Lezione 4 – Algebra relazionale II

Prof.ssa Maria De Marsico
demarsico@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Parte del materiale è tratto dal materiale associato al testo Atzeni, Ceri, Paraboschi, Torlone
Basi di dati - Modelli e linguaggi di interrogazione , McGraw-Hill, 2002.

Informazioni in più relazioni

- Vedremo che per garantire determinate «buone» qualità di una relazione occorre rappresentare **separatamente** (in relazioni diverse) **concetti diversi**

- Capita molto spesso che le informazioni che interessano per rispondere ad una interrogazione sono **distribuite** in più relazioni, in quanto coinvolgono **più oggetti** in qualche modo associati

- Occorre **individuare** le relazioni in cui si trovano le informazioni che ci interessano, e **combinare** queste informazioni in maniera **opportuna**

Prodotto cartesiano

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- Consente di costruire una relazione contenente tutte le ennuple che si ottengono concatenando una ennupla del primo operando con una ennupla del secondo operando

- Si denota con il simbolo x

$$r_1 x r_2$$

- Si usa quando le informazioni che occorrono a rispondere ad una query si trovano in relazioni diverse

- Ma attenzione …

Prodotto cartesiano

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Ordine

O#	C#	A#	N-pezzi
O1	C1	A1	100
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200
O1	C1	A3	100

Query: Dati dei clienti e degli ordini

(*Cliente* × *Ordine*)

Prodotto cartesiano

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Ordine

O#	C#	A#	N-pezzi
O1	C1	A1	100
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200
O1	C1	A3	100

Per poter distinguere gli attributi con lo stesso nome nello schema risultante possiamo usare l'operazione di ridenominazione (ρ) per utilizzare una copia della relazione Ordine in cui l'attributo C# diventa CC#

$$OrdineR = \rho_{CC\# \leftarrow C\#}(Ordine)$$

Risultato del prodotto cartesiano

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	CC#	A#	N-pezzi
Rossi	C1	Roma	O1	C1	A1	100
Rossi	C1	Roma	O2	C2	A2	200
Rossi	C1	Roma	O3	C3	A2	150
Rossi	C1	Roma	O4	C4	A3	200
Rossi	C1	Roma	O1	C1	A2	200
Rossi	C2	Milano	O1	C1	A1	100
…	…	…		…	…	…
Bianchi	C3	Roma	O3	C1	A1	100
…	…	…		…	…	…
Verdi	C4	Roma	O4		A3	200
…	…	…		…	…	…

Risultato del prodotto cartesiano

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	CC#	A#	N-pezzi
Rossi	C1	Roma	O1	C1	A1	100
Rossi	C1	Roma	O2	C2	A2	200
Rossi	C1	Roma		C3	A2	150
Rossi	C1	Roma		C4	A3	200
Rossi	C1	Roma		C1	A2	200
Rossi	C2	Milano		C1	A1	100
…	…	…		…	…	…
Bianchi	C3	Roma		C1	A1	100
…	…	…		…	…	…
Verdi	C4	Roma			A3	200
…	…	…		…	…	…

Una query corretta

Nome	**C#**	**Città**	O#	**CC#**	**A#**	**N-pezzi**
Rossi	C1	Roma	O1	C1	A1	100
Rossi	C1	Roma	O1	C1	A2	200
Rossi	C1	Roma	O1	C1	A3	100
Rossi	C2	Milano	O2	C2	A2	200
Bianchi	C3	Roma	O3	C3	A2	150
Verdi	C4	Roma	O4	C4	A3	200

Query: Dati dei clienti e dei loro ordini

$\sigma_{C\#=CC\#}(Cliente \times OrdineR)$

Una query più «elegante»

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	CC#	A#	N-pezzi
Rossi	C1	Roma	O1	C1	A1	100
Rossi	C1	Roma	O1	C1	A2	200
Rossi	C1	Roma	O1	C1	A3	100
Rossi	C2	Milano	O2	C2	A2	200
Bianchi	C3	Roma	O3	C3	A2	150
Verdi	C4	Roma	O4	C4	A3	200

$\sigma_{C\#=CC\#}(Cliente \times OrdineR)$

Query: Dati dei clienti e dei loro ordini

$\pi_{\text{Nome C\# Città O\# A\# N-pezzi}}(\sigma_{C\#=CC\#}(Cliente \times OrdineR))$

Eliminiamo gli attributi duplicati (che di solito sono proprio quelli della condizione di selezione)

Una query più «elegante»

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A1	100
Rossi	C1	Roma	O1	A2	200
Rossi	C1	Roma	O1	A3	100
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

Query: Dati dei clienti e dei loro ordini

$\pi_{\text{Nome C\# Città O\# A\# N-pezzi}}(\sigma_{\text{C\#=CC\#}}(\mathit{Cliente} \times \mathit{OrdineR}))$

Eliminiamo gli attributi duplicati (che di solito sono proprio quelli della condizione di selezione)

Una query un po' più «complessa»

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A1	100
Rossi	C1	Roma	O1	A2	200
Rossi	C1	Roma	O1	A3	100
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

Query: Dati dei clienti e dei loro ordini che superano i 100 pezzi

Ripartiamo dalle tuple «corrette» …

Una query un po' più «complessa»

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A2	200
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

Query: Dati dei clienti e dei loro ordini che superano i 100 pezzi

$$\pi_{\text{Nome C\# Città O\# A\# N-pezzi}}(\sigma_{\text{C\#=CC\#} \wedge \text{ N-pezzi>100}}(\mathit{Cliente} \times \mathit{OrdineR}))$$

Join naturale

- Consente di selezionare le tuple del prodotto cartesiano dei due operandi che soddisfano la condizione:
  - $R_1.A_1 = R_2.A_1 \wedge R_1.A_2 = R_2.A_2 \wedge \ldots \wedge R_1.A_k = R_2.A_k$

- (dove R_1 ed R_2 sono i nomi delle relazioni operando e $A_1, A_2, \ldots, A_k$ sono gli attributi comuni, **cioè con lo stesso nome**, delle relazioni operando) eliminando le ripetizioni degli attributi

- $r_1 \rhd\lhd r_2 = \pi_{XY}(\sigma_C (r_1 \times r_2))$

- dove:
- *C:* $R_1.A_1 = R_2.A_1 \wedge \ldots \wedge R_1.A_k = R_2.A_k$
- *X* è l'insieme di attributi di r_1
- *Y* è l'insieme di attributi di r_2 *che non sono attributi di* r_1

- DA RICORDARE:
- nel join naturale gli **attributi** della **condizione** che **consente di unire solo le ennuple giuste** hanno lo **stesso nome**
- vengono unite le ennuple **in cui questi attributi hanno lo stesso valore**

Join naturale

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Ordine

O#	C#	A#	N-pezzi
O1	C1	A1	100
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200
O1	C1	A3	100

Query: Dati dei clienti e dei loro ordini

Cliente $\triangleright\triangleleft$ *Ordine*

Risultato del join naturale

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A1	100
Rossi	C1	Roma	O1	A2	200
Rossi	C1	Roma	O1	A3	100
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

Esempio 1

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Torniamo al nostro esempio e risolviamolo con il join naturale

Query : Nomi dei clienti che hanno ordinato più di 100 pezzi per almeno un articolo

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Ordine

O#	C#	A#	N-pezzi
O1	C1	A1	100
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200
O1	C1	A3	100

$$\pi_{\text{Nome}}(\sigma_{\text{N-pezzi}>100}(\text{Cliente} \triangleright\!\triangleleft \text{Ordine}))$$

Questa volta ci interessano solo i nomi .. Ma attenzione …

Esempio 1

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A1	100
Rossi	C1	Roma	O1	A2	200
Rossi	C1	Roma	O1	A3	100
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

Cliente $\triangleright\!\triangleleft$ Ordine

Esempio 1

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A2	200
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

$$\sigma_{\text{N-pezzi}>100}(\text{Cliente} \triangleright\!\triangleleft \text{Ordine})$$

Esempio 1

Nome
Rossi
Bianchi
Verdi

$$\pi_{\text{Nome}}(\sigma_{\text{N-pezzi}>100}(\text{Cliente} \triangleright\!\triangleleft \text{Ordine}))$$

ATTENZIONE! Notare che il nome da solo non identifica il cliente, e la query elimina i nomi duplicati

$$\pi_{\text{Nome, Citta}}(\sigma_{\text{N-pezzi}>100}(\text{Cliente} \triangleright\!\triangleleft \text{Ordine}))$$

Nome	Città
Rossi	Roma
Rossi	Milano
Bianchi	Roma
Verdi	Roma

Ricordiamo un esempio già visto … per sicurezza meglio utilizzare **una chiave** (il codice cliente) perché potremmo avere degli omonimi nella stessa città …

$$\pi_{\text{Nome, C\#}}(\sigma_{\text{N-pezzi}>100}(\ldots))$$

Esempio 2

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Query : Nomi e città dei clienti che hanno ordinato più di 100 pezzi per almeno un articolo con prezzo superiore a 2

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Ordine

O#	C#	A#	N-pezzi
O1	C1	A1	100
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200
O1	C1	A3	100

Articolo

A#	Denom.	Prezzo
A1	Piatto	3
A2	Bicchiere	2
A3	Tazza	4

Osserviamo che dobbiamo coinvolgere anche una terza relazione, che nella nostra base di dati contiene ile informazioni complete sugli articoli commercializzati

$$\pi_{\text{Nome,Città}}(\sigma_{\text{N-pezzi}>100 \wedge \text{Prezzo}>2}((\text{Cliente} \triangleright\!\triangleleft \text{Ordine}) \triangleright\!\triangleleft \text{Articolo}))$$

Esempio 2

Nome	C#	Città	O#	A#	N-pezzi	Denom.	Prezzo
Rossi	C1	Roma	O1	A1	100	Piatto	3
Rossi	C1	Roma	O1	A2	200	Bicchiere	2
Rossi	C1	Roma	O1	A3	100	Tazza	4
Rossi	C2	Milano	O2	A2	200	Bicchiere	2
Bianchi	C3	Roma	O3	A2	150	Bicchiere	2
Verdi	C4	Roma	O4	A3	200	Tazza	4

$$(\text{Cliente} \triangleright\!\triangleleft \text{Ordine}) \triangleright\!\triangleleft \text{Articolo}$$

$$(\sigma_{\text{N-pezzi}>100 \wedge \text{Prezzo}>2}((\text{Cliente} \triangleright\!\triangleleft \text{Ordine}) \triangleright\!\triangleleft \text{Articolo}))$$

Anche in questo caso ci interessa solo un sottoinsieme «sensato» del prodotto cartesiano, cioè quello in cui vengono combinate le informazioni relative **agli oggetti realmente associati** tra di loro

Esempio 2

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	A#	N-pezzi	Denom.	Prezzo
Verdi	C4	Roma	O4	A3	200	Tazza	4

$$\sigma_{\text{N-pezzi}>100 \wedge \text{Prezzo}>2}((\text{Cliente} \triangleright\!\triangleleft \text{Ordine}) \triangleright\!\triangleleft \text{Articolo})$$

$$\pi_{\text{Nome,Città}}(\sigma_{\text{N-pezzi}>100 \wedge \text{Prezzo}>2}((\text{Cliente} \triangleright\!\triangleleft \text{Ordine}) \triangleright\!\triangleleft \text{Articolo}))$$

Esempio 2

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	Città
Verdi	Roma

$$\pi_{\text{Nome,Città}}(\sigma_{\text{N-pezzi}>100 \wedge \text{Prezzo}>2}((\text{Cliente} \triangleright\!\triangleleft \text{Ordine}) \triangleright\!\triangleleft \text{Articolo}))$$

Esempio 2: alternativa

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Query : Nomi e città dei clienti che hanno ordinato più di 100 pezzi per almeno un articolo con prezzo superiore a 2

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Ordine

O#	C#	A#	N-pezzi
O1	C1	A1	100
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200
O1	C1	A3	100

Articolo

A#	Denom.	Prezzo
A1	Piatto	3
A2	Bicchiere	2
A3	Tazza	4

Effettuando **prima** la selezione delle tuple che ci interessano, l'operazione è più efficiente perché evitiamo di combinare inutilmente dati che poi non ci serviranno

Esempio 2: alternativa

O#	C#	A#	N-pezzi
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200

$\sigma_{\text{N-pezzi}>100}(\text{Ordine})$

Esempio 2: alternativa

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

$\triangleright\triangleleft$

O#	C#	A#	N-pezzi
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200

$\sigma_{\text{N-pezzi}>100}(\text{Ordine})$

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A2	200
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

$\text{Cliente} \triangleright\triangleleft \sigma_{\text{N-pezzi}>100}(\text{Ordine})$

Esempio 2 : alternativa

Articolo

A#	Denom.	Prezzo
A1	Piatto	3
A2	Bicchiere	2
A3	Tazza	4

$\pi_{A\#,Prezzo}$ →

A#	Prezzo
A1	3
A2	2
A3	4

$\pi_{A\#,Prezzo}(Articolo)$

$\sigma_{Prezzo>2}$ ↓

A#	Prezzo
A1	3
A3	4

$\sigma_{Prezzo>2}\ (\pi_{A\#,Prezzo}(Articolo))$

Esempio 2 : alternativa

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A2	200
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

$\triangleright\triangleleft$

A#	Prezzo
A1	3
A3	4

$\sigma_{Prezzo>2}\ (\pi_{A\#,Prezzo}(Articolo))$

$Cliente \triangleright\triangleleft \sigma_{N\text{-}pezzi>100}(Ordine)$

Nome	C#	Città	A#	N-pezzi	Prezzo
Verdi	C4	Roma	A3	200	4

$(Cliente \triangleright\triangleleft \sigma_{N\text{-}pezzi>100}(Ordine)) \triangleright\triangleleft \sigma_{Prezzo>2}\ (\pi_{A\#,Prezzo}(Articolo))$

Esempio 2

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	Città
Verdi	Roma

$$\pi_{\text{Nome,Città}} ((\text{Cliente} \triangleright\!\triangleleft \sigma_{\text{N-pezzi}>100}(\text{Ordine})) \triangleright\!\triangleleft \sigma_{\text{Prezzo}>2} (\pi_{\text{A\#,Prezzo}}(\text{Articolo})))$$

Join naurale: casi limite

Caso limite 1:

- Le relazioni contengono attributi con lo stesso nome ma non esistono ennuple con lo stesso valore per tali attributi in entrambe le relazioni
- Risultato: il join naturale **è vuoto**

Esempio: problema precedente con condizione su costo dell'articolo <2 e una nuova tabella Articoli

Articolo

A#	Denom.	Prezzo
A1	Piatto	3
A2	Bicchiere	2
A3	Tazza	4
A4	Piattino	1

A#	Prezzo
A4	1

$\sigma_{Prezzo<2}\ (\pi_{A\#,Prezzo}(Articolo))$

Join naurale: casi limite

Esempio: problema precedente con condizione su costo dell'articolo <2 e una nuova tabella **Articolo**

Il join tra **Cliente** e **Ordine** è uguale al caso precedente e dà lo stesso risultato ma …

Nome	C#	Città	O#	A#	N-pezzi
Rossi	C1	Roma	O1	A2	200
Rossi	C2	Milano	O2	A2	200
Bianchi	C3	Roma	O3	A2	150
Verdi	C4	Roma	O4	A3	200

$\triangleright\triangleleft$

A#	Prezzo
A4	1

$\sigma_{Prezzo<2}\ (\pi_{A\#,Prezzo}(Articolo))$

$Cliente \triangleright\triangleleft \sigma_{N\text{-}pezzi>100}(Ordine)$

… non ci sono tuple con lo stesso valore per A# nelle due relazioni, quindi il risultato da qui in poi è **vuoto**

Join naurale: casi limite

Caso limite 2:

- Le relazioni **non** contengono attributi con lo stesso nome, quindi la condizione $R_1.A_1 = R_2.A_1 \wedge R_1.A_2 = R_2.A_2 \wedge \ldots \wedge R_1.A_k = R_2.A_k$

dove R_1 ed R_2 sono i nomi delle relazioni operando e $A_1, A_2, \ldots, A_k$ sono gli attributi comuni, **cioè con lo stesso nome**, diventa **vuota se questi attributi non esistono,** quindi **non c'è** condizione e quindi si degenera nel prodotto cartesiano

In pratica abbiamo il problema opposto rispetto a quello di distinguere la provenienza dell'attributo, quindi applichiamo la ridenominazione proprio per ottenete lo stesso nome per gli attributi corrispondenti nelle relazioni coinvolte

Join naurale: casi limite

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Ordine

O#	CC#	A#	N-pezzi
O1	C1	A1	100
O2	C2	A2	200
O3	C3	A2	150
O4	C4	A3	200
O1	C1	A2	200
O1	C1	A3	100

Join naurale: casi limite

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	CC#	A#	N-pezzi
Rossi	C1	Roma	O1	C1	A1	100
Rossi	C1	Roma	O2	C2	A2	200
Rossi	C1	Roma	O3	C3	A2	150
Rossi	C1	Roma	O4	C4	A3	200
Rossi	C1	Roma	O1	C1	A2	200
Rossi	C2	Milano	O1	C1	A1	100
…	…	…		…	…	…
Bianchi	C3	Roma	O3	C1	A1	100
…	…	…		…	…	…
Verdi	C4	Roma	O4		A3	200
…	…	…		…	…	…

Join naurale: casi limite

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Nome	C#	Città	O#	CC#	A#	N-pezzi
Rossi	C1	Roma	O1	C1	A1	100
Rossi	C1	Roma	O2	C2	A2	200
Rossi	C1	Roma		C3	A2	150
Rossi	C1	Roma		C4	A3	200
Rossi	C1	Roma		C1	A2	200
Rossi	C2	Milano		C1	A1	100
…	…	…		…	…	…
Bianchi	C3	Roma		C1	A1	100
…	…	…		…	…	…
Verdi	C4	Roma			A3	200
…	…	…		…	…	…

Join naurale: casi limite

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- Soluzione

$$OrdineR = \rho_{C\# \leftarrow CC\#}(Ordine)$$

Join naurale: possibili errori

- Ovviament eperchè il join abbia senso gli attributi con lo stesso nome devono avere anche **lo stesso significato**

Artista

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Quadro

Titolo	C#	Artista
Tit1	C1	C1
Tit2	C2	C3
Tit3	C3	C1
Tit4	C4	C2
Tit5	C5	C4
Tit6	C6	C2

Per avere senso, il join va effettuato tra Artista.C# e Quadro.Artista, quindi si usa un θ-join (vediamo dopo) oppure una ridenominazione

θ-join

- Consente di selezionare le tuple del prodotto cartesiano dei due operandi che soddisfano una condizione del tipo
  - AθB
- dove
  - θ è un operatore di confronto ($\theta \in \{<, =, >, \leq, \geq\}$),
  - A è un attributo dello schema del primo operando,
  - B è un attributo dello schema del secondo operando e
  - dom(A)=dom(B)

- $r_1 \underset{A\theta B}{\triangleright\!\triangleleft} r_2 = \sigma_{A\theta B}(r_1 \times r_2)$

Condizioni negative

Cliente

Nome	C#	Città
Rossi	C1	Roma
Rossi	C2	Milano
Bianchi	C3	Roma
Verdi	C4	Roma

Query: Dati dei clienti che si chiamano Rossi e NON risiedono a Roma

$$\sigma_{\neg(\text{Città='Roma'})\wedge\text{Nome='Rossi'}}(\text{Cliente})$$

Rossi	C2	Milano

Ricapitolando

- Quando le informazioni che ci occorrono non sono memorizzate tutte nella stessa relazione:
- Identifichiamo **tutte** e sole le relazioni che contengono informazioni di interesse
- Eventualmente **selezioniamo** preventivamente dei sottoinsiemi **rilevanti** per la nostra interrogazioni
- **Combiniamo** le informazioni attraverso i **valori** che da ogni tupla di una relazione fanno riferimento alle opportune tuple nelle altre