

E. Giunchiglia Basi di dati 1

(trasparenze basate su Atzeni, Ceri, Paraboschi, Torlone: Basi di dati,
Capitolo 3)

ALGEBRA RELAZIONALE CALCOLO RELAZIONALE DATALOG

4/10/2004

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")

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

2

Linguaggi di interrogazione

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

3

Algebra relazionale

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

4

Operatori dell'algebra relazionale

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

5

Operatori insiemistici

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

6

Unione

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

Laureati \cup Quadri

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

7

Intersezione

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

Laureati \cap Quadri

Matricola	Nome	Età
7432	Neri	54
9824	Verdi	45

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

8

Differenza

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

Laureati - Quadri

Matricola	Nome	Età
7274	Rossi	42

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

9

Un'unione sensata ma impossibile

Paternità		Maternità	
Padre	Figlio	Madre	Figlio
Adamo	Abele	Eva	Abele
Adamo	Caino	Eva	Set
Abramo	Isacco	Sara	Isacco

Paternità \cup Maternità
??

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

10

Ridenominazione

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

11

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

REN_{Genitore} ← Padre (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

12

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 13

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 14

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 15

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 16

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 17

- impiegati che guadagnano più di 50

Impiegati

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

SEL_{Stipendio > 50} (Impiegati)

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 18

- impiegati che guadagnano più di 50 e lavorano a Milano

Impiegati

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

19

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

Impiegati

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

SEL_{Cognome = Filiale}(Impiegati)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

20

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

21

Condizione

Le condizioni possono essere *atomiche* o *composte*:

- **Atomiche:** $X \theta V$, dove
 - X è il nome di un attributo
 - θ è uno fra ($<$, $>$, $=$, \leq , \geq , \neq)
 - V è un valore del dominio di X , oppure il nome di un attributo con dominio compatibile con quello di X
- **Composte:** ottenute come combinazione Booleana delle condizioni atomiche (i.e., tramite gli operatori \wedge , \vee , \neg).

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

22

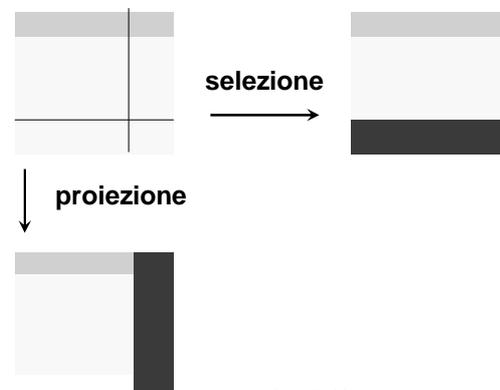
Selezione e proiezione

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

23



4/10/2004

Basi di Dati 1: AR, CR, DATALOG

24

Proiezione

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

26

- matricola e cognome di tutti gli impiegati

Matricola	Cognome	
7309	Neri	
5998	Neri	
9553	Rossi	
5698	Rossi	

PROJ_{Matricola, Cognome} (Impiegati)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

27

- cognome e filiale di tutti gli impiegati

	Cognome	Filiale	
	Neri	Napoli	
	Neri	Milano	
	Rossi	Roma	

PROJ_{Cognome, Filiale} (Impiegati)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

28

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

29

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

30

Selezione e proiezione

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

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))

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

32

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

33

Join

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

35

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

36

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

37

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

38

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

39

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

40

Un join vuoto

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

Impiegato	Reparto	Capo

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

41

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

42

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 superchiave 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 $|R_1|$

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

43

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 :

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

44

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"

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

45

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

46

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 ...

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

47

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

Impiegati JOIN_{LEFT} Reparti

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

48

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

Impiegati JOIN _{RIGHT} Reparti		
Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 49

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

Impiegati JOIN _{FULL} Reparti		
Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 50

Join e proiezioni

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

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Impiegato	Reparto	Reparto	Capo
Neri	B	B	Mori
Bianchi	B	B	Mori

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 51

Proiezioni e join

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

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

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 52

Join e proiezioni

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

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

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

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 53

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)

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 54

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 55

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

$$SEL_{Condizione} (R_1 \text{ JOIN } R_2)$$

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

$$R_1 \text{ JOIN}_{Condizione} R_2$$

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 56

Perché "theta-join"?

- La condizione C è spesso una congiunzione (AND) di atomi di confronto $A_1 \theta A_2$ dove θ è uno degli operatori di confronto (=, >, <, ...)
- se l'operatore è sempre l'uguaglianza (=) allora si parla di equi-join

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 57

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

Impiegati JOIN_{Reparto=Codice} Reparti

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

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 58

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

Impiegati JOIN Reparti

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 59

Join naturale ed equi-join

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

Impiegati JOIN Reparti

$$PROJ_{Impiegato,Reparto,Capo} (SEL_{Reparto=Codice} (Impiegati \text{ JOIN } REN_{Codice \leftarrow Reparto} (Reparti)))$$

4/10/2004 Basi di Dati 1: AR, CR, DATALOG 60

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

61

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

SEL_{Stipendio>40}(Impiegati)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

62

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

SEL_{Stipendio>40}(Impiegati)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

63

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

64

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

65

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

66

- 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))))

```

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

67

- 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)
  JOINMatr=Capo
  (Supervisione JOINImpiegato=Matricola Impiegati)))

```

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

68

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

```

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

```

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

69

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"

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

70

Equivalenze importanti

- Push selections (se A non è attributo di R_1)
 $SEL_{A=10}(R_1 JOIN R_2) = R_1 JOIN SEL_{A=10}(R_2)$
 Riduce in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione)
- Commutatività del Join
 $R_1 JOIN R_2 = R_2 JOIN R_1$
- Associatività del Join
 $(R_1 JOIN R_2) JOIN R_3 = R_1 JOIN (R_2 JOIN R_3)$

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

71

Selezione con valori nulli

Impiegati

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

```
SELEtà > 40 (Impiegati)
```

- la condizione atomica è vera solo per valori non nulli

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

72

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!

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

73

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

74

- Quindi:

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

=

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

=

Persone

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

75

Impiegati

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

76

Viste (relazioni derivate)

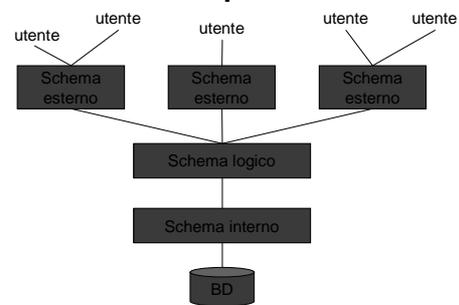
- Rappresentazioni diverse per gli stessi dati (schema esterno)
- Relazioni derivate:
 - relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- Relazioni di base: contenuto autonomo
- Le relazioni derivate possono essere definite su altre derivate,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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

77

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



4/10/2004

Basi di Dati 1: AR, CR, DATALOG

78

Viste virtuali e materializzate

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

79

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

80

Viste virtuali

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

81

Viste, esempio

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

82

Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

SEL_{Capo='Leoni'} (Supervisione)
 viene eseguita come
 PROJ_{Impiegato, Capo} (Afferenza JOIN Direzione)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

83

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

84

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))
```

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

85

Viste e aggiornamenti, attenzione

Afferenza	Reparto	Direzione	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?

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

86

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

87

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)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

88

- 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)))
```

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

89

```
PROJ_Matr, Nome, Stip, MatrC, NomeC, StipC
(SEL_Stip>StipC(
REN_MatrC, NomeC, StipC, EtàC ← Matr, Nome, Stip, Età (Imp)
JOIN_MatrC=Capo
(Sup JOIN_Imp=Matr Imp)))

Capi := Imp

PROJ_Imp.Matr, Imp.Nome, Imp.Stip, Capi.Matr, Capi.Nome, Capi.Stip
(SEL_Imp.Stip>Capi.Stip(
Capi JOIN_Capi.Matr=Capo (Sup JOIN_Imp=Imp.Matr Imp)))
```

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

90

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

91

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

92

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

93

Base di dati per gli esempi

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

94

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) \wedge s > 40 }

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

95

Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

PROJ_{Matricola, Nome, Età}(Impiegati)

{ Matricola: m, Nome: n, Età: e |
 \exists 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)}

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

96

Esempio 1

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

97

Esempio 2

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

98

Esempio 3

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

$(REN_{MatrC, NomeC, StipC, Et\grave{a}C \leftarrow Matr, Nome, Stip, Et\grave{a}}(Impiegati))$
 $JOIN_{MatrC=Capo}$
 $(Supervisione JOIN_{Impiegato=Matricola}(SEL_{Stipendio > 40}(Impiegati)))$
 $\{ NomeC: nc, StipC: sc \mid$
 $Impiegati(Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s) \wedge$
 $s > 40 \wedge Supervisione(Capo:c, Impiegato:m) \wedge$
 $Impiegati(Matricola:c, Nome:nc, Et\grave{a}:ec, Stipendio: sc) \}$

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

99

Esempio 4

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

$PROJ_{Matr, Nome, Stip, MatrC, NomeC, StipC}$
 $(SEL_{Stipendio > StipC}(REN_{MatrC, NomeC, StipC, Et\grave{a}C \leftarrow$
 $Matr, Nome, Stip, Et\grave{a}}(Impiegati))$
 $JOIN_{MatrC=Capo}$
 $(Supervisione JOIN_{Impiegato=Matricola}((Impiegati))))$
 $\{ Matr: m, Nome: n, Stip: s, MatrC: c, NomeC: nc, StipC: sc \mid$
 $Impiegati(Matricola: m, Nome: n, Et\grave{a}: e, Stipendio: s) \wedge$
 $Supervisione(Capo:c, Impiegato:m) \wedge$
 $Impiegati(Matricola: c, Nome: nc, Et\grave{a}: ec, Stipendio: sc) \wedge s > sc \}$

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

100

Esempio 5

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

$PROJ_{Matricola, Nome}(Impiegati JOIN_{Matricola=Capo}(PROJ_{Capo}(Supervisione) -$
 $PROJ_{Capo}(Supervisione JOIN_{Impiegato=Matricola}(SEL_{Stipendio \le 40}(Impiegati))))$
 $\{ Matricola: c, Nome: n \mid$
 $Impiegati(Matricola: c, Nome: n, Et\grave{a}: e, Stipendio: s) \wedge$
 $Supervisione(Capo:c, Impiegato:m) \wedge$
 $\neg \exists m'(\exists n'(\exists e'(\exists s'(Impiegati(Matr: m', Nome: n', Et\grave{a}: e', Stip: s') \wedge$
 $Supervisione(Capo:c, Impiegato:m') \wedge s' \le 40))))$

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

101

Quantificatori esistenziali o universali?

- Sono intercambiabili, per le leggi di De Morgan:

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

102

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

103

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)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

104

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

105

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 $x.A \neq c$, $x.A \neq y.B$
 - connettivi
 - quantificatori che associano un range alle variabili

$\exists x(R)(...)$ $\forall x(R)(...)$

Basi di Dati 1: AR, CR, DATALOG

106

Esempio 0a

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

$SEL_{Stipendio > 40}(Impiegati)$

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

107

Esempio 0b

- Trovare matricola, nome ed età di tutti gli impiegati

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

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

108

Esempio 1

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

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

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

109

Esempio 2

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

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

110

Esempio 3

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

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

111

Esempio 4

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

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

112

Esempio 5

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

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

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

113

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

114

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

115

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

116

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	Falchi	Leoni
		Rossi	Falchi
		Lupi	Leoni
		Rossi	Leoni

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

117

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!

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

118

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

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

119

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 "?")

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

120

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)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

121

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)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

122

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)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

123

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)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

124

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)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

125

Esempio 6

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

Superiore (Impiegato: i, SuperCapo: c) ←
Supervisione (Impiegato: i, Capo: c)

Superiore (Impiegato: i, SuperCapo: c) ←
Superiore (Impiegato: i, Capo: c'),
Superiore (Impiegato: c', SuperCapo: c)

4/10/2004

Basi di Dati 1: AR, CR, DATALOG

126

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