

**Atzeni, Ceri, Paraboschi, Torlone**  
**Basi di dati**

McGraw-Hill, 1996-2002

Capitolo 3:

**ALGEBRA E CALCOLO**  
**RELAZIONALE**

24/09/2002

**Linguaggi per basi di dati**

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

24/09/2002

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")

24/09/2002

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)

24/09/2002

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

4

**Algebra relazionale**

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

24/09/2002

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)

24/09/2002

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

24/09/2002

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

24/09/2002

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

24/09/2002

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

24/09/2002

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

??

24/09/2002

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

24/09/2002

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

12

**Paternità**

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

**REN<sub>Genitore → Padre</sub> (Paternità)**

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

24/09/2002 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 13

**Paternità**

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

**REN<sub>Genitore → Padre</sub> (Paternità)**

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

**Maternità**

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

**REN<sub>Genitore → Madre</sub> (Maternità)**

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

24/09/2002 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 14

**REN<sub>Genitore → Padre</sub> (Paternità)**

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

**REN<sub>Genitore → Padre</sub> (Paternità)**

E

**REN<sub>Genitore → Madre</sub> (Maternità)**

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

**REN<sub>Genitore → Madre</sub> (Maternità)**

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

24/09/2002 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<sub>Sede, Retribuzione → Ufficio, Stipendio</sub> (Impiegati)**

E

**REN<sub>Sede, Retribuzione → Fabbrica, Salario</sub> (Operai)**

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

24/09/2002 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

24/09/2002 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

24/09/2002 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 ennuola)
- semantica
  - il risultato contiene le ennuole dell'operando che soddisfano la condizione

24/09/2002

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

19

- impiegati che guadagnano più di 50

### Impiegati

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

**SEL**<sub>Stipendio > 50</sub> (Impiegati)

24/09/2002

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

20

- impiegati che guadagnano più di 50 e lavorano a Milano

### Impiegati

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

**SEL**<sub>Stipendio > 50 AND Filiale = 'Milano'</sub> (Impiegati)

24/09/2002

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

21

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

### Impiegati

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

**SEL**<sub>Cognome = Filiale</sub> (Impiegati)

24/09/2002

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

22

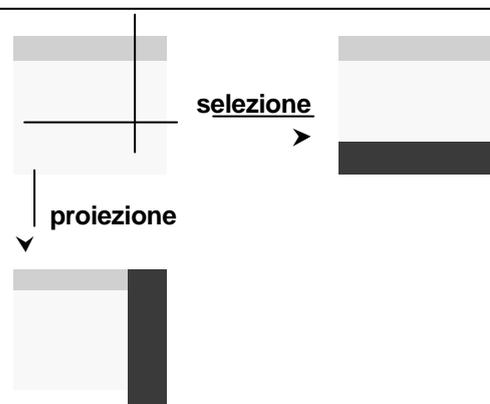
## Selezione e proiezione

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

24/09/2002

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

23



24/09/2002

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

24/09/2002

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

24/09/2002

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

24/09/2002

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)

24/09/2002

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)

24/09/2002

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  $R$

24/09/2002

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

30

## Selezione e proiezione

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

24/09/2002

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<sub>Matricola,Cognome</sub> (SEL<sub>Stipendio > 50</sub> (Impiegati))

24/09/2002

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

24/09/2002

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

24/09/2002

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

24/09/2002

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

24/09/2002

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

24/09/2002

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

24/09/2002

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  $X_1X_2$
- $$\{ t \text{ su } X_1X_2 \mid \text{esistono } t_1 \hat{\in} R_1 \text{ e } t_2 \hat{\in} R_2 \text{ con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$$

24/09/2002

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

39

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

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

- ogni ennupla contribuisce al risultato:
  - join completo

24/09/2002

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

40

## Un join non completo

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

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

24/09/2002

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

41

## Un join vuoto

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

Impiegato	Reparto	Capo

24/09/2002

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
Bianchi	B	Bruni

24/09/2002

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  $|R_1|$

24/09/2002

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

44

## Cardinalità del join, 2

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

$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1| \cdot |R_2|$$
- 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  $R_2$ :  

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

24/09/2002

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"

24/09/2002

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

24/09/2002

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 ...

24/09/2002

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<sub>LEFT</sub> Reparti**

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

24/09/2002 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<sub>RIGHT</sub> Reparti**

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

24/09/2002 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<sub>FULL</sub> Reparti**

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

24/09/2002 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

**Impiegato**

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

**Reparto**

Reparto	Capo
B	Mori

24/09/2002 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 52

**Proiezioni e join**

**Impiegato**

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

**Impiegato**

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

**Impiegato**

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

24/09/2002 Atzeni-Ceri-Paraboschi-Torlone, Basi di dati, Capitolo 3 53

**Join e proiezioni**

- $R_1(X_1), R_2(X_2)$
- $PROJ_{X_1}(R_1 JOIN R_2) \hat{=} R_1$
- $R(X), X = X_1 \hat{=} X_2$
- $(PROJ_{X_1}(R)) JOIN (PROJ_{X_2}(R)) \hat{=} R$

24/09/2002 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)

24/09/2002

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

55

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

### 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

24/09/2002

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

$R_1 JOIN_{Condizione} R_2$

24/09/2002

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  $A_1 J A_2$  dove J è uno degli operatori di confronto (=, >, <, ...)
- se l'operatore è sempre l'uguaglianza (=) allora si parla di equi-join

24/09/2002

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

58

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

### Impiegati JOIN<sub>Reparto=Codice</sub> Reparti

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

24/09/2002

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

59

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

### Impiegati JOIN Reparti

24/09/2002

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

60

## Join naturale ed equi-join

**Impiegati**                      **Reparti**  
**Impiegato Reparto**      **Reparto Capo**

Impiegati JOIN Reparti

**PROJ**<sub>Impiegato,Reparto,Capo</sub> ( **SEL**<sub>Reparto=Codice</sub>  
 ( **Impiegati JOINREN**<sub>Codice → Reparto</sub> ( **Reparti** ) )

24/09/2002

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

61

## 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

24/09/2002

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

62

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

**SEL**<sub>Stipendio>40</sub>(**Impiegati**)

24/09/2002

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

63

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

**SEL**<sub>Stipendio>40</sub>(**Impiegati**)

24/09/2002

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

64

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

**PROJ**<sub>Matricola, Nome, Età</sub>  
 ( **SEL**<sub>Stipendio>40</sub>( **Impiegati** ) )

24/09/2002

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

65

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

**PROJ**<sub>Matricola, Nome, Età</sub>  
 ( **SEL**<sub>Stipendio>40</sub>( **Impiegati** ) )

24/09/2002

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

66

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

```

PROJCapo (Supervisione
JOINImpiegato=Matricola
(SELStipendio>40(Impiegati)))

```

24/09/2002

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

67

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

```

PROJNome,Stipendio (
Impiegati JOINMatricola=Capo
PROJCapo(Supervisione
JOINImpiegato=Matricola(SELStipendio>40(Impiegati))))

```

24/09/2002

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

68

- 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)))

```

24/09/2002

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

69

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

```

PROJCapo (Supervisione) -
PROJCapo (Supervisione
JOINImpiegato=Matricola
(SELStipendio > 40(Impiegati)))

```

24/09/2002

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

70

### 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"

24/09/2002

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

71

### Un 'equivalenza importante

- Push selections (se A è attributo di R<sub>1</sub>)  
SEL<sub>A=10</sub> (R<sub>1</sub> JOIN R<sub>2</sub>) = R<sub>1</sub> JOIN SEL<sub>A=10</sub> (R<sub>2</sub>)
- Riduce in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione)

24/09/2002

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

72

## Selezione con valori nulli

### Impiegati

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

$SEL_{Età > 40}$  (Impiegati)

- la condizione atomica è vera solo per valori non nulli

24/09/2002

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

73

## Un risultato non desiderabile

$SEL_{Età > 30}$  (Persone) È  $SEL_{Età \neq 30}$  (Persone) <sup>1</sup>  
Persone

- Perché? Perché le selezioni vengono valutate separatamente!
- Ma anche

$SEL_{Età > 30 \cup Età \neq 30}$  (Persone) <sup>1</sup> Persone

- Perché? Perché anche le condizioni atomiche vengono valutate separatamente!

24/09/2002

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

74

## Selezione con valori nulli: soluzione

$SEL_{Età > 40}$  (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)

24/09/2002

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

75

- Quindi:

$SEL_{Età > 30}$  (Persone) È  $SEL_{Età \neq 30}$  (Persone) È  
 $SEL_{Età IS NULL}$  (Persone)

=

$SEL_{Età > 30 \cup Età \neq 30 \cup Età IS NULL}$  (Persone)

=

Persone

24/09/2002

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

76

### Impiegati

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

$SEL_{(Età > 40) OR (Età IS NULL)}$  (Impiegati)

24/09/2002

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

77

## Viste (relazioni derivate)

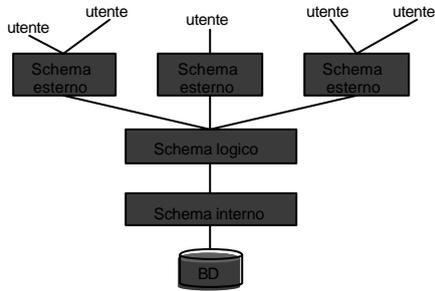
- Rappresentazioni diverse per gli stessi dati (**schema esterno**)
- Relazioni derivate:
  - relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- Relazioni di base: contenuto autonomo
- Le relazioni derivate possono essere definite su altre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

24/09/2002

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

78

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



24/09/2002

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

79

## Viste virtuali e materializzate

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

24/09/2002

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

80

## 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

24/09/2002

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

81

## Viste virtuali

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

24/09/2002

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

82

## Viste, esempio

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

- una vista:  
Supervisione =  
PROJ<sub>Impiegato, Capo</sub> (Afferenza JOIN Direzione)

24/09/2002

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

83

## Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

SEL<sub>Capo='Leoni'</sub> (Supervisione)  
viene eseguita come  
PROJ<sub>Impiegato, Capo</sub> (Afferenza JOIN Direzione)

24/09/2002

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

84

## 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 e' 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

24/09/2002

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

85

## 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))
```

24/09/2002

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

86

## 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?

24/09/2002

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

87

## 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

24/09/2002

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

88

## 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)

24/09/2002

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

89

- 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,B à C - Matr,Nome,Stip,Età (Impiegati)
JOINMatrC=Capo
(Supervisione JOINImpiegato=Matricola Impiegati)))
```

24/09/2002

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

90



### Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

```

PROJMatricola Nome, Età(Impiegati)
    { Matricola: m, Nome: n, Età: e |
$$ (Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s))
    { Matricola: m, Nome: n, Età: e |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s))
    
```

24/09/2002

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

97

### Esempio 1

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

```

PROJMatricola, Nome, Età(SELStipendio>40(Impiegati))
    { Matricola: m, Nome: n, Età: e |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)
     $\bar{\cup}$  s > 40 }
    
```

24/09/2002

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

98

### Esempio 2

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

```

PROJCapo(Supervisione JOINImpiegato=Matricola
(SELStipendio>40(Impiegati)))
    { Capo: c | Supervisione(Capo:c, Impiegato:m)  $\bar{\cup}$ 
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)
     $\bar{\cup}$  s > 40 }
    
```

24/09/2002

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

99

### Esempio 3

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

```

PROJNomeC, StipC
(RENMatrC, NomeC, StipC, EtàC - Matr, Nome, Stip, Età
(JOINMatrC=Capo
(Supervisione JOINImpiegato=Matricola
(SELStipendio>40(Impiegati))))
    { NomeC: nc, StipC: sc |
Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)  $\bar{\cup}$ 
s > 40  $\bar{\cup}$  Supervisione(Capo:c, Impiegato:m)  $\bar{\cup}$ 
Impiegati(Matricola:c, Nome:nc, Età:ec, Stipendio: sc) }
    
```

24/09/2002

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

100

### 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
(SELStipendio>StipC (RENMatrC, NomeC, StipC, EtàC -
Matr, Nome, Stip, Età
(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)  $\bar{\cup}$ 
Supervisione(Capo:c, Impiegato:m)  $\bar{\cup}$ 
Impiegati(Matricola: c, Nome: nc, Età: ec, Stipendio: sc)  $\bar{\cup}$  s > sc }
    
```

24/09/2002

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

101

### 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
(SELStipendio > 40(Impiegati))))))
    {Matricola: c, Nome: n |
Impiegati(Matricola: c, Nome: n, Età: e, Stipendio: s)  $\bar{\cup}$ 
Supervisione(Capo:c, Impiegato:m)  $\bar{\cup}$ 
- S m' (S n' (S e' (S s' (Impiegati (Matr: m', Nome: n', Età: e', Stip: s')  $\bar{\cup}$ 
Supervisione(Capo:c, Impiegato:m')  $\bar{\cup}$  s' > 40))))))
    
```

24/09/2002

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

102

## Quantificatori esistenziali o universali?

- Sono intercambiabili, per le leggi di De Morgan:

$$\{ \text{Matricola: c, Nome: n} \mid \text{Impiegati}(\text{Matricola: c, Nome: n, Et\`a: e, Stipendio: s}) \dot{\cup} \text{Supervisione}(\text{Capo: c, Impiegato: m}) \dot{\cup} \neg \text{Sm}'(\text{S}'(\text{S}'(\text{S}'(\text{Impiegati}(\text{Matr: m}', \text{Nome: n}', \text{Et\`a: e}', \text{Stip: s}') \dot{\cup} \text{Supervisione}(\text{Capo: c, Impiegato: m}') \dot{\cup} \text{s}' > 40)))))) \}$$

$$\{ \text{Matricola: c, Nome: n} \mid \text{Impiegati}(\text{Matricola: c, Nome: n, Et\`a: e, Stipendio: s}) \dot{\cup} \text{Supervisione}(\text{Capo: c, Impiegato: m}) \dot{\cup} \text{m}'(\text{ " n}'(\text{ " e}'(\text{ " s}'(\neg(\text{Impiegati}(\text{Matr: m}', \text{Nome: n}', \text{Et\`a: e}', \text{Stip: s}') \dot{\cup} \text{Supervisione}(\text{Capo: c, Impiegato: m}') \dot{\cup} \text{s}' > 40)))))) \}$$

24/09/2002

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

103

## Calcolo su domini, discussione

- Pregi:
  - dichiarativit\`a
- Difetti:
  - "verbosit\`a": tante variabili!
  - espressioni senza senso:
 
$$\{ A: x \mid \neg R(A: x) \}$$

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

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

24/09/2002

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

104

## 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)

24/09/2002

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

105

## 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

24/09/2002

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

106

## Calcolo su ennuple con dichiarazioni di range, sintassi

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

$$\text{Sx}(\text{R})(\dots) \quad \text{ " } \text{x}(\text{R})(\dots)$$

24/09/2002

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

107

## Esempio 0a

- Trovare matricola, nome, et\`a e stipendio degli impiegati che guadagnano pi\`u di 40 milioni

$$\text{SEL}_{\text{Stipendio} > 40}(\text{Impiegati})$$

$$\{ \text{Matricola: m, Nome: n, Et\`a: e, Stipendio: s} \mid \text{Impiegati}(\text{Matricola: m, Nome: n, Et\`a: e, Stipendio: s}) \dot{\cup} \text{s} > 40 \}$$

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

24/09/2002

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

108

### Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

$PROJ_{Matricola, Nome, Età}(Impiegati)$   
 $\{ Matricola: m, Nome: n, Età: e |$   
 $Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)\}$   
 $\{ i.(Matricola, Nome, Età) | i(Impiegati) | \}$

24/09/2002

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

109

### Esempio 1

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

$PROJ_{Matricola, Nome, Età}(SEL_{Stipendio > 40}(Impiegati))$   
 $\{ Matricola: m, Nome: n, Età: e |$   
 $Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s)$   
 $\quad \cup \quad s > 40 \}$   
 $\{ i.(Matricola, Nome, Età) | i(Impiegati) | i.Stipendio > 40 \}$

24/09/2002

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

110

### Esempio 2

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

$\{ Capo: c | Supervisione(Capo:c, Impiegato:m) \cup$   
 $Impiegati(Matricola: m, Nome: n, Età: e,$   
 $Stipendio: s) \cup s > 40 \}$   
 $\{ s.Capo | i(Impiegati), s(Supervisione) |$   
 $i.Matricola=s.Impiegato \cup i.Stipendio > 40 \}$

24/09/2002

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

111

### Esempio 3

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

$\{ NomeC: nc, StipC: sc |$   
 $Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s) \cup$   
 $\quad s > 40 \cup$   
 $Supervisione(Capo:c, Impiegato:m) \cup$   
 $Impiegati(Matricola:c, Nome:nc, Età:ec, Stipendio:sc) \}$   
 $\{ NomeC, StipC: i'.(Nome, Stip) |$   
 $i'(Impiegati), s(Supervisione), i(Impiegati) |$   
 $i'.Matricola=s.Capo \cup i.Matricola=s.Impiegato \cup$   
 $i.Stipendio > 40 \}$

24/09/2002

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

112

### Esempio 4

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

$\{ Matr: m, Nome: n, Stip: s, NomeC: nc, StipC: sc |$   
 $Impiegati(Matricola: m, Nome: n, Età: e, Stipendio: s) \cup$   
 $Supervisione(Capo:c, Impiegato:m) \cup$   
 $Impiegati(Matricola: c, Nome: nc, Età: ec, Stipendio: sc) \cup$   
 $\quad s > sc \}$   
 $\{ i.(Nome, Matr, Stip), NomeC, MatrC, StipC: i'.(Nome, Matr, Stip) |$   
 $i'(Impiegati), s(Supervisione), i(Impiegati) |$   
 $i'.Matricola=s.Capo \cup i.Matricola=s.Impiegato \cup i.Stipendio >$   
 $i'.Stipendio \}$

24/09/2002

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

113

### Esempio 5

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

$\{ Matricola: c, Nome: n |$   
 $Impiegati(Matricola: c, Nome: n, Età: e, Stipendio: s) \cup$   
 $Supervisione(Capo:c, Impiegato:m) \cup$   
 $\neg \exists m'(\exists n'(\exists e'(\exists s'(Impiegati(Matricola: m', Nome: n', Età: e', Stip: s') \cup$   
 $Supervisione(Capo:c, Impiegato:m') \cup s' \leq 40))) \}$   
 $\{ i.(Matricola, Nome) | s(Supervisione), i(Impiegati) |$   
 $i.Matricola=s.Capo \cup \neg(\exists i'(Impiegati)(\exists s'(Supervisione)$   
 $(s.Capo=s'.Capo \cup s'.Impiegato=i'.Matricola \cup i'.Stipendio \leq 40)))) \}$

24/09/2002

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

114

## 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

24/09/2002

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

115

## 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

24/09/2002

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

116

## Chiusura transitiva

### Supervisione(Impiegato, Capo)

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

Impiegato	Capo	Impiegato	Superiore
Rossi	Lupi	Rossi	Lupi
Neri	Bruni	Neri	Bruni
Lupi	Falchi	Lupi	Falchi
		Rossi	Falchi

24/09/2002

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

117

## Chiusura transitiva, come si fa?

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

- Ma:

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

24/09/2002

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

118

## 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!

24/09/2002

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

119

## 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

24/09/2002

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

120

## Datalog, sintassi

- Regole:

$testa \neg corpo$

- *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 "?")

24/09/2002

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

121

## 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)

24/09/2002

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

122

## 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)

24/09/2002

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

123

## Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

PROJ<sub>Matricola, Nome, Età</sub>(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) , s > 40

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

24/09/2002

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

124

## Esempio 2

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

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

CapiDeiRicchi (Capo:c) ∪

ImpRicchi(Matricola: m, Nome: n, Età: e, Stipendio: s),  
Supervisione (Capo:c, Impiegato:m)

? CapiDeiRicchi (Capo:c)

24/09/2002

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

125

## Esempio 5

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

- serve la negazione

CapiDiNonRicchi (Capo:c) ∪  
Supervisione (Capo:c, Impiegato:m),  
Impiegati (Matricola: m, Nome: n, Età: e, Stipendio: s) ,  
s ≤ 40

CapiSoloDiRicchi (Matricola: c, Nome: n) ∪  
Impiegati (Matricola: c, Nome: n, Età: e, Stipendio: s) ,  
Supervisione (Capo:c, Impiegato:m),  
not CapiDiNonRicchi (Capo:c)

? CapiSoloDiRicchi (Matricola: c, Nome: n)

24/09/2002

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

126

### Esempio 6

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

Superiore (Impiegato: i, SuperCapo: c)  $\rightarrow$   
Supervisione (Impiegato: i, Capo: c)

Superiore (Impiegato: i, SuperCapo: c)  $\rightarrow$   
Supervisione (Impiegato: i, Capo: c'),  
Superiore (Impiegato: c', SuperCapo: c)

24/09/2002

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

127

### 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

24/09/2002

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

128