

# Algebra Relazionale

P. Rullo

[rullo@unical.it](mailto:rullo@unical.it)

# Algebra Relazionale

- É un linguaggio di interrogazione costituito da un insieme di *operatori* definiti su relazioni che producono relazioni
- Le interrogazioni vengono formulate attraverso espressioni algebriche
- Operatori
  - Insiemistici – mutuati dalla teoria degli insiemi
  - Specifici dell'AR
    - unari
    - binari di tipo Join

# Operatori insiemistici

- Le relazioni sono insiemi (di tuple) e per questo possono essere manipolate attraverso operatori mutuati dalla teoria degli insiemi
- A differenza degli insiemi, le relazioni hanno uno schema
- Relazioni che hanno lo stesso schema hanno elementi omogenei
- Le operazioni di unione, sottrazione e intersezione possono essere fatte a parità di schema, ad es., si può fare l'unione di un insieme di persone con un altro insieme di persone, ma non con un insieme di macchine

# Operatori insiemistici

- Date due relazioni  $R(X)$  e  $S(X)$ , con lo *stesso insieme di attributi*  $X=\{A_1, \dots, A_n\}$ , la loro unione, differenza e intersezione è una relazione  $T(X)$ , che ha gli stessi attributi, e la cui estensione è la seguente

$$T = R \cup S = \{t \mid t \in R \text{ or } t \in S\}$$

$$T = R - S = \{t \mid t \in R \text{ and } t \notin S\}$$

$$T = R \cap S = \{t \mid t \in R \text{ and } t \in S\}$$

# Operatori insiemistici

R(A,B)

A	B
a	b
b	c
h	e

S(A,B)

A	B
a	b
f	c
h	e

$$T(A,B) = R(A,B) \cup S(A,B)$$

A	B
a	b
f	c
h	e
b	c

$$T(A,B) = R(A,B) - S(A,B)$$

A	B
b	c

$$T(A,B) = R(A,B) \cap S(A,B)$$

A	B
a	b
h	e

# Operatori specifici unari

- *Selezione*:  $\sigma_F (R)$  è un operatore unario che restituisce una relazione  $S$  le cui tuple sono tutte e solo le tuple di  $R$  che soddisfano il predicato  $F$

$$S = \sigma_F (R) = \{t \in R \mid t \text{ soddisfa } F\}$$

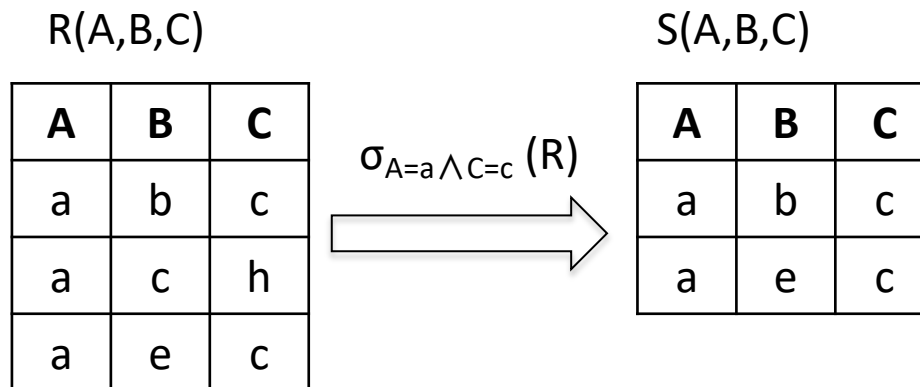
- $F$  è un predicato definito sugli attributi di  $R$  che utilizza operatori logici e operatori di confronto
- Si noti che gli attributi di  $S$  coincidono con quelli di  $R$

# Operatori specifici unari

- **Esempio:** data la relazione  $R(A,B,C)$

$$\sigma_{A=a \wedge C=c} (R)$$

restituisce una relazione  $S(A,B,C)$  che ha gli stessi attributi di  $R$  e la cui estensione consiste di tutte le tuple di  $R$  che soddisfano la condizione  $A=a \wedge C=c$



# Operatori specifici unari

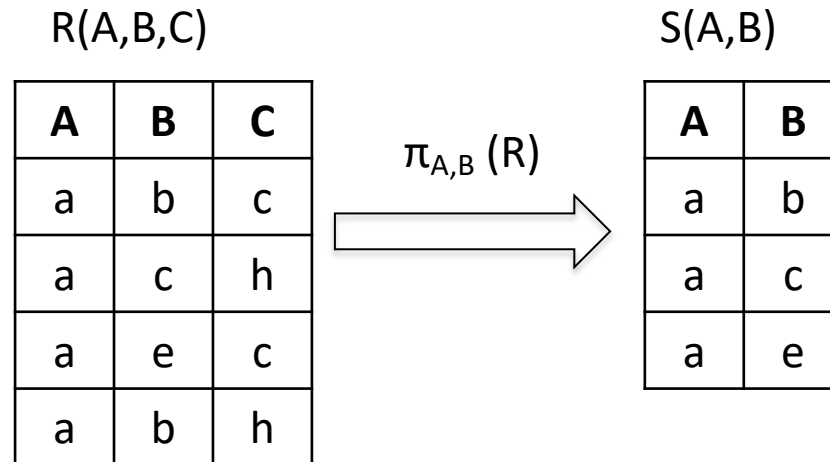
- *Proiezione*:  $\pi_Y (R)$  è un operatore unario che, data una relazione con schema  $R(X)$  e un sottoinsieme di attributi  $Y$  di  $X$ , restituisce una relazione  $S(Y)$  le cui tuple si ottengono da quelle di  $R$  considerando solo i valori sugli attributi in  $Y$

$$S(Y) = \pi_Y (R) = \{t[Y] \mid t \in R\}$$



# Operatori specifici unari

- **Esempio:** data  $R(A,B,C)$ , la proiezione  $\pi_{A,B}(R)$  restituisce una relazione  $S(A,B)$  che ha gli attributi A, B specificati come pedice dell'operatore e la cui estensione consiste di tutte e solo le tuple di R proiettate sugli stessi attributi



# Operatori specifici unari

- *Ridenominazione*: operatore unario che modifica lo **schema** di una relazione cambiando il nome di uno o più attributi

$$\rho_{A1...An \leftarrow B1... Bn} (R)$$

- **Esempio**: date le relazioni *Padre(padre, figlio)* e *Madre(madre, figlio)*
  - $\text{Padre}(\text{genitore}, \text{figlio}) \leftarrow \rho_{\text{genitore} \leftarrow \text{padre}} (\text{Padre})$
  - $\text{Madre}(\text{genitore}, \text{figlio}) \leftarrow \rho_{\text{genitore} \leftarrow \text{madre}} (\text{Madre})$
- Tale ridenominazione è necessaria se, ad esempio, vogliamo creare la relazione Genitore

$$\text{Genitore}(\text{genitore}, \text{figlio})$$

$$= \rho_{\text{genitore} \leftarrow \text{padre}} (\text{Padre}) \cup \rho_{\text{genitore} \leftarrow \text{madre}} (\text{Madre})$$

# Operatori binary di tipo Join

- É l'operatore che consente di connettere informazioni contenute in relazioni diverse
  - Join naturale
  - Equi-join
  - Theta-join
  - Join esterni

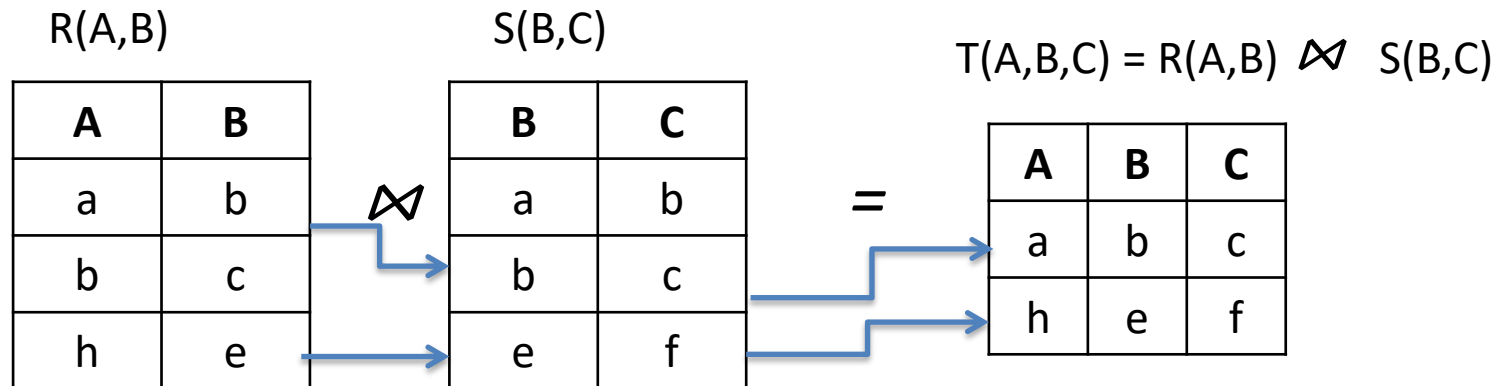
# Join naturale

- Correla tuple in relazioni diverse sulla base di valori uguali di *attributi con lo stesso nome*
- Il join naturale  $R(X) \bowtie S(Y)$  produce la relazione  $T(Z)$ , con  $Z = X \cup Y$  ed estensione definita come segue

$$T = \{t \text{ su } X \cup Y \mid t[X] \in R \text{ e } t[Y] \in S\}$$

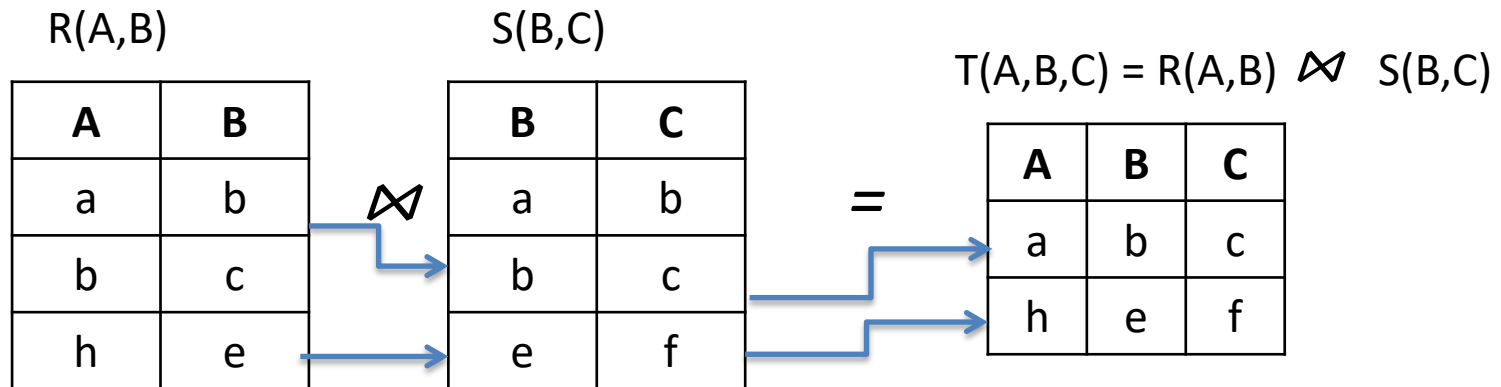
# Join naturale

- **Esempio:** Siano date due relazioni  $R(X)$  e  $S(Y)$ , con  $X = \{A,B\}$  e  $Y = \{B,C\}$ . Il join naturale produce una relazione  $T(Z)$ , con  $Z = X \cup Y = \{A,B,C\}$ , e le tuple sotto riportate



# Join naturale

- **Esempio:** Si noti che le tuple di T si ottengono concatenando le tuple di R e di S con lo stesso valore dell'attributo in comune B



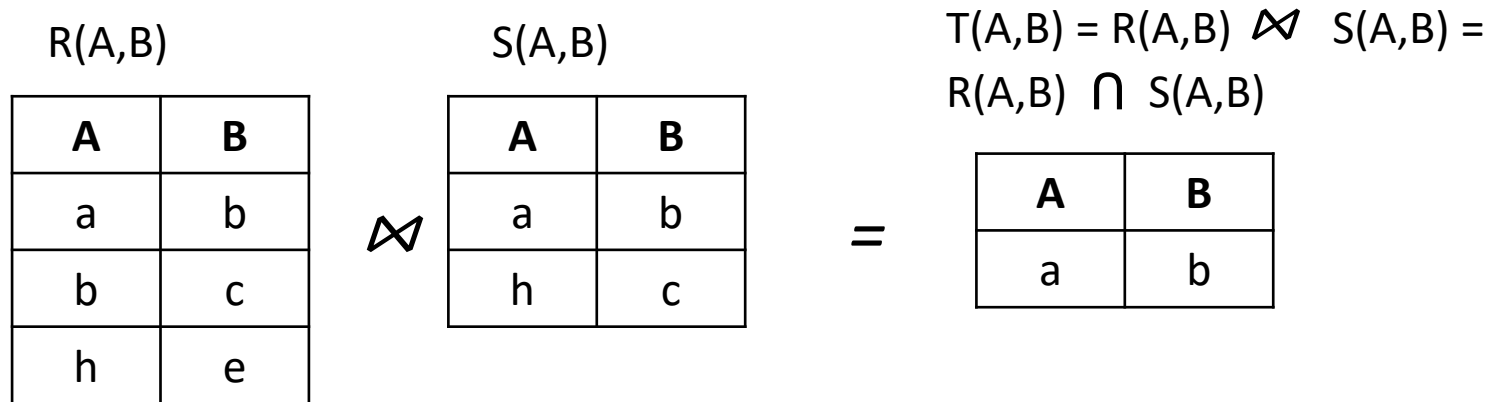
# Join naturale

- NOTA: se  $X \cap Y = \phi$ , cioè le due relazioni non hanno attributi in comune, il join naturale degenera nel prodotto cartesiano

R(A,B)			S(C,D)			T(A,B,C,D) = R(A,B) $\bowtie$ S(C,D)			
<b>A</b>	<b>B</b>		<b>C</b>	<b>D</b>		<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
a	b	$\bowtie$	a	b	=	a	b	a	b
b	c		b	c		a	b	b	c
h	e					b	c	a	b
						b	c	b	c
						h	e	a	b
						h	e	b	c

# Join naturale

- NOTA: se  $X = Y$ , cioè le due relazioni hanno gli stessi attributi, il join naturale coincide con l'intersezione





# Equi-join

- É un operatore che consente di eseguire join anche tra relazioni che non hanno attributi in comune, senza che questo degeneri nel prodotto cartesiano

$$R \bowtie_F S$$

dove  $F$  è un predicato che esprime condizioni di uguaglianza tra attributi delle due relazioni

- Ad esempio, dati  $R(A,B,C)$  e  $S(X,Y,Z)$ ,  $F$  potrebbe essere

$$F = (A=X \text{ and } B=Y)$$

- Il risultato è una relazione che ha gli attributi di  $R$  e di  $S$  prendendo una sola volta quelli che appaiono in  $F$  e come tuple la concatenazione delle tuple di  $R$  e di  $S$  che hanno gli stessi valori sugli attributi di join

# Equi-join

- **Esempio:**  $R(A,B,C) \bowtie_{A=X \text{ and } B=Y} S(X,Y,Z)$

R(A,B,C)				S(X,Y,Z)				T(A,B,C,Z)			
A	B	C		X	Y	Z		A	B	C	Z
a	b	e	$\bowtie$	a	b	s	$=$	a	b	e	s
b	c	f		h	c	a					
h	e	g									

# Equi-Join

- $\text{Prof}(\underline{\text{codP}}, \text{nome}, \text{età}, \text{dip}^*)$
- $\text{Dip}(\underline{\text{codD}}, \text{nomeD}, \text{univ}, \text{dir}^*)$

$$\text{Prof} \bowtie_{\text{dip}=\text{codD}} \text{Dip}$$

- Il risultato di tale espressione è una relazione che ha il seguente schema
  - $\text{ProfDip}(\underline{\text{codP}}, \text{nome}, \text{età}, \text{dip}, \text{nomeD}, \text{univ}, \text{dir}^*)$
- in cui i due attributi  $\text{dip}$  e  $\text{codD}$  sono rappresentati solo da uno di essi
- É facile verificare la seguente equivalenza:

$$\text{Prof} \bowtie_{\text{dip}=\text{codD}} \text{Dip} \equiv \text{Prof} \bowtie (\rho_{\text{dip} \leftarrow \text{codD}} \text{Dip})$$

# Equi-Join

Prof

CodP	Nome	Età	dip
I1	Aldo	35	d1
I2	Lucia	40	d1
I3	Luisa	38	d2
I4	Pino	45	d3

Dip

CodD	NomeD	Univ	dir
d1	Matematica	Unical	I2
d2	Informatica	Milano	I3
d3	Economia	Messina	I4

$$Prof \bowtie_{dip=codD} Dip$$

CodP	Nome	Età	dip	NomeD	Univ	dir
I1	Aldo	35	d1	Matematica	Unical	I2
I2	Lucia	40	d1	Matematica	Unical	I2
I3	Luisa	38	d2	Informatica	Milano	I3
I4	Pino	45	d3	Economia	Messina	I4

# Equi-join

- L'equi-join è un operatore *derivato*, nel senso che può essere espresso attraverso altri operatori
- Siano R e S due relazioni senza attributi in comune

$$R \bowtie_F S = \pi_Z(\sigma_F(R \bowtie S))$$

dove  $\pi_Z$  produce la proiezione su tutti gli attributi di R e S, prendendo una sola volta quelli che appaiono in F

- **Esercizio:** verificare la suddetta formula sui dati Prof-Dip
- **NOTA:** quanto detto vale per ogni forma di join

# Join esterni

- Il join esterno (outer join) include tutte le tuple di una relazione estese con le tuple dell'altra relazione che rispettano la condizione di join
- Join esterno sinistro: include tutte le tuple della relazione R (di sinistra)

$$R \bowtie^L S$$

- Join esterno destro: include tutte le tuple della relazione S (di destra)

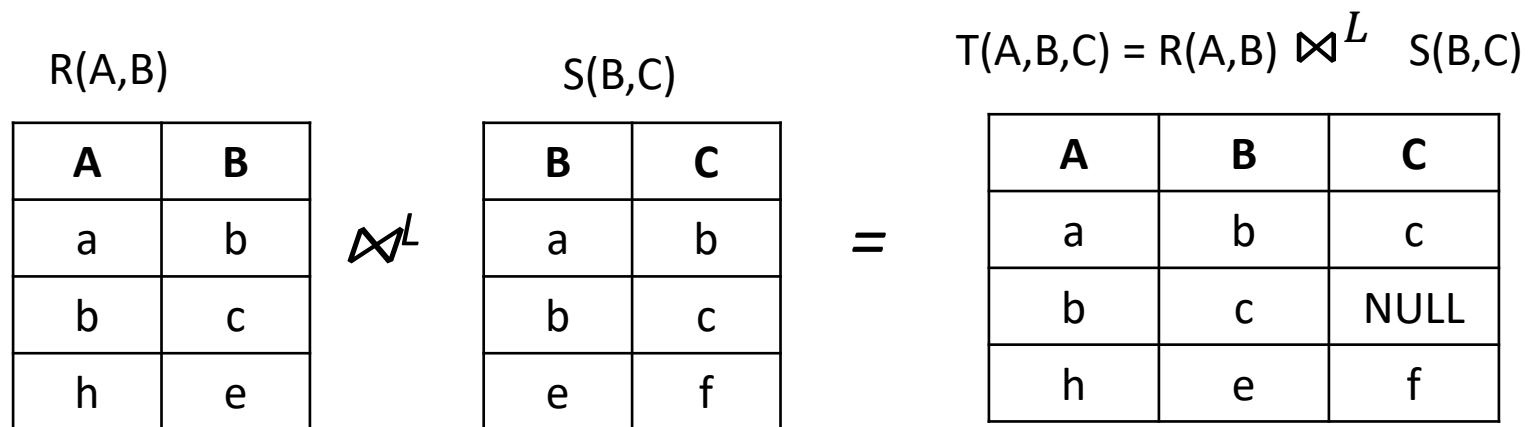
$$R \bowtie^R S$$

- Join esterno full: include tutte le tuple delle relazioni R ed S

$$R \bowtie^F S$$

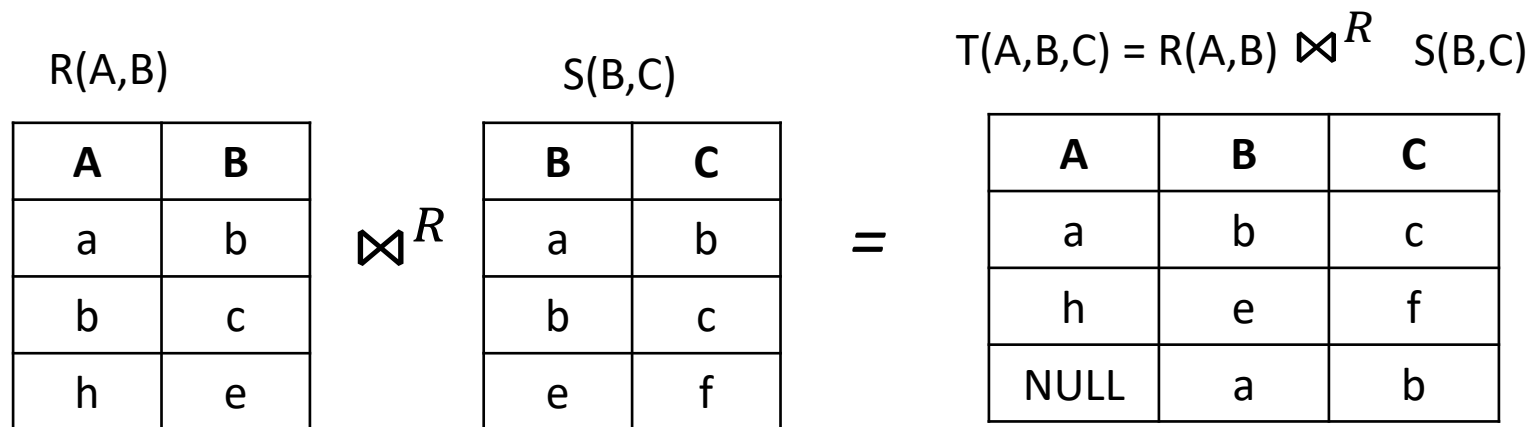
# Join esterni

- Join esterno sinistro  $R \bowtie^L S$



# Join esterni

- Join esterno destro  $R \bowtie^R S$





# Join esterni

- Join esterno full  $R \bowtie^F S$

$R(A,B)$

A	B
a	b
b	c
h	e

$\bowtie^F$

$S(B,C)$

B	C
a	b
b	c
e	f

$=$

$T(A,B,C) = R(A,B) \bowtie^F S(B,C)$

A	B	C
a	b	c
h	e	f
b	c	NULL
NULL	a	b

NOTA:  $R \bowtie^F S = (R \bowtie^L S) \cup (R \bowtie^R S)$