

# Algebra Relazionale e Calcolo Relazionale

L. Vigliano

# Operazioni associate al modello relazionale

- Notazione algebrica  $\Rightarrow$  **Algebra relazionale**  
 $\Rightarrow$  Linguaggio procedurale  $\Rightarrow$   
interrogazioni espresse applicando operatori  
alle relazioni
- Notazione logica  $\Rightarrow$  **Calcolo relazionale**  $\Rightarrow$   
Linguaggio dichiarativo  $\Rightarrow$  interrogazioni  
espresse tramite formule logiche le cui  
risposte devono essere rese vere dalle tuple

# Algebra Relazionale

L. Vigliano

# Algebra relazionale e Calcolo relazionale

- Entrambe le famiglie di linguaggi sono equivalenti in senso espressivo

# Algebra relazionale

- L' **Algebra relazionale** è un linguaggio *procedurale*, le cui operazioni quindi vengono specificate descrivendo il procedimento da seguire per ottenere la soluzione.

# Algebra relazionale

- L' algebra relazionale di base ha solo 5 operazioni :
  - Unione, Differenza, Prodotto Cartesiano, Selezione e Proiezione
- Ulteriori operazioni che descriveremo sono derivabili dalle 5 operazioni di base

# Unione , Intersezione , Differenza

- **Unione**  $R_1 \cup R_2$  di due relazioni  $R_1$  e  $R_2$  è una relazione contenente le tuple che appartengono o a  $R_1$  o a  $R_2$
- **Differenza**  $R_1 - R_2$  è una relazione che contiene le tuple che appartengono a  $R_1$  e non appartengono a  $R_2$
- Dato che  $R \cap S = R - (R - S)$  possiamo dire che l'**Intersezione**  $R \cap S$  è una relazione che contiene le tuple che appartengono sia a  $R$  sia a  $S$

# Unione, Intersezione, Differenza

## Laureati

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38

## Dirigenti

Matricola	Cognome	Età
9297	Neri	56
7432	Neri	39
9824	Verdi	38

**Laureati**

∪

**Dirigenti**

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38
9297	Neri	56



# Unione, Intersezione, Differenza

## Laureati

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38

## Dirigenti

Matricola	Cognome	Età
9297	Neri	56
7432	Neri	39
9824	Verdi	38

## **Laureati**

$\cap$

## **Dirigenti**

Matricola	Cognome	Età
7432	Neri	39
9824	Verdi	38

## **Laureati - Dirigenti**

Matricola	Cognome	Età
7274	Rossi	37

# Ridenominazione

- L' operatore di **Ridenominazione** cambia il nome degli attributi allo scopo di facilitare operazioni insiemistiche
- La ridenominazione  $\rho_{B_1, B_2, \dots, B_k \leftarrow A_1, A_2, \dots, A_k}(R)$  contiene tuple  $t'$  tali che  $t'$  è una tupla e  $t'[B_i] = t[A_i]$ ,  
cioè cambiano i nomi degli attributi ma i valori non cambiano

# Ridenominazione

## Esempio di utilizzo

### Paternità

Padre	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco

### Maternità

Madre	Figlio
Eva	Caino
Eva	Abele
Sara	Isacco

$$\rho_{\text{Genitore} \leftarrow \text{Padre}}(\text{Paternità}) \cup \rho_{\text{Genitore} \leftarrow \text{Madre}}(\text{Maternità})$$

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

# Selezione

- L' operatore di **Selezione** produce un sottoinsieme delle tuple, su tutti gli attributi.

"Decomposizione orizzontale"

- La selezione  $\sigma_F(R)$  produce una relazione sugli stessi attributi di R che contiene le tuple di R su cui la formula F è vera.

# Selezione

## Impiegati

Cognome	Nome	Età	Stipendio
Rossi	Mario	25	1000
Neri	Luca	40	1500
Verdi	Nico	36	2400
Rossi	Marco	40	1900

$\sigma_{\text{Età} > 30 \wedge \text{Stipendio} > 2000}(\text{Impiegati})$

Cognome	Nome	Età	Stipendio
Verdi	Nico	36	2400

# Selezione

## Cittadini

Cognome	Nome	Nascita	Residenza
Rossi	Mario	Roma	Milano
Neri	Luca	Roma	Roma
Verdi	Nico	Firenze	Firenze
Rossi	Marco	Napoli	Firenze

$\sigma_{\text{Nascita=Residenza}}$ (Cittadini)

Cognome	Nome	Nascita	Residenza
Neri	Luca	Roma	Roma
Verdi	Nico	Firenze	Firenze

# Proiezione

- L' operatore di **Proiezione** produce un risultato cui contribuiscono tutte le tuple, ma su un sottoinsieme degli attributi.

## "Decomposizione verticale"

- La proiezione  $\pi_Y(R)$  è l' insieme di tuple su un sottoinsieme Y di attributi X di R, ottenuta dalle tuple di R considerando solo i valori su Y cioè :

$$\pi_Y(R) = \{t[Y] \text{ tale che } t \in R\}$$

# Proiezione

## Impiegati

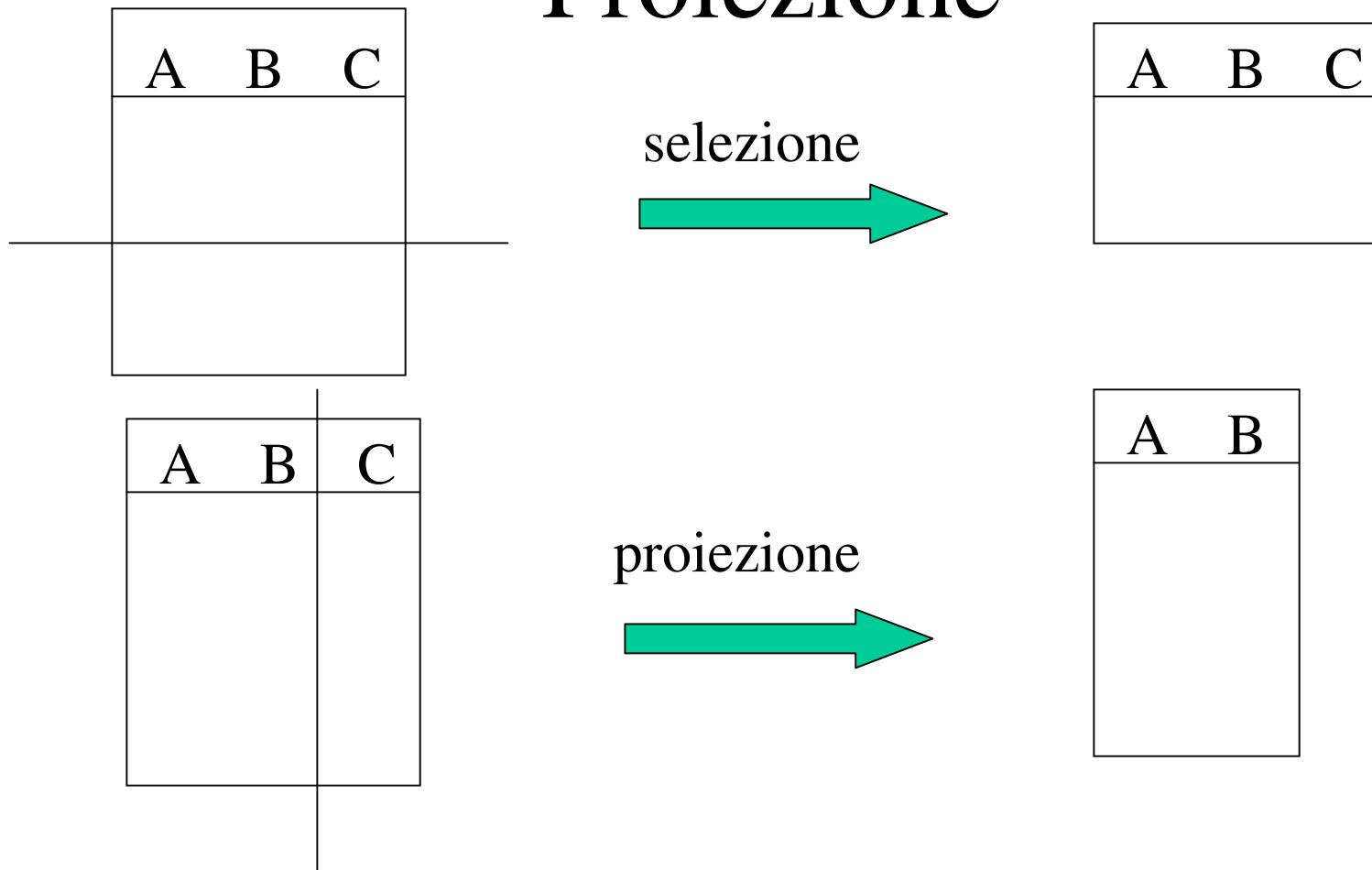
Cognome	Nome	Reparto	Capo
Rossi	Mario	Vendite	Gatti
Neri	Luca	Vendite	Gatti
Verdi	Mario	Personale	Lupi
Rossi	Marco	Personale	Lupi

$\pi_{\text{Cognome, Nome}}(\text{Impiegati})$

Cognome	Nome
Rossi	Mario
Neri	Luca
Verdi	Mario
Rossi	Marco



# Differenza tra Selezione e Proiezione



# Join

- L' operatore d**Join** permette di correlare dati contenuti in relazioni diverse, confrontandone i valori, e utilizzando una delle caratteristiche fondamentali del modello, quella di essere basata sui valori.
- Due tipi di join : **Join naturale** e **Theta join**

# Join naturale

- L' operatore di **Join naturale**  $R_1 \bowtie R_2$  (o  $R_1 \text{ join } R_2$ ) correla dati in relazioni diverse sulla base di valori uguali in attributi con lo stesso nome.
- Definito come :
$$R_1 \text{ join } R_2 = \{t \text{ su } X_1 X_2 \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2 \text{ con } t[X_1] = t_1 \text{ e } t[X_2] = t_2\}$$

# Join naturale

**R<sub>1</sub>**

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

**R<sub>2</sub>**

Reparto	Capo
produzione	Mori
vendite	Bruni

**R<sub>1</sub>  $\bowtie$  R<sub>2</sub>**

Impiegato	Reparto	Capo
Rossi	vendite	Bruni
Neri	produzione	Mori
Bianchi	produzione	Mori

# Join naturale

## Esempi di Join non completi

R<sub>1</sub>

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

R<sub>2</sub>

Reparto	Capo
produzione	Mori
acquisti	Bruni

*Tupla dangling*

R<sub>1</sub> ⋈ R<sub>2</sub>

Impiegato	Reparto	Capo
Neri	produzione	Mori
Bianchi	produzione	Mori

# Join naturale

## Esempi di Join combinabili con tutto

**R**<sub>1</sub>

Impiegato	Progetto
Rossi	A
Neri	A
Bianchi	A

**R**<sub>2</sub>

Progetto	Capo
A	Mori
A	Bruni

**R**<sub>1</sub>  $\bowtie$  **R**<sub>2</sub>

Impiegato	Progetto	Capo
Rossi	A	Mori
Neri	A	Mori
Bianchi	A	Mori
Rossi	A	Bruni
Neri	A	Bruni
Bianchi	A	Bruni

# Join esterno (outer Join)

## variante del Join naturale

- L' operatore di **Join esterno** prevede che tutte le tuple diano sempre un contributo al risultato, eventualmente estese con valori nulli ove non vi siano controparti opportune.
- Tre tipi di outer join :  
**left join   -   right join   -   full join**

# Join esterno

## Left Join

**R<sub>1</sub>**

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

**R<sub>2</sub>**

Reparto	Capo
produzione	Mori
acquisti	Bruni

**R<sub>1</sub> |X| LEFT R<sub>2</sub>**

Impiegato	Reparto	Capo
Rossi	vendite	NULL
Neri	produzione	Mori
Bianchi	produzione	Mori



# Join esterno

## Right Join

**R<sub>1</sub>**

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

**R<sub>2</sub>**

Reparto	Capo
produzione	Mori
acquisti	Bruni

**R<sub>1</sub> |X| RIGHT R<sub>2</sub>**

Impiegato	Reparto	Capo
Neri	produzione	Mori
Bianchi	produzione	Mori
NULL	acquisti	Bruni

# Join esterno

## Full Join

**R<sub>1</sub>**

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

**R<sub>2</sub>**

Reparto	Capo
produzione	Mori
acquisti	Bruni

**R<sub>1</sub> |X| FULL R<sub>2</sub>**

Impiegato	Reparto	Capo
Rossi	vendite	NULL
Neri	produzione	Mori
Bianchi	produzione	Mori
NULL	acquisti	Bruni

# Theta Join

- L' operatore di **Theta Join**  $R_1 \bowtie_F R_2$  è spesso definito come un prodotto cartesiano tra due relazioni  $R_1$  ed  $R_2$  seguito da una selezione che individua le tuple correlate secondo le esigenze.
- Definito come :

$$R_1 \bowtie_F R_2 = \sigma_F(R_1 \times R_2)$$

dove  $F$  è una formula e  $R_1$  e  $R_2$  non hanno attributi di nome comune

# Theta Join (Equi Join)

R<sub>1</sub>

Impiegato	Progetto
Rossi	A
Neri	A
Neri	B

R<sub>2</sub>

Codice	Nome
A	Venere
B	Marte

$R_1 \bowtie_{\text{Progetto=Codice}} R_2$

Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	A	Venere
Neri	B	B	Marte

# Selezione, Proiezione, Join

- La possibilità, data dai precedenti operatori (soprattutto dal join), di eseguire cammini diversi tra le relazioni (la navigazione) è il punto di forza di questo modello.

# Interrogazioni in algebra relazionale

- Un' **interrogazione** può essere definita come una funzione che, applicata a istanze di basi di dati, produce relazioni
- (le interrogazioni sono in pratica espressioni di relazioni che producono relazioni)

# Esempi di interrogazioni in Algebra relazionale

- Consideriamo la seguente Base di dati :
- Relazioni
  - IMPIEGATI(Matr, Nome, Età, Stipendio)
  - SUPERVISIONE(Capo, Impiegato)
- La cui rappresentazione tabellare potrebbe essere :

# Esempi di interrogazioni in Algebra relazionale

Impiegati

Matr	Nome	Età	Stip
101	Rossi	34	40
103	Bianchi	23	35
104	Neri	38	61
210	Celli	49	60
231	Bisi	50	60
252	Bini	44	70
301	S. Rossi	34	70
375	M. Rossi	50	65

Supervisione

Capo	Impiegato
210	101
210	103
210	104
301	210
301	231
375	252



# Esempi di interrogazioni in Algebra relazionale

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

$$\pi_{\text{Matr, Nome, Et\`a}}(\sigma_{\text{Stip} > 40}(\text{IMPIEGATI}))$$

Matr	Nome	Età
104	Neri	38
210	Celli	49
231	Bisi	50
352	Bini	44
301	S. Rossi	34
375	M. Rossi	50

# Esempi di interrogazioni in Algebra relazionale

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

$\pi_{\text{Capo}, (\text{SUPERVISIONE} | X |_{\text{Impiegato}=\text{Matr}} (\sigma_{\text{Stip} > 40} (\text{IMPIEGATI})))$

Capo
210
301
375

# Esempi di interrogazioni in Algebra relazionale

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

$\pi_{\text{NomeC}, \text{StipC}}(\rho_{\text{MatrC}, \text{NomeC}, \text{StipC}, \text{EtàC} \leftarrow \text{Matr}, \text{Nome}, \text{Stip}, \text{Età}}(\text{IMPIEGATI}) \mid X \mid_{\text{MatrC}=\text{Capo}}(\text{SUPERVISIONE} \mid X \mid_{\text{Impiegato}=\text{Matr}}(\sigma_{\text{Stip} > 40}(\text{IMPIEGATI}))))$

NomeC	StipC
Celli	60
S. Rossi	70
M. Rossi	65

# Infine .....

- Si possono formulare espressioni equivalenti fra loro che semplificano la notazione o permettono di ridurre le dimensioni dei risultati intermedi
- Per i valori nulli (e per evitare ambiguità nelle risposte) si introducono due nuove forme di condizione di selezione :
  - **A is null** assume valore vero se il valore della tupla su A è nullo, falso altrimenti
  - **A is not null** viceversa

# Calcolo Relazionale

L. Vigliano

# Calcolo relazionale

- Il **Calcolo relazionale** è un linguaggio *dichiarativo*, in cui le espressioni descrivono le proprietà del risultato (vero-falso), piuttosto che la procedura per ottenerlo.
- E' basato sul calcolo dei predicati del primo ordine

# Calcolo relazionale

- La forma di logica chiamata **Calcolo relazionale** è alla base di quasi tutti i linguaggi di interrogazione esistenti e basati sul modello relazionale.

# Calcolo relazionale

- Esistono diverse versioni del calcolo relazionale, ne vedremo solo due :
  - 1) **Calcolo relazionale sui Domini**
    - Quello più vicino al calcolo dei predicati
  - 2) **Calcolo relazionale sulle Tuple con dichiarazioni di range**
    - Variazione del precedente
    - Base di molti costrutti degli attuali linguaggi



# Calcolo relazionale sui Domini

- Le espressioni hanno la forma :

$$\{A_1 : x_1, \dots, A_k : x_k \mid f\}$$

Dove

$A_1 : x_1, \dots, A_k : x_k$  target list o lista degli obiettivi

$A_1, \dots, A_k$  sono attributi (rispetto a cui viene formulata l' interrogazione)

$x_1, \dots, x_k$  sono variabili

$f$  è una formula del tipo :

$R(A_1 : x_1, \dots, A_p : x_p)$  con  $R(A_1, \dots, A_p)$  schema di relazione e  $x_1, \dots, x_p$  variabili

$x\theta y$  o  $x\theta c$  con  $x$  e  $y$  variabili e  $c$  costante  $\theta$  operatore di confronto ( $=, <, >, \neq, \geq, \leq$ )

# Calcolo relazionale sui Domini

## (2)

Oppure  $f$  è del tipo:

se  $f_1$  e  $f_2$  sono formule lo sono anche  $f_1 \vee f_2$ ,  $f_1 \wedge f_2$ , e  $\neg f_1$

Se  $f$  è una formula e  $x$  è una variabile allora anche  $\exists x(f)$  e  $\forall x(f)$  anche sono formule

# Esempi di interrogazioni nel Calcolo relazionale sui Domini

- Sulla base del modello già visto :  
trovare matricola , nome, età e stipendio degli impiegati che guadagnano più di 40mila euro

$\{ \text{Matr:}m, \text{Nome:}n, \text{Età:}e, \text{Stip:} s \mid$

$\text{IMPIEGATI}(\text{Matr:}m, \text{Nome:}n, \text{Età:}e, \text{Stip:} s ) \wedge s > 40 \}$

# Calcolo relazionale sui Domini

- ...ma un' espressione di un linguaggio di interrogazione sarebbe utile che fosse indipendente dal dominio
- Abbiamo bisogno di un' altra versione del calcolo relazionale, in cui le variabili, anziché denotare singoli valori, denotino tuple.

# Considerazioni sul Calcolo relazionale sui Domini

- Il Calcolo relazionale sui Domini ha dei difetti :
  - Agisce sui domini invece che sui valori
  - Per il motivo precedente diventa "verboso" (ha bisogno di tante variabili)
  - Può portare a espressioni che non hanno senso
- Occorre un linguaggio che ' focalizzi' le tuple di interesse.....

# Calcolo Relazionale su Tuple con dichiarazione di range

# Calcolo relazionale su Tuple con dichiarazione di range

- Le espressioni hanno la forma :

{Target list | Range list | formula}

Dove

- Target list lista degli obiettivi con elementi

$Y:x.Z$  o  $x.Z$  se  $Z:x.Z$  o  $x.*$

- Range list elenco delle variabili libere della formula con i relativi campi di variabilità

# Calcolo relazionale su Tuple con dichiarazione di range (2)

- Le espressioni hanno la forma :

$\{\text{Target list} \mid \text{Range list} \mid \text{formula}\}$

e dove

- **formula** è una formula del tipo :

- $x.A\theta c$  o  $x.A\theta y.B$
- **connettivi di formule**
- $\exists x(R)(f)$  o  $\forall x(R)(f)$



# Esempi di interrogazioni nel Calcolo relazionale su Tuple con dichiarazione di range

Sulla base del modello già visto :

Trovare matricola , nome, età degli impiegati che  
guadagnano più di 40mila euro

$$\{i.(Matr, Nome, Et\grave{a}) \mid i(IMPIEGATI) \mid i.Stip > 40\}$$

Se voglio tutti gli attributi diventa :

$$\{i.* \mid i(IMPIEGATI) \mid i.Stip > 40\}$$

# Esempi di interrogazioni nel Calcolo relazionale su Tuple con dichiarazione di range

Trovare le matricole dei capi degli  
impiegati che guadagnano più di 40mila  
euro

$$\{s.\text{Capo} \mid i(\text{IMPIEGATI}), s(\text{SUPERVISIONE}) \mid \\ i.\text{Matr}=s.\text{Impiegato} \wedge i.\text{Stip} > 40\}$$

# Esempi di interrogazioni nel Calcolo relazionale su Tuple con dichiarazione di range

Trovare nome e stipendio dei capi degli  
impiegati che guadagnano più di 40mila  
euro

$$\{\text{NomeC}, \text{StipC} : i' . (\text{Nome}, \text{Stip}) \mid i' (\text{IMPIEGATI}), \\ s(\text{SUPERVISIONE}), i(\text{IMPIEGATI}) \mid \\ i' \text{Matr} = s.\text{Capo} \wedge s.\text{Impiegato} = i.\text{Matr} \wedge i.\text{Stip} \\ > 40\}$$

# Calcolo relazionale su Tuple con dichiarazione di range

Il Calcolo su tuple però non permette di esprimere tutte le interrogazioni che possono essere formulate in Algebra relazionale.

Ad es. non c'è l'*unione*, per questo nei linguaggi interrogativi viene aggiunto esplicitamente un costrutto di *unione*.

# Riassumendo :

Algebra relazionale

Calcolo relazionale

