolo 3:
ALGEBRA E CALCOLO
RELAZIONALE

07/06/2005

Linguaggi per basi di dati

- operazioni sullo schema
 - DDL: data definition language
- operazioni sui dati
 - DML: data manipulation language
 - interrogazione ("query")
 - aggiornamento

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

2

Linguaggi di interrogazione per basi di dati relazionali

- Dichiarativi
 - specificano le proprietà del risultato ("che cosa")
- Procedurali
 - specificano le modalità di generazione del risultato ("come")

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

3

Linguaggi di interrogazione

- Algebra relazionale: procedurale
- Calcolo relazionale:
 - dichiarativo (teorico)
 - SQL (Structured Query Language):
parzialmente dichiarativo (reale)
- QBE (Query by Example):
 - dichiarativo (reale)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

4

Algebra relazionale

- Insieme di operatori
 - su relazioni
 - che producono relazioni
 - e possono essere composti

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

5

Operatori dell'algebra relazionale

- unione, intersezione, differenza
- ridenominazione
- selezione
- proiezione
- join (join naturale, prodotto cartesiano, theta-join)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

6

Operatori insiemistici

- le relazioni sono insiemi
- i risultati debbono essere relazioni
- è possibile applicare unione, intersezione, differenza solo a relazioni definite sugli stessi attributi

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

7

Unione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cup Quadri

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45
9297	Neri	33

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

8

Intersezione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cap Quadri

Matricola	Nome	Età
7432	Neri	54
9824	Verdi	45

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

9

Differenza

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati - Quadri

Matricola	Nome	Età
7274	Rossi	42

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

10

Un'unione sensata ma impossibile

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Paternità \cup Maternità

??

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

11

Ridenominazione

- operatore monadico (con un argomento)
- "modifica lo schema" lasciando inalterata l'istanza dell'operando

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

12

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

REN_{Genitore ← Padre} (Paternità)

Genitore	Figlio
Adamo	Abele
Abramo	Isacco

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 13

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

REN_{Genitore ← Padre} (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

REN_{Genitore ← Madre} (Maternità)

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 14

REN_{Genitore ← Padre} (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

REN_{Genitore ← Padre} (Paternità)

⌋

REN_{Genitore ← Madre} (Maternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco
Eva	Abele
Eva	Set
Sara	Isacco

REN_{Genitore ← Madre} (Maternità)

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 15

Impiegati

Cognome	Ufficio	Stipendio
Rossi	Roma	55
Neri	Milano	64

Operai

Cognome	Fabbrica	Salario
Bruni	Monza	45
Verdi	Latina	55

REN_{Sede, Retribuzione ← Ufficio, Stipendio} (Impiegati)

⌋

REN_{Sede, Retribuzione ← Fabbrica, Salario} (Operai)

Cognome	Sede	Retribuzione
Rossi	Roma	55
Neri	Milano	64
Bruni	Monza	45
Verdi	Latina	55

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 16

Selezione

- operatore monadico
- produce un risultato che
 - ha lo stesso schema dell'operando
 - contiene un sottoinsieme delle ennuple dell'operando,
 - quelle che soddisfano una condizione

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 17

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Neri	Napoli	64

- impiegati che
 - guadagnano più di 50
 - guadagnano più di 50 e lavorano a Milano
 - hanno lo stesso nome della filiale presso cui lavorano

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 18

Selezione, sintassi e semantica

- sintassi
 - $SEL_{Condizione} (Operando)$
 - **Condizione**: espressione booleana (come quelle dei vincoli di ennupla)
- semantica
 - il risultato contiene le ennupe dell'operando che soddisfano la condizione

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

19

- impiegati che guadagnano più di 50

SEL _{Stipendio > 50} (Impiegati)			
Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
5698	Neri	Napoli	64

SEL_{Stipendio > 50} (Impiegati)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

20

- impiegati che guadagnano più di 50 e lavorano a Milano

SEL_{Stipendio > 50 AND Filiale = 'Milano'} (Impiegati)

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

SEL_{Stipendio > 50 AND Filiale = 'Milano'} (Impiegati)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

21

- impiegati che hanno lo stesso nome della filiale presso cui lavorano

SEL_{Cognome = Filiale} (Impiegati)

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

SEL_{Cognome = Filiale} (Impiegati)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

22

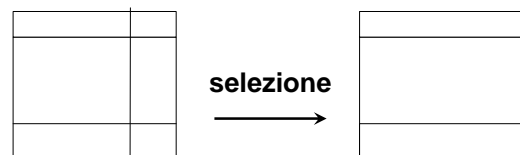
Selezione e proiezione

- operatori "ortogonali"
- selezione:
 - decomposizione orizzontale
- proiezione:
 - decomposizione verticale

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

23



07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

24

Proiezione

- operatore monadico
- produce un risultato che
 - ha parte degli attributi dell'operando
 - contiene ennuple cui contribuiscono tutte le ennuple dell'operando

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

25

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Neri	Napoli	55
5998	Neri	Milano	64
9553	Rossi	Roma	44
5698	Rossi	Roma	64

- per tutti gli impiegati:
 - matricola e cognome
 - cognome e filiale

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

26

Proiezione, sintassi e semantica

- sintassi
 $PROJ_{ListaAttributi}$ (Operando)
- semantica
 - il risultato contiene le ennuple ottenute da tutte le ennuple dell'operando ristrette agli attributi nella lista

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

27

- matricola e cognome di tutti gli impiegati

Matricola	Cognome
7309	Neri
5998	Neri
9553	Rossi
5698	Rossi

$PROJ_{Matricola, Cognome}$ (Impiegati)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

28

- cognome e filiale di tutti gli impiegati

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma

$PROJ_{Cognome, Filiale}$ (Impiegati)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

29

Cardinalità delle proiezioni

- una proiezione
 - contiene al più tante ennuple quante l'operando
 - può contenerne di meno
- se X è una superchiave di R , allora $PROJ_X(R)$ contiene esattamente tante ennuple quante R

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

30

Selezione e proiezione

- Combinando selezione e proiezione, possiamo estrarre interessanti informazioni da una relazione

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

31

- matricola e cognome degli impiegati che guadagnano più di 50

Matricola	Cognome		
7309	Rossi		
5998	Neri		
5698	Neri		

PROJ_{Matricola,Cognome} (SEL_{Stipendio > 50} (Impiegati))

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

32

- Combinando selezione e proiezione, possiamo estrarre informazioni da una relazione
- non possiamo però correlare informazioni presenti in relazioni diverse, né informazioni in ennuple diverse di una stessa relazione

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

33

Join

- il join è l'operatore più interessante dell'algebra relazionale
- permette di correlare dati in relazioni diverse

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

34

Prove scritte in un concorso pubblico

- I compiti sono anonimi e ad ognuno è associata una busta chiusa con il nome del candidato
- Ciascun compito e la relativa busta vengono contrassegnati con uno stesso numero

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

35

1	25	1	Mario Rossi
2	13	2	Nicola Russo
3	27	3	Mario Bianchi
4	28	4	Remo Neri

Mario Rossi	25
Nicola Russo	13
Mario Bianchi	27
Remo Neri	28

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

36

Numero	Voto	Numero	Candidato
1	25	1	Mario Rossi
2	13	2	Nicola Russo
3	27	3	Mario Bianchi
4	28	4	Remo Neri

Numero	Candidato	Voto
1	Mario Rossi	25
2	Nicola Russo	13
3	Mario Bianchi	27
4	Remo Neri	28

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

37

Join naturale

- operatore binario (generalizzabile)
- produce un risultato
 - sull'unione degli attributi degli operandi
 - con ennuple costruite ciascuna a partire da una ennupla di ognuno degli operandi

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

38

Join, sintassi e semantica

- $R_1(X_1), R_2(X_2)$
 - $R_1 \text{ JOIN } R_2$ è una relazione su $X_1 X_2$
- $$\{ t \text{ su } X_1 X_2 \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2 \text{ con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

39

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Mori
B	Bruni

Impiegato	Reparto	Capo
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

- ogni ennupla contribuisce al risultato:
 - join completo

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

40

Un join non completo

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

41

Un join vuoto

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
D	Mori
C	Bruni

Impiegato	Reparto	Capo
-----------	---------	------

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

42

Un join completo, con n x m ennuple

Impiegato	Reparto	Reparto	Capo
Rossi	B	B	Mori
Neri	B	B	Bruni

Impiegato	Reparto	Capo
Rossi	B	Mori
Rossi	B	Bruni
Neri	B	Mori
Neri	B	Bruni

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

43

Cardinalità del join

- Il join di R_1 e R_2 contiene un numero di ennuple compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$
- se il join coinvolge una chiave di R_2 , allora il numero di ennuple è compreso fra zero e $|R_1|$
- se il join coinvolge una chiave di R_2 e un vincolo di integrità referenziale, allora il numero di ennuple è pari a $|R_1|$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

44

Cardinalità del join, 2

- $R_1(A,B)$, $R_2(B,C)$
- in generale

$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1| \times |R_2|$$
- se B è chiave in R_2

$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1|$$
- se B è chiave in R_2 ed esiste vincolo di integrità referenziale fra B (in R_1) e R_2 :

$$|R_1 \text{ JOIN } R_2| = |R_1|$$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

45

Join, una difficoltà

Impiegato	Reparto	Reparto	Capo
Rossi	A	B	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

- alcune ennuple non contribuiscono al risultato: vengono "tagliate fuori"

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

46

Join esterno

- Il join esterno estende, con valori nulli, le ennuple che verrebbero tagliate fuori da un join (interno)
- esiste in tre versioni:
 - sinistro, destro, completo

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

47

Join esterno

- sinistro: mantiene tutte le ennuple del primo operando, estendendole con valori nulli, se necessario
- destro: ... del secondo operando ...
- completo: ... di entrambi gli operandi ...

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

48

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{LEFT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 49

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{RIGHT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 50

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{FULL} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 51

Join e proiezioni

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{RIGHT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

Impiegati JOIN_{LEFT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 52

Proiezioni e join

Impiegati

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Bruni
Verdi	A	Bini

Reparti

Reparto	Capo
B	Mori
B	Bruni
A	Bini

Impiegati JOIN_{RIGHT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Neri	B	Bruni
Bianchi	B	Mori
Bianchi	B	Bruni
Verdi	A	Bini

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 53

Join e proiezioni

- $R_1(X_1), R_2(X_2)$

$$PROJ_{X_1}(R_1 JOIN R_2) \subseteq R_1$$

- $R(X), X = X_1 \cup X_2$

$$(PROJ_{X_1}(R)) JOIN (PROJ_{X_2}(R)) \supseteq R$$

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 54

Prodotto cartesiano

- un join naturale su relazioni senza attributi in comune
- contiene sempre un numero di ennuple pari al prodotto delle cardinalità degli operandi (le ennuple sono tutte combinabili)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

55

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
Rossi	A	B	Bruni
Neri	B	A	Mori
Neri	B	B	Bruni
Bianchi	B	A	Mori
Bianchi	B	B	Bruni

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

56

- Il prodotto cartesiano, in pratica, ha senso (quasi) solo se seguito da selezione:

$SEL_{Condizione} (R_1 JOIN R_2)$

- L'operazione viene chiamata theta-join e indicata con

$R_1 JOIN_{Condizione} R_2$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

57

Perché "theta-join"?

- La condizione C è spesso una congiunzione (AND) di atomi di confronto $A_1 \vartheta A_2$ dove ϑ è uno degli operatori di confronto ($=, >, <, \dots$)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

58

Equi-join

- Se l'operatore di confronto nel theta-join è sempre l'uguaglianza ($=$) allora si parla di equi-join

Nota: ci interessa davvero l'equi-join, non il theta-join più generale

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

59

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN_{Reparto=Codice} Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

60

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
A	Mori
B	Bruni

Impiegati JOIN Reparti

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 61

Join naturale ed equi-join

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
A	Mori
B	Bruni

Impiegati JOIN Reparti

PROJ_{Impiegato,Reparto,Capo} (SEL_{Reparto=Codice} (Impiegati JOIN REN_{Codice ← Reparto} (Reparti)))

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 62

Esempi

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Supervisione

Impiegato	Capo
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 63

• Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni

SEL_{Stipendio>40}(Impiegati)

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 64

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

SEL_{Stipendio>40}(Impiegati)

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 65

• Trovare matricola, nome ed età degli impiegati che guadagnano più di 40 milioni

PROJ_{Matricola, Nome, Età} (SEL_{Stipendio>40}(Impiegati))

07/06/2005 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 66

Matricola	Nome	Età
7309	Rossi	34
5698	Bruni	43
4076	Mori	45
8123	Lupi	46

**PROJ_{Matricola, Nome, Età}
(SEL_{Stipendio>40}(Impiegati))**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

67

- Trovare le matricole dei capi degli impiegati che guadagnano più di 40 milioni

**PROJ_{Capo} (Supervisione
JOIN_{Impiegato=Matricola}
(SEL_{Stipendio>40}(Impiegati)))**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

68

- Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 40 milioni

**PROJ_{Nome,Stipendio} (
Impiegati JOIN_{Matricola=Capo}
PROJ_{Capo}(Supervisione
JOIN_{Impiegato=Matricola} (SEL_{Stipendio>40}(Impiegati))))**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

69

- Trovare gli impiegati che guadagnano più del proprio capo, mostrando matricola, nome e stipendio dell'impiegato e del capo

**PROJ_{Matr, Nome, Stip, MatrC, NomeC, StipC}
(SEL_{Stipendio>StipC}(
REN_{MatrC, NomeC, StipC, EtàC ← Matr, Nome, Stip, Età}(Impiegati)
JOIN_{MatrC=Capo}
(Supervisione JOIN_{Impiegato=Matricola} Impiegati)))**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

70

- Trovare le matricole dei capi i cui impiegati guadagnano tutti più di 40 milioni

**PROJ_{Capo} (Supervisione) -
PROJ_{Capo} (Supervisione
JOIN_{Impiegato=Matricola}
(SEL_{Stipendio ≤ 40}(Impiegati)))**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

71

Equivalenza di espressioni

- Due espressioni sono equivalenti se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati
- L'equivalenza è importante in pratica perché i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "costose"

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

72

Un'equivalenza importante

- Push selections (se A è attributo di R_2)
 $SEL_{A=10}(R_1 JOIN R_2) = R_1 JOIN SEL_{A=10}(R_2)$
- Riduce in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

73

Nota

- In questo corso, ci preoccupiamo poco dell'efficienza:
 - l'obiettivo è di scrivere interrogazioni corrette e leggibili
- Motivazione:
 - I DBMS si preoccupano di scegliere le strategie realizzative efficienti

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

74

Selezione con valori nulli

Impiegati

Matricola	Cognome	Filiale	Età
7309	Rossi	Roma	32
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

$SEL_{Età > 40}(\text{Impiegati})$

- la condizione atomica è vera solo per valori non nulli

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

75

Un risultato non desiderabile

$SEL_{Età > 30}(\text{Persone}) \cup SEL_{Età \leq 30}(\text{Persone}) \neq \text{Persone}$

- Perché? Perché le selezioni vengono valutate separatamente!
- Ma anche
 $SEL_{Età > 30 \vee Età \leq 30}(\text{Persone}) \neq \text{Persone}$
- Perché? Perché anche le condizioni atomiche vengono valutate separatamente!

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

76

Selezione con valori nulli: soluzione

$SEL_{Età > 40}(\text{Impiegati})$

- la condizione atomica è vera solo per valori non nulli
- per riferirsi ai valori nulli esistono forme apposite di condizioni:
 $IS\ NULL$
 $IS\ NOT\ NULL$
- si potrebbe usare (ma non serve) una "logica a tre valori" (vero, falso, sconosciuto)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

77

- Quindi:

$SEL_{Età > 30}(\text{Persone}) \cup SEL_{Età \leq 30}(\text{Persone}) \cup$
 $SEL_{Età\ IS\ NULL}(\text{Persone})$
 $=$
 $SEL_{Età > 30 \vee Età \leq 30 \vee Età\ IS\ NULL}(\text{Persone})$
 $=$
 Persone

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

78

Impiegati

Matricola	Cognome	Filiale	Età
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

SEL (Età > 40) OR (Età IS NULL) (Impiegati)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

79

Viste (relazioni derivate)

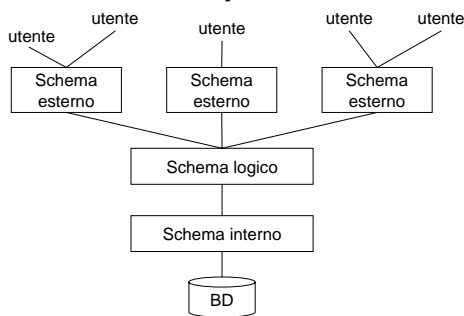
- Rappresentazioni diverse per gli stessi dati (**schema esterno**)
- Relazioni derivate:
 - relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- Relazioni di base: contenuto autonomo
- Le relazioni derivate possono essere definite su altre derivate, ma ...

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

80

Architettura standard (ANSI/SPARC) a tre livelli per DBMS



07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

81

Viste, esempio

Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Direzione

Reparto	Capo
A	Mori
B	Bruni

- una vista:
Supervisione = PROJ_{Impiegato, Capo} (Afferenza JOIN Direzione)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

82

Viste virtuali e materializzate

- Due tipi di relazioni derivate:
 - viste materializzate
 - relazioni virtuali (o viste)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

83

Viste materializzate

- relazioni derivate memorizzate nella base di dati
 - vantaggi:
 - immediatamente disponibili per le interrogazioni
 - svantaggi:
 - ridondanti
 - appesantiscono gli aggiornamenti
 - sono raramente supportate dai DBMS

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

84

Viste virtuali

- relazioni virtuali (o viste):
 - sono supportate dai DBMS (tutti)
 - una interrogazione su una vista viene eseguita "ricalcolando" la vista (o quasi)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

85

Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

```

SEL_Capo='Leoni' (Supervisione)
viene eseguita come
SEL_Capo='Leoni' (
PROJ Impiegato, Capo (Afferenza JOIN Direzione))
    
```

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

86

Viste, motivazioni

- Schema esterno: ogni utente vede solo
 - ciò che gli interessa e nel modo in cui gli interessa, senza essere distratto dal resto
 - ciò che è autorizzato a vedere (autorizzazioni)
 - Strumento di programmazione:
 - si può semplificare la scrittura di interrogazioni: espressioni complesse e sottoespressioni ripetute
 - Utilizzo di programmi esistenti su schemi ristrutturati
- Invece:
- L'utilizzo di viste non influisce sull'efficienza delle interrogazioni

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

87

Viste come strumento di programmazione

- Trovare gli impiegati che hanno lo stesso capo di Rossi
- Senza vista:


```

PROJ Impiegato (Afferenza JOIN Direzione) JOIN
REN ImpR,RepR ← Imp,Reparto (
SEL Impiegato='Rossi' (Afferenza JOIN Direzione))
            
```
- Con la vista:


```

PROJ Impiegato (Supervisione) JOIN
REN ImpR,RepR ← Imp,Reparto (
SEL Impiegato='Rossi' (Supervisione))
            
```

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

88

Viste e aggiornamenti, attenzione

Afferenza		Direzione	
Impiegato	Reparto	Reparto	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Verdi	A	C	Bruni

Supervisione	Impiegato	Capo
	Rossi	Mori
	Neri	Bruni
	Verdi	Mori

- Vogliamo inserire, nella vista, il fatto che Lupi ha come capo Bruni; oppure che Belli ha come capo Falchi; come facciamo?

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

89

Viste e aggiornamenti

- "Aggiornare una vista":
 - modificare le relazioni di base in modo che la vista, "ricalcolata" rispecchi l'aggiornamento
- L'aggiornamento sulle relazioni di base corrispondente a quello specificato sulla vista deve essere univoco
- In generale però non è univoco!
- Ben pochi aggiornamenti sono ammissibili sulle viste

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

90

Una convenzione e notazione alternativa per i join

- Nota: è sostanzialmente l'approccio usato in SQL
- Ignoriamo il join naturale (cioè non consideriamo implicitamente condizioni su attributi con nomi uguali)
- Per "riconoscere" attributi con lo stesso nome gli premettiamo il nome della relazione
- Usiamo "assegnazioni" (viste) per ridenominare le relazioni (e gli attributi solo quando serve per l'unione)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

91

- Trovare gli impiegati che guadagnano più del proprio capo, mostrando matricola, nome e stipendio dell'impiegato e del capo

```

PROJMatr, Nome, Stip, MatrC, NomeC, StipC
  (SELStipendio>StipC(
RENMatrC, NomeC, StipC, EtàC ← Matr, Nome, Stip, Età(Impiegati)
  JOINMatrC=Capo
(Supervisione JOINImpiegato=Matricola Impiegati)))
    
```

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

92

```

PROJMatr, Nome, Stip, MatrC, NomeC, StipC
  (SELStip>StipC(
RENMatrC, NomeC, StipC, EtàC ← Matr, Nome, Stip, Età(Imp)
  JOINMatrC=Capo
(Sup JOINImp=Matr Imp)))
    
```

Capi := Imp

```

PROJImp.Matr, Imp.Nome, Imp.Stip, Capi.Matr, Capi.Nome, Capi.Stip
  (SELImp.Stip>Capi.Stip(
Capi JOINCapi.Matr=Capo (Sup JOINImp=Imp.Matr Imp)))
    
```

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

93

Calcolo relazionale

- Una famiglia di linguaggi dichiarativi, basati sul calcolo dei predicati del primo ordine
- Diverse versioni:
 - calcolo relazionale su domini
 - calcolo su ennuple con dichiarazioni di range

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

94

Calcolo su domini, sintassi e semantica

- Le espressioni hanno la forma:
 - $\{ A_1: x_1, \dots, A_k: x_k \mid f \}$
 - f e' una formula (con connettivi booleani e quantificatori)
 - $A_1: x_1, \dots, A_k: x_k$ "target list":
 - A_1, \dots, A_k attributi distinti (anche non nella base di dati)
 - x_1, \dots, x_k variabili distinte
- Semantica: il risultato e' una relazione su A_1, \dots, A_k che contiene ennuple di valori per x_1, \dots, x_k che rendono vera la formula f

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

95

Commenti

- Differenze rispetto al calcolo dei predicati (per chi lo conosce):
 - simboli di predicato
 - relazioni nella base di dati
 - predicati "standard" predefiniti (=, >, ...)
 - non ci sono "simboli di funzione"
 - interessano (quasi) solo "formule aperte"
 - utilizziamo notazione non posizionale

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

96

Base di dati per gli esempi

Impiegati(Matricola, Nome, Età, Stipendio)
Supervisione(Capo, Impiegato)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

97

Esempio 0a

- Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni

SEL_{Stipendio>40}(Impiegati)

**{ Matricola: m, Nome: n, Età: e, Stipendio: s |
Impiegati(Matricola: m, Nome: n, Età: e,
Stipendio: s) ∧ s > 40 }**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

98

Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

PROJ_{Matricola, Nome, Età}(Impiegati)

**{ Matricola: m, Nome: n, Età: e |
∃s (Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)) }**

**{ Matricola: m, Nome: n, Età: e |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s) }**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

99

Esempio 1

- Trovare matricola, nome ed età degli impiegati che guadagnano più di 40 milioni

PROJ_{Matricola, Nome, Età}(SEL_{Stipendio>40}(Impiegati))

**{ Matricola: m, Nome: n, Età: e |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)
∧ s > 40 }**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

100

Esempio 2

- Trovare le matricole dei capi degli impiegati che guadagnano più di 40 milioni

**PROJ_{Capo}(Supervisione JOIN_{Impiegato=Matricola}
(SEL_{Stipendio>40}(Impiegati)))**

**{ Capo: c | Supervisione(Capo:c, Impiegato:m) ∧
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)
∧ s > 40 }**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

101

Esempio 3

- Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 40 milioni

**PROJ_{NomeC, StipC}
(REN_{MatrC, NomeC, StipC, EtàC ← Matr, Nome, Stip, Età}
JOIN_{MatrC=Capo}
(Supervisione JOIN_{Impiegato=Matricola}
(SEL_{Stipendio>40}(Impiegati))))**

**{ NomeC: nc, StipC: sc |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s) ∧
s > 40 ∧ Supervisione(Capo:c, Impiegato:m) ∧
Impiegati(Matricola:c, Nome:nc, Età:ec, Stipendio: sc) }**

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

102

Esempio 4

- Trovare gli impiegati che guadagnano più del rispettivo capo, mostrando matricola, nome e stipendio di ciascuno di essi e del capo

```

PROJMatr, Nome, Stip, MatrC, NomeC, StipC
(SEL-Stipendio>StipC(RENMatrC, NomeC, StipC, EtàC ←
  Matr, Nome, Stip, Età (Impiegati)
  JOINMatrC=Capo
  (Supervisione JOINImpiegato=Matricola (Impiegati))))
{ Matr: m, Nome: n, Stip: s, MatrC: c, NomeC: nc, StipC: sc |
  Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s) ∧
  Supervisione(Capo:c, Impiegato:m) ∧
  Impiegati(Matricola: c, Nome: nc, Età: ec, Stipendio: sc) ∧ s > sc}

```

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

103

Esempio 5

- Trovare matricola e nome dei capi i cui impiegati guadagnano tutti più di 40 milioni.

```

PROJMatricola, Nome (Impiegati JOINMatricola=Capo
  (PROJcapo (Supervisione) -
  PROJcapo (Supervisione JOINimpiegato=Matricola (
  SEL-Stipendio ≤ 40 (Impiegati))))
  {Matricola: c, Nome: n |
  Impiegati(Matricola: c, Nome: n, Età: e, Stipendio: s) ∧
  Supervisione(Capo:c, Impiegato:m) ∧
  ¬ ∃ m' (∃ n' (∃ e' (∃ s' (Impiegati(Matricola: m', Nome: n', Età: e', Stip: s') ∧
  Supervisione(Capo:c, Impiegato:m') ∧ s' ≤ 40))))}

```

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

104

Quantificatori esistenziali o universali?

- Sono intercambiabili, per le leggi di De Morgan:

```

{Matricola: c, Nome: n |
  Impiegati(Matricola: c, Nome: n, Età: e, Stipendio: s) ∧
  Supervisione(Capo:c, Impiegato:m) ∧
  ¬ ∃ m' (∃ n' (∃ e' (∃ s' (Impiegati(Matricola: m', Nome: n', Età: e', Stip: s') ∧
  Supervisione(Capo:c, Impiegato:m') ∧ s' ≤ 40))))}

```

```

{Matricola: c, Nome: n |
  Impiegati(Matricola: c, Nome: n, Età: e, Stipendio: s) ∧
  Supervisione(Capo:c, Impiegato:m) ∧
  ∀ m' (∀ n' (∀ e' (∀ s' (¬ (Impiegati(Matricola: m', Nome: n', Età: e', Stip: s') ∧
  Supervisione(Capo:c, Impiegato:m') ∧ s' > 40))))}

```

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

105

Calcolo su domini, discussione

- Pregi:
 - dichiaratività
- Difetti:
 - "verbosità": tante variabili!
 - espressioni senza senso:

$$\{ A: x \mid \neg R(A: x) \}$$

$$\{ A: x, B: y \mid R(A: x) \}$$

$$\{ A: x, B: y \mid R(A: x) \wedge y=y \}$$
 queste espressioni sono "dipendenti dal dominio" e vorremmo evitarle; nell'algebra espressioni come queste non sono formulabili: l'algebra è indipendente dal dominio

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

106

Calcolo e algebra

- Calcolo e algebra sono "equivalenti"
 - per ogni espressione del calcolo relazionale che sia indipendente dal dominio esiste un'espressione dell'algebra relazionale equivalente a essa
 - per ogni espressione dell'algebra relazionale esiste un'espressione del calcolo relazionale equivalente a essa (e di conseguenza indipendente dal dominio)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

107

Calcolo su ennuple con dichiarazioni di range

- Per superare le limitazioni del calcolo su domini:
 - dobbiamo "ridurre" le variabili; un buon modo: una variabile per ciascuna ennupla
 - far si' che i valori provengano dalla base di dati
- Il calcolo su ennuple con dichiarazioni di range risponde ad entrambe le esigenze

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

108

Calcolo su ennuple con dichiarazioni di range, sintassi

- Le espressioni hanno la forma:
 $\{ TargetList \mid RangeList \mid Formula \}$
- RangeList* elenca le variabili libere della *Formula* ognuna con il relativo campo di variabilità (una relazione)
- TargetList* ha elementi del tipo $Y: x.Z$ (oppure $x.Z$ o anche $x.*$)
- Formula* ha:
 - atomi di confronto $x.A \& c$, $x.A \& y.B$
 - connettivi
 - quantificatori che associano un range alle variabili

$\exists x(R)(\dots)$ $\forall x(R)(\dots)$
Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

07/06/2005

109

Esempio 0a

- Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni

$SEL_{Stipendio > 40}(Impiegati)$

$\{ Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s \mid$
 $Impiegati(Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s)$
 $\wedge s > 40 \}$

$\{ i.* \mid i(Impiegati) \mid i.Stipendio > 40 \}$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

110

Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

$PROJ_{Matricola, Nome, Et\grave{a}}(Impiegati)$

$\{ Matricola: m, Nome: n, Et\grave{a}: e \mid$
 $Impiegati(Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s) \}$

$\{ i.(Matricola, Nome, Et\grave{a}) \mid i(Impiegati) \}$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

111

Esempio 1

- Trovare matricola, nome ed età degli impiegati che guadagnano più di 40 milioni

$PROJ_{Matricola, Nome, Et\grave{a}}(SEL_{Stipendio > 40}(Impiegati))$

$\{ Matricola: m, Nome: n, Et\grave{a}: e \mid$
 $Impiegati(Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s)$
 $\wedge s > 40 \}$

$\{ i.(Matricola, Nome, Et\grave{a}) \mid i(Impiegati) \mid i.Stipendio > 40 \}$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

112

Esempio 2

- Trovare le matricole dei capi degli impiegati che guadagnano più di 40 milioni

$\{ Capo: c \mid Supervisione(Capo:c, Impiegato:m) \wedge$
 $Impiegati(Matricola: m, Nome: n, Et\grave{a}: e,$
 $Stipendio: s) \wedge s > 40 \}$

$\{ s.Capo \mid i(Impiegati), s(Supervisione) \mid$
 $i.Matricola=s.Impiegato \wedge i.Stipendio > 40 \}$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

113

Esempio 3

- Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 40 milioni

$\{ NomeC: nc, StipC: sc \mid$
 $Impiegati(Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s) \wedge$
 $s > 40 \wedge$
 $Supervisione(Capo:c, Impiegato:m) \wedge$
 $Impiegati(Matricola:c, Nome:nc, Et\grave{a}:ec, Stipendio:sc) \}$

$\{ NomeC, StipC: i'.(Nome, Stip) \mid$
 $i'(Impiegati), s(Supervisione), i(Impiegati) \mid$
 $i'.Matricola=s.Capo \wedge i.Matricola=s.Impiegato \wedge$
 $i.Stipendio > 40 \}$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

114

Esempio 4

- Trovare gli impiegati che guadagnano più del rispettivo capo, mostrando matricola, nome e stipendio di ciascuno di essi e del capo

$$\{ \text{Matr: m, Nome: n, Stip: s, NomeC: nc, StipC: sc} \mid \text{Impiegati}(\text{Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s}) \wedge \text{Supervisione}(\text{Capo: c, Impiegato: m}) \wedge \text{Impiegati}(\text{Matricola: c, Nome: nc, Et\grave{a}: ec, Stipendio: sc}) \wedge s > sc \}$$

$$\{ i.(\text{Nome, Matr, Stip}), \text{NomeC, MatrC, StipC: i'.}(\text{Nome, Matr, Stip}) \mid i'(\text{Impiegati}), s(\text{Supervisione}), i(\text{Impiegati}) \mid i'.\text{Matricola} = s.\text{Capo} \wedge i.\text{Matricola} = s.\text{Impiegato} \wedge i.\text{Stipendio} > i'.\text{Stipendio} \}$$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

115

Esempio 5

- Trovare matricola e nome dei capi i cui impiegati guadagnano tutti più di 40 milioni.

$$\{ \text{Matricola: c, Nome: n} \mid \text{Impiegati}(\text{Matricola: c, Nome: n, Et\grave{a}: e, Stipendio: s}) \wedge \text{Supervisione}(\text{Capo: c, Impiegato: m}) \wedge \neg \exists m'(\exists n'(\exists e'(\exists s'(\text{Impiegati}(\text{Matr: m}', \text{Nome: n}', \text{Et\grave{a}: e}', \text{Stip: s}')) \wedge \text{Supervisione}(\text{Capo: c, Impiegato: m}') \wedge s' \leq 40)) \}$$

$$\{ i.(\text{Matricola, Nome}) \mid s(\text{Supervisione}), i(\text{Impiegati}) \mid i.\text{Matricola} = s.\text{Capo} \wedge \neg(\exists i'(\text{Impiegati})(\exists s'(\text{Supervisione})(s.\text{Capo} = s'.\text{Capo} \wedge s'.\text{Impiegato} = i'.\text{Matricola} \wedge i'.\text{Stipendio} \leq 40))) \}$$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

116

Attenzione!

- Il calcolo su ennuple con dichiarazioni di range non permette di esprimere alcune interrogazioni importanti, in particolare le unioni:

$$R_1(AB) \cup R_2(AB)$$

- Quale potrebbe essere il range per una variabile? Oppure due variabili?
- Nota: intersezione e differenza sono esprimibili
- Per questa ragione SQL (che è basato su questo calcolo) prevede un operatore esplicito di unione, ma non tutte le versioni prevedono intersezione e differenza

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

117

Calcolo e algebra relazionale: limiti

- Calcolo e algebra sono sostanzialmente equivalenti: l'insieme di interrogazioni con essi esprimibili è quindi significativo; il concetto è robusto
- Ci sono però interrogazioni interessanti non esprimibili:
 - calcolo di valori derivati: possiamo solo estrarre valori, non calcolarne di nuovi; calcoli di interesse:
 - a livello di ennupla o di singolo valore (conversioni somme, differenze, etc.)
 - su insiemi di ennuple (somme, medie, etc.)
 le estensioni sono ragionevoli, le vedremo in SQL
 - interrogazioni inerentemente ricorsive, come la chiusura transitiva

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

118

Chiusura transitiva

Supervisione(Impiegato, Capo)

- Per ogni impiegato, trovare tutti i superiori (cioè il capo, il capo del capo, e così via)

Impiegato	Capo
Rossi	Lupi
Neri	Bruni
Lupi	Falchi

Impiegato	Superiore
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Rossi	Falchi

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

119

Chiusura transitiva, come si fa?

- Nell'esempio, basterebbe il join della relazione con se stessa, previa opportuna ridenominazione

- Ma:

Impiegato	Capo
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Falchi	Leoni

Impiegato	Superiore
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Falchi	Leoni
Rossi	Falchi
Lupi	Leoni
Rossi	Leoni

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

120

Chiusura transitiva, impossibile!

- Non esiste in algebra e calcolo relazionale la possibilità di esprimere l'interrogazione che, per ogni relazione binaria, ne calcoli la chiusura transitiva
- Per ciascuna relazione, è possibile calcolare la chiusura transitiva, ma con un'espressione ogni volta diversa:
 - quanti join servono?
 - non c'è limite!

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

121

Datalog

- Un linguaggio di programmazione logica per basi di dati derivato dal Prolog
- Utilizza predicati di due tipi:
 - estensionali: relazioni della base di dati
 - intensionali: corrispondono alle viste
- Il linguaggio è basato su regole utilizzate per "definire" i predicati estensionali

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

122

Datalog, sintassi

- Regole:
 - $testa \leftarrow corpo$
 - *testa* è un predicato atomico (intensionale)
 - *corpo* è una lista (congiunzione) di predicati atomici
- Le interrogazioni sono specificate per mezzo di predicati atomici (convenzionalmente preceduti da "?")

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

123

Esempio -1

- Trovare matricola, nome, età e stipendio degli impiegati che hanno 30 anni

{ Matricola: m, Nome: n, Età: e, Stipendio: s |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)
^ s = 30 }

? Impiegati(Matricola: m, Nome: n, Età: 30, Stipendio: s)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

124

Esempio 0a

- Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni
 - { Matricola: m, Nome: n, Età: e, Stipendio: s |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s) ^ s > 40 }
- Serve un predicato intensionale
 - ImpRicchi(Matricola: m, Nome: n, Età: e, Stipendio: s) ←
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s) , s > 40
 - ? ImpRicchi(Matricola: m, Nome: n, Età: e, Stipendio: s)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

125

Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

PROJ_{Matricola, Nome, Età}(Impiegati)

{ Matricola: m, Nome: n, Età: e |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)}

InfoPubbliche(Matricola: m, Nome: n, Età: e)
← Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)

? InfoPubbliche(Matricola: m, Nome: n, Età: e)

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

126

Esempio 2

- Trovare le matricole dei capi degli impiegati che guadagnano più di 40 milioni

$\{ \text{Capo: } c \mid \text{Supervisione}(\text{Capo:}c, \text{Impiegato:}m) \wedge \text{Impiegati}(\text{Matricola: } m, \text{Nome: } n, \text{Età: } e, \text{Stipendio: } s) \wedge s > 40 \}$

$\text{CapiDeiRicchi}(\text{Capo:}c) \leftarrow$
 $\text{ImpRicchi}(\text{Matricola: } m, \text{Nome: } n, \text{Età: } e, \text{Stipendio: } s),$
 $\text{Supervisione}(\text{Capo:}c, \text{Impiegato:}m)$

? $\text{CapiDeiRicchi}(\text{Capo:}c)$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

127

Esempio 5

- Trovare matricola e nome dei capi i cui impiegati guadagnano tutti più di 40 milioni.

• serve la negazione
 $\text{CapiDiNonRicchi}(\text{Capo:}c) \leftarrow$
 $\text{Supervisione}(\text{Capo:}c, \text{Impiegato:}m),$
 $\text{Impiegati}(\text{Matricola: } m, \text{Nome: } n, \text{Età: } e, \text{Stipendio: } s),$
 $s \leq 40$
 $\text{CapiSoloDiRicchi}(\text{Matricola: } c, \text{Nome: } n) \leftarrow$
 $\text{Impiegati}(\text{Matricola: } c, \text{Nome: } n, \text{Età: } e, \text{Stipendio: } s),$
 $\text{Supervisione}(\text{Capo:}c, \text{Impiegato:}m),$
 $\text{not CapiDiNonRicchi}(\text{Capo:}c)$

? $\text{CapiSoloDiRicchi}(\text{Matricola: } c, \text{Nome: } n)$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

128

Esempio 6

- Per ogni impiegato, trovare tutti i superiori.
- Serve la ricorsione

$\text{Superiore}(\text{Impiegato: } i, \text{SuperCapo: } c) \leftarrow$
 $\text{Supervisione}(\text{Impiegato: } i, \text{Capo: } c)$

$\text{Superiore}(\text{Impiegato: } i, \text{SuperCapo: } c) \leftarrow$
 $\text{Supervisione}(\text{Impiegato: } i, \text{Capo: } c'),$
 $\text{Superiore}(\text{Impiegato: } c', \text{SuperCapo: } c)$

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

129

Datalog, semantica

- La definizione della semantica delle regole ricorsive è delicata (in particolare con la negazione)
- Potere espressivo:
 - Datalog non ricorsivo senza negazione è equivalente al calcolo senza negazione e senza quantificatore universale
 - Datalog non ricorsivo con negazione è equivalente al calcolo e all'algebra
 - Datalog ricorsivo senza negazione e calcolo sono incomparabili
 - Datalog ricorsivo con negazione è più espressivo di calcolo e algebra

07/06/2005

Atzeni-Ceri-Paraboschi-Torlone,
Basi di dati, Capitolo 3

130