

The left side of the slide features a series of vertical stripes in various shades of blue. Overlaid on these stripes are several teal-colored circles of different sizes. One large circle is positioned near the top left, while several smaller circles are arranged in a descending pattern towards the bottom left. The title text is positioned to the right of these decorative elements.

# ALGEBRA RELAZIONALE

1

# OPERAZIONI NEL MODELLO RELAZIONALE

- Si possono esprimere in due formalismi di base
  - equivalenti sotto opportune ipotesi
- **Algebra relazionale**
  - Query = espressioni basate su relazioni base + operatori fra relazioni
- **Calcolo relazionale**
  - Query = insieme di elementi che soddisfano una formula logica

# ALGEBRA RELAZIONALE

- Le sue operazioni usano solo relazioni come argomento e producono relazioni
- Caratterizzata da cinque operazioni di base
  - Proiezione
  - Selezione
  - Prodotto cartesiano
  - Unione
  - Differenza
- Arricchita da operazioni derivate
  - non aumentano il potere espressivo dell'algebra
  - comode in pratica

**Join!!!**

# A CHE COSA SERVE?

- Modello alla base della semantica di SQL
- Linguaggio più semplice
  - più facile ragionare sul significato, casi limite...
  - si capisce meglio come si valutano le espressioni
  - più facile ragionare su complessità e possibili ottimizzazioni

**NON UNA VERA OPERAZIONE  
(MODIFICA SOLO LO SCHEMA):  
RIDENOMINAZIONE**

# ALGEBRA RELAZIONALE

- Per usare la notazione per nome serve un'operazione aggiuntionale
  - Ridenominazione
  - Permette di modificare il nome degli attributi
    - per evitare clash quando si mettono assieme relazioni con gli stessi nomi di attributi
- Se usassimo la notazione per posizione non servirebbe
  - mettendo assieme le relazioni le posizioni scalerebbero automaticamente

# RIDENOMINAZIONE

## Dati

- una relazione  $R$  con nomi di attributi  $C_1, C_2, \dots, C_k$  e dominio  $dom$
- una **lista**  $A_1, \dots, A_n$  di nomi **distinti** ( $i \neq j \Rightarrow A_i \neq A_j$ ) di suoi attributi
  - ovvero  $\{A_1, \dots, A_n\} \subseteq U_R$
  - ovvero  $\forall i \in [1, n] \exists j \in [1, k] \text{ t.c. } A_i = C_j$
- una lista  $B_1, \dots, B_n$  di nomi
  - distinti** fra loro
  - distinti** da tutti i nomi in  $U_R - \{A_1, \dots, A_n\}$

La **ridenominazione**  $\rho_{A_1, \dots, A_n \leftarrow B_1, \dots, B_n}(R)$  cambia i nomi degli attributi  $A_i$  in  $B_i$  cioè se  **$R' = \rho_{A_1, \dots, A_n \leftarrow B_1, \dots, B_n}(R)$**

- $R'$  ha nomi di attributi  $C'_1, C'_2, \dots, C'_k$  dove  $C'_i = B_j$  se  $A_j = C_i$  altrimenti  $C'_i = C_i$ 
  - $U_{R'} = U_R - \{A_1, \dots, A_n\} \cup \{B_1, \dots, B_n\} \Rightarrow$  **Rango di  $R'$  = Rango di  $R$**
- Il dominio  $dom'(C'_i) = dom(C_i)$
- $R'$  ha le stesse tuple di  $R \Rightarrow$  **Cardinalità di  $R'$  = Cardinalità di  $R$**

# RIDENOMINAZIONE – ESEMPIO

- Se ho una relazione Noleggio con nomi di attributi colloc, dataNol, codCli, dataRest
- La ridenominazione

$\rho_{\text{dataNol, dataRest}} \leftarrow \text{dataIn, dataFine}(\text{Noleggio})$

- Attributi?
  - colloc, dataIn, codCli, dataFine
- Domini?
  - gli stessi domini di colloc, dataNol, codCli, dataRest
- Tuple ?
  - le stesse di Noleggio
- In SQL come si esprime?  
SELECT colloc, dataNol as dataIn, codCli, dataRest as dataFine  
FROM Noleggio



# OPERAZIONI UNARIE: PROIEZIONE E SELEZIONE

# PROIEZIONE

## Dati

- una relazione  $R$  con nomi di attributi  $C_1, C_2, \dots, C_k$  e dominio  $dom$
- una lista  $A_1, \dots, A_n$  di nomi distinti di suoi attributi
- La **proiezione**  $\Pi_{A_1, \dots, A_n}(R)$  tiene solo le colonne  $A_i$   
se  **$R' = \Pi_{A_1, \dots, A_n}(R)$** 
  - $R'$  ha nomi di attributi  $A_1, \dots, A_n \Rightarrow$  **Rango di  $R' = n$**
  - Il dominio  $dom'(A_i) = dom(A_i)$
  - $t'$  è una tupla di  $R'$  se e solo se  $\exists$  una tupla  $t$  di  $R$  t.c.  $t'[A_i] = t[A_i]$  per ogni  $i$   
  
 $\Rightarrow$  **Cardinalità di  $R' \leq$  Cardinalità di  $R$**
- eliminazione dei duplicati

# PROIEZIONE – ESEMPIO

R

A	B	C
a	b	3
d	a	5
c	b	27

In SQL come si esprimono?

...quasi

in SQL le tabelle sono multinsiemi, in algebra relazionale insiemi!

SELECT A, C  
FROM R

SELECT B, A  
FROM R

distinct

SELECT B  
FROM R

B
b
a

$\Pi_B(R)$

A	C
a	3
d	5
c	27

$\Pi_{A,C}(R)$

B	A
b	a
a	d
b	c

$\Pi_{B,A}(R)$

11

# SELEZIONE

## Dati

- una relazione  $R$
- una formula  $F$
- La **selezione**  $\sigma_F(R)$  tiene solo le tuple di  $R$  su cui  $F$  vale true  
se  **$R' = \sigma_F(R)$** 
  - $R'$  ha gli stessi attributi di  $R$  con lo stesso dominio  $\Rightarrow$  **Rango di  $R' = \text{Rango di } R$**
  - $t$  è una tupla di  $R'$  se e solo se è una tupla  $t$  di  $R$  che soddisfa  $F$   
 $\Rightarrow$  **Cardinalità di  $R' \leq \text{Cardinalità di } R$**

# FORMULE/PREDICATI PER SELEZIONE

- Una **formula** (predicato)  $F$  su una relazione  $R$  ha una delle seguenti forme:
  - **atomo** (predicato semplice)
  - combinazione booleana di formule mediante i connettivi  $\wedge$  (AND),  $\vee$  (OR),  $\neg$  (NOT)
- Un atomo
  - ha la forma  $e \text{ op } e'$  dove
    - $\text{op}$  è un operatore relazionale di confronto  $>, <, >=, <=, =, \neq$
    - $e, e'$  sono o nomi di attributi di  $R$  o costanti
      - non entrambe costanti
  - ha la forma  $A \text{ is null}$  dove  $A$  è un attributo

# FORMULE/PREDICATI – VALIDITÀ

- Valutare una formula  $F$  su una tupla  $t$  di una relazione  $R$  può produrre true, falso o ? (unknown)
  - il risultato si indica  $t[F]$
  - la valutazione è definita induttivamente
- $t[e \text{ op } e'] = ?$  se  $t[e]$  oppure  $t[e']$  sono il valore nullo
- se  $t[e]$  e  $t[e']$  sono entrambi non nulli
  - $t[e \text{ op } e'] = \text{true}$  se  $t[e] \text{ op } t[e']$
  - $t[e \text{ op } e'] = \text{false}$  altrimenti
- $t[e \text{ is null}]$  vale vero se  $t[e]$  è il valore nullo, false altrimenti
- La validità degli operatori booleani è quella usuale, estesa su ?

interpretazione  
semantica  
dell'operatore

NOT	
TRUE	FALSE
FALSE	TRUE
?	?

AND	TRUE	FALSE	?
TRUE	TRUE	FALSE	?
FALSE	FALSE	FALSE	FALSE
?	?	FALSE	?

OR	TRUE	FALSE	?
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	?
?	TRUE	?	?

- $t[A]$  è il valore dell'attributo  $A$  in  $t$  e  $t[v] = v$

interpretazione  
semantica della costante

# FORMULE/PREDICATI– ESEMPI

- $\text{codCli}=6635$                       atomo/predicato semplice
- $\text{dataNol}=\text{dataRest}$               atomo/predicato semplice
- $\text{codCli}=6635 \wedge \text{dataNol}=\text{dataRest}$   
   formula

## SELEZIONE - DEFINIZIONE

- Una tupla  $t$  appartiene a  $R' = \sigma_F(R)$  se e solo se
  - $t$  appartiene a  $R$
  - $t \models F$
- Se nessuna tupla di  $R$  verifica il predicato  $F$ , allora il risultato è una relazione vuota



# SELEZIONE – ESEMPIO

R

A	B	C
a	b	c
d	a	f
c	b	d

A	B	C
a	b	c
c	b	d

$\sigma_{B=b \text{ OR } A=C}(R)$

SELECT DISTINCT \*  
FROM R  
WHERE B = b OR A = C

A	B	C
a	b	c
c	b	d

$\sigma_{B=b}(R)$

SELECT DISTINCT \*  
FROM R  
WHERE B = b

In SQL come si  
esprimono?

Non introducono duplicati  
ma li possono propagare

A	B	C
d	a	f

$\sigma_{\text{NOT } B=b}(R)$

SELECT DISTINCT \*  
FROM R  
WHERE NOT B = b

A	B	C

$\sigma_{B=b \text{ AND } A=C}(R)$

SELECT DISTINCT \*  
FROM R  
WHERE B = b AND A = C

# COMBINARE TUPLE IN RELAZIONI DIVERSE: PRODOTTO CARTESIANO

# PRODOTTO CARTESIANO

Date

- una relazione  $R_1$  con nomi di attributi  $A_1, A_2, \dots, A_n$  e dominio  $dom_1$
  - una relazione  $R_2$  con nomi di attributi  $B_1, B_2, \dots, B_k$  e dominio  $dom_2$
- tali che  $\{A_1, A_2, \dots, A_n\} \cap \{B_1, B_2, \dots, B_k\} = \emptyset$

Il **prodotto cartesiano**  $R = R_1 \times R_2$

- ha nomi di attributi  $A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_k$   
 $\Rightarrow$  **Rango di  $R$  = Rango di  $R_1$  + Rango di  $R_2$**
- Il dominio  $dom$  coincide con  $dom_1$  sugli attributi  $A_i$  e con  $dom_2$  sugli attributi  $B_i$
- una tupla  $t$  appartiene ad  $R \Leftrightarrow$   
è la concatenazione di una tupla  $t_1$  di  $R_1$  e una  $t_2$  di  $R_2 \Leftrightarrow$   
 $\exists$  tuple  $t_1$  di  $R_1$  e  $t_2$  di  $R_2$  t.c.  $t[A_i] = t_1[A_i] \ \forall i \in [1, n]$  e  $t[B_i] = t_2[B_i] \ \forall i \in [1, k]$   
 $\Rightarrow$  **Cardinalità di  $R$  = Cardinalità di  $R_1 \times$  Cardinalità di  $R_2$**   
caso degenerare: se  $R_1 = \emptyset$  (oppure  $R_2 = \emptyset$ ), allora  $R = \emptyset$

# PRODOTTO CARTESIANO – ESEMPIO

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
a	b	3
d	a	5
c	b	27

R<sub>1</sub>

R<sub>2</sub>

In SQL come si esprimono?

B <sub>1</sub>	B <sub>2</sub>
1	g
6	k

B <sub>1</sub>	B <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
1	g	a	b	3
6	k	a	b	3
1	g	d	a	5
6	k	d	a	5
1	g	c	b	27
6	k	c	b	27

R<sub>2</sub> X R<sub>1</sub>

SELECT  
DISTINCT \*  
FROM R2, R1

R<sub>1</sub> X R<sub>2</sub>

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	B <sub>1</sub>	B <sub>2</sub>
a	b	3	1	g
a	b	3	6	k
d	a	5	1	g
d	a	5	6	k
c	b	27	1	g
c	b	27	6	k

SELECT  
DISTINCT \*  
FROM R1, R2

# OPERAZIONI INSIEMISTICHE: UNIONE E DIFFERENZA (+ INTERSEZIONE OPERAZIONE DERIVATA)

# UNIONE

Date due relazioni  $R_1$  e  $R_2$  con nomi di attributi  $A_1, A_2, \dots, A_n$  e dominio  $dom$

L'**unione**  $R = R_1 \cup R_2$

- ha nomi di attributi  $A_1, A_2, \dots, A_n$

$\Rightarrow$  **Rango di  $R$  = Rango di  $R_1$  = Rango di  $R_2$**

- Dominio  $dom$

- una tupla  $t$  appartiene ad  $R \Leftrightarrow$

- $t$  appartiene ad  $R_1$

**oppure**

- $t$  appartiene ad  $R_2$

$\Rightarrow$  **Cardinalità di  $R \leq$  Cardinalità di  $R_1$  + Cardinalità di  $R_2$**

- tuple che compaiono in entrambi appartengono una sola volta al risultato

- caso degenere: se  $R_1 \subseteq R_2$ , allora  $R = R_2$  e se  $R_2 \subseteq R_1$ , allora  $R = R_1$

Esercizio proposto: generalizzare la definizione richiedendo che le relazioni abbiano lo stesso grado ed i domini di attributi corrispondenti siano uguali o compatibili

# UNIONE - ESEMPIO

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
a	b	3
d	a	5
c	b	27

R<sub>1</sub>

R<sub>2</sub>

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
a	b	3
j	a	5

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
j	a	5
a	b	3
d	a	5
c	b	27

R<sub>2</sub> ∪ R<sub>1</sub>

perché le relazioni sono  
insiemi anche se le  
rappresentiamo come  
tabelle le righe non hanno  
un ordine

R<sub>1</sub> ∪ R<sub>2</sub>

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
a	b	3
d	a	5
c	b	27
j	a	5

In SQL come si  
esprimono?  
Con UNION

**SONO  
UGUALI**

# DIFFERENZA

Date due relazioni  $R_1$  e  $R_2$  con nomi di attributi  $A_1, A_2, \dots, A_n$  e dominio  $dom$

La **differenza**  $R = R_1 - R_2$

- ha nomi di attributi  $A_1, A_2, \dots, A_n$   
 $\Rightarrow$  **Rango di  $R$  = Rango di  $R_1$  = Rango di  $R_2$**
  - Dominio  $dom$
  - una tupla  $t$  appartiene ad  $R \Leftrightarrow$ 
    - $t$  appartiene ad  $R_1$
    - e**
    - $t$  **non** appartiene ad  $R_2$
- $\Rightarrow$  **Cardinalità di  $R$   $\leq$  Cardinalità di  $R_1$**
- tuple che compaiono in entrambi vengono eliminate
  - caso degenero: se  $R_1 \cap R_2 = \emptyset$ , allora  $R = R_1$



# DIFFERENZA – ESEMPIO

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
a	b	3
d	a	5
c	b	27

R<sub>1</sub>

R<sub>2</sub>

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
a	b	3
j	a	5

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
j	a	5

R<sub>2</sub> - R<sub>1</sub>

R<sub>1</sub> - R<sub>2</sub>

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
d	a	5
c	b	27

In SQL come si esprimono?  
Con EXCEPT

# INTERSEZIONE

L'intersezione di due relazioni R, S è definita da

$$R \cap S = R - (R - S)$$

- Semantica statica: R e S devono avere lo stesso schema (per definizione di differenza) con stessi domini
- La relazione risultato ha lo stesso schema e lo stesso dominio delle relazioni argomento
  - Grado di  $R \cap S$  = Grado di R = Grado di S
- La semantica è l'insieme delle tuple contenute in R e in S

# INTERSEZIONE - ESEMPIO

R

A	B	C
a	b	c
d	a	f
c	b	d

S

A	B	C
d	a	f
b	g	a

In SQL come si  
esprimono?  
Con INTERSECT

A	B	C
a	b	c
c	b	d

R - S

A	B	C
d	a	f

$$R \cap S = R - (R - S)$$

# INTERSEZIONE - ESEMPIO

anno	genere
1990	fantastico
1993	animazione
1994	drammatico
1996	fantascienza
1999	horror
2003	fantastico
2005	animazione
2005	fantastico

$R$

anno	genere
2003	drammatico
1997	fantascienza
1991	commedia

$S$

anno	genere
------	--------

$R \cap S$

# INTERSEZIONE – ESEMPIO

genere

---

fantastico

animazione

drammatico

fantascienza

horror

$$GR = \Pi_{\text{genere}}(R)$$

genere

---

drammatico

fantascienza

commedia

$$GS = \Pi_{\text{genere}}(S)$$

genere

---

fantastico

animazione

drammatico

fantascienza

horror

commedia

$$GR \cup GS$$

genere

---

fantastico

animazione

horror

$$GR - GS$$

genere

---

drammatico

fantascienza

$$GR \cap GS$$

# JOIN (OPERAZIONE DERIVATA, DIVERSE VARIANTI)



30

# THETA – JOIN

[Theta-] Join di due relazioni R ed S

$$R \bowtie_{A\theta A'} S = \sigma_{A\theta A'} (R \times S)$$

- **predicato di join**  $A\theta A'$ 
  - $\theta$  è un operatore relazionale
  - $A \in U_R$  e  $A' \in U_S$
- Semantica statica
  - $U_R \cap U_S = \emptyset$  (per la correttezza del prodotto cartesiano)
- Semantica dinamica: derivata dalla definizione
  - $\text{Rango di } R \bowtie_{A\theta A'} S = \text{Rango di } R + \text{Rango di } S$
  - $\text{Cardinalità di } R \bowtie_{A\theta A'} S \leq \text{Cardinalità di } R \times \text{Cardinalità di } S$
- Se  $\theta$  è un'uguaglianza il theta-join prende il nome di **equi-join**

# JOIN – ESEMPIO

In SQL come si esprimono?

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
a	b	3
d	a	5
c	b	27

R

S

B <sub>1</sub>	B <sub>2</sub>
1	a
6	k

B <sub>1</sub>	B <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
1	a	a	b	3
6	k	a	b	3
6	k	d	a	5
6	k	c	b	27

$S \bowtie_{B_2 \geq A_1} R$

SELECT DISTINCT \*  
FROM S JOIN R ON  
B2>=A1

$R \bowtie_{A_3 < B_1} S$

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	B <sub>1</sub>	B <sub>2</sub>
a	b	3	1	a
a	b	3	6	k
d	a	5	1	a
d	a	5	6	k
c	b	27	1	a
c	b	27	6	k

SELECT DISTINCT \*  
FROM R JOIN S ON  
A3<B1



Sintatticamente  
corretto, ma  
rappresenta qualcosa di  
significativo?

## JOIN - ESEMPIO

codCli	nome
6610	anna
6635	paola
6642	marco

titolo	anno
pulp fiction	1994
le iene	1992

$\Pi_{\text{codCli, nome}}(\text{Cliente})$        $\Pi_{\text{titolo, anno}}(\sigma_{\text{regista}='quentin tarantino'}(\text{Film}))$

codCli	nome	titolo	anno
6610	anna	pulp fiction	1994
6635	paola	pulp fiction	1994
6642	marco	pulp fiction	1994
6610	anna	le iene	1992

$\Pi_{\text{codCli, nome}}(\text{Cliente}) \bowtie_{\text{nome} < \text{titolo}} \Pi_{\text{titolo, anno}}(\sigma_{\text{regista}='quentin tarantino'}(\text{Film}))$

# EQUI-JOIN – ESEMPIO

- Determinare il **titolo ed il regista** dei film **noleggiati il 15 Marzo 2006** dal cliente di codice 6635

$\sigma_{\text{codCli} = 6635 \wedge \text{dataNol} = '15\text{-Mar-2006'}$  (Noleggio)

$\rho_{\text{colloc}} \leftarrow c(\text{Video})$

<u>c</u> colloc	titolo	regista	tipo
1111	underground	emir kusturica	v
1112	underground	emir kusturica	d
1113	big fish	tim burton	v
1114	big fish	tim burton	d
1115	edward mani di forbice	tim burton	d
1116	nightmare before christmas	tim burton	v
1117	nightmare before christmas	tim burton	d
1118	ed wood	tim burton	d
1119	mars attacks	tim burton	d
1120	il mistero di sleepy hollow	tim burton	d
1121	la sposa cadavere	tim burton	d
1122	la fabbrica di cioccolato	tim burton	d
1123	la fabbrica di cioccolato	tim burton	d
1124	io non ho paura	gabriele salvatores	d
1125	nirvana	gabriele salvatores	d
1126	mediterraneo	gabriele salvatores	d
1127	pulp fiction	quentin tarantino	v
1128	pulp fiction	quentin tarantino	d
1129	le iene	quentin tarantino	d

colloc	dataNol	codCli	dataRest
<del>1111</del>	<del>01-Mar-2006</del>	<del>6635</del>	<del>02-Mar-2006</del>
<del>1115</del>	<del>01-Mar-2006</del>	<del>6635</del>	<del>02-Mar-2006</del>
<del>1117</del>	<del>02-Mar-2006</del>	<del>6635</del>	<del>06-Mar-2006</del>
<del>1118</del>	<del>02-Mar-2006</del>	<del>6635</del>	<del>06-Mar-2006</del>
<del>1111</del>	<del>04-Mar-2006</del>	<del>6642</del>	<del>05-Mar-2006</del>
<del>1119</del>	<del>08-Mar-2006</del>	<del>6635</del>	<del>10-Mar-2006</del>
<del>1120</del>	<del>08-Mar-2006</del>	<del>6635</del>	<del>10-Mar-2006</del>
<del>1116</del>	<del>08-Mar-2006</del>	<del>6642</del>	<del>09-Mar-2006</del>
<del>1118</del>	<del>10-Mar-2006</del>	<del>6642</del>	<del>11-Mar-2006</del>
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
<del>1119</del>	<del>15-Mar-2006</del>	<del>6642</del>	<del>16-Mar-2006</del>
<del>1120</del>	<del>15-Mar-2006</del>	<del>6610</del>	<del>16-Mar-2006</del>
<del>1112</del>	<del>16-Mar-2006</del>	<del>6610</del>	<del>16-Mar-2006</del>
<del>1114</del>	<del>16-Mar-2006</del>	<del>6610</del>	<del>17-Mar-2006</del>
<del>1120</del>	<del>16-Mar-2006</del>	<del>6642</del>	<del>20-Mar-2006</del>
<del>1124</del>	<del>20-Mar-2006</del>	<del>6610</del>	<del>21-Mar-2006</del>
<del>1115</del>	<del>20-Mar-2006</del>	<del>6610</del>	<del>21-Mar-2006</del>
<del>1124</del>	<del>21-Mar-2006</del>	<del>6642</del>	<del>22-Mar-2006</del>
<del>1116</del>	<del>21-Mar-2006</del>	<del>6610</del>	<del>?</del>
<del>1117</del>	<del>21-Mar-2006</del>	<del>6610</del>	<del>?</del>
<del>1127</del>	<del>22-Mar-2006</del>	<del>6635</del>	<del>?</del>
<del>1125</del>	<del>22-Mar-2006</del>	<del>6635</del>	<del>?</del>
<del>1122</del>	<del>22-Mar-2006</del>	<del>6642</del>	<del>?</del>
<del>1113</del>	<del>22-Mar-2006</del>	<del>6642</del>	<del>?</del>

# EQUI-JOIN – ESEMPIO

- Determinare il **titolo ed il regista** dei film **noleggiati** il 15 Marzo 2006 dal cliente di codice 6635

$\sigma_{\text{codCli} = 6635 \wedge \text{dataNol} = '15\text{-Mar-2006'} \text{ (Noleggio)}} \bowtie_{\text{colloc} = c} \rho_{\text{colloc} \leftarrow c} \text{(Video)}$

c	titolo	regista	tipo	colloc	dataNol	codCli	dataRest
1111	underground	emir kusturica	v	1121	15-Mar-2006	6635	18-Mar-2006
1112	underground	emir kusturica	d	1122	15-Mar-2006	6635	18-Mar-2006
1113	big fish	tim burton	v	1113	15-Mar-2006	6635	18-Mar-2006
1114	big fish	tim burton	d	1129	15-Mar-2006	6635	20-Mar-2006
1115	edward mani di forbice	tim burton	d				
1116	nightmare before christmas	tim burton	v				
1117	nightmare before christmas	tim burton	d				
1118	ed wood	tim burton	d				
1119	mars attacks	tim burton	d				
1120	il mistero di sleepy hollow	tim burton	d				
1121	la sposa cadavere	tim burton	d				
1122	la fabbrica di cioccolato	tim burton	d				
1123	la fabbrica di cioccolato	tim burton	d				
1124	io non ho paura	gabriele salvatores	d				
1125	nirvana	gabriele salvatores	d				
1126	mediterraneo	gabriele salvatores	d				
1127	pulp fiction	quentin tarantino	v				
1128	pulp fiction	quentin tarantino	d				
1129	le iene	quentin tarantino	d				

# EQUI-JOIN - ESEMPIO

- Determinare il **titolo ed il regista** dei film noleggiati il 15 Marzo 2006 dal cliente di codice 6635

$\Pi_{\text{titolo, regista}}(\text{Video})$

$\sigma_{\text{codCli} = 6635 \wedge \text{dataNol} = '15\text{-Mar-2006'}}(\text{Noleggio}) \bowtie_{\text{colloc} = c} \rho_{\text{colloc} \leftarrow c}(\text{Video})$

c	titolo	regista	tipo	colloc	dataNol	codCli	dataRest
1121	la sposa cadavere	tim burton	c	1121	15-Mar-2006	6635	18-Mar-2006
1122	la fabbrica di cioccolato	tim burton	c	1122	15-Mar-2006	6635	18-Mar-2006
1113	big fish	tim burton	v	1113	15-Mar-2006	6635	18-Mar-2006
1129	le iene	quentin tarantino	c	1129	15-Mar-2006	6635	20-Mar-2006

titolo	regista
la sposa cadavere	tim burton
la fabbrica di cioccolato	tim burton
big fish	tim burton
le iene	quentin tarantino

# JOIN NATURALE $\bowtie$

- Una ‘semplificazione’ del join
  - si ‘*incollano*’ le colonne **con lo stesso nome**
  - non serve ridenominare
  - non bisogna specificare il predicato di join
  - non si hanno colonne duplicate nel risultato
- Ha senso solo nella notazione **con nome** del modello relazionale
- È più generale: corrisponde ad una **serie** di equi-join

## JOIN NATURALE - ESEMPIO

R

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

S

B	C	D
b	c	d
b	c	e
a	d	b

$R \bowtie S$

A	B	C	D
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

Nessuna corrispondenza  
Non contribuirà al risultato  
finale In SQL come si  
esprime?

SELECT DISTINCT  
FROM R NATURAL  
JOIN S

# JOIN NATURALE – IDEA INFORMALE

Per fare  $R \bowtie S$

1. in  $S$  si ridenominano le colonne con lo stesso nome di colonne di  $R$ 
  - usando nomi nuovi e temporanei (quindi irrilevanti)
2. si fa il prodotto cartesiano
3. si selezionano solo le tuple in cui le colonne di  $R$  comuni a  $S$  coincidono con le rispettive colonne di  $S$  ridenominate
4. si proietta eliminando le colonne ridenominate

# JOIN NATURALE

Date due relazioni

- $R_1$  con attributi  $C^1_1, \dots, C^1_n$  e dominio  $\text{dom}_1$
- $R_2$  con attributi  $C^2_1, \dots, C^2_z$  e dominio  $\text{dom}_2$

siano

- $\{A_1, \dots, A_k\} = U_R \cap U_S$  i nomi di attributi **comuni** a  $R_1$  e  $R_2$
- $\{B_1, \dots, B_k\}$  nomi '**nuovi**' (cioè  $\{B_1, \dots, B_k\} \cap (U_R \cup U_S) = \emptyset$ )

$$R_1 \bowtie R_2 = \Pi_{\text{schema}} \sigma_F (R_1 \times \rho_{A_1, \dots, A_k \leftarrow B_1, \dots, B_k} (R_2))$$

$\text{schema} = C^1_1, \dots, C^1_n, C_{n+1}, \dots, C_m$

- $C^1_1, \dots, C^1_n$  sono gli attributi di  $R_1$
- $C_{n+1}, \dots, C_m = \text{purge}(C^2_1, \dots, C^2_z, \{A_1, \dots, A_k\})$ 
  - $\text{purge}$  è la funzione che toglie da una lista tutte le occorrenze di elementi in un insieme
  - sono **gli attributi di  $R_2$  che non compaiono in  $R_1$**

$F$  è la formula  
 $A_1 = B_1 \wedge A_2 = B_2 \wedge \dots \wedge A_k = B_k$



# JOIN NATURALE

Per come è definito il join naturale

- $\text{dom}_1(A_i) = \text{dom}_2(A_i)$  affinché sia corretto  $\rho$
- il dominio del risultato è
$$\text{dom}(C_i^1) = \text{dom}_1(C_i^1) \text{ e } \text{dom}(C_i) = \text{dom}_2(C_i)$$
- lo schema è quello di  $R_1$  seguito da quello di  $R_2$  in cui sono stati *cancellati* gli attributi che già compaiono in  $R_1$
- le tuple sono la concatenazione di una tupla di  $R_1$  con una di  $R_2$  che coincide con la prima sugli attributi in comune, da cui sono stati *cancellati* i valori di tali attributi

# JOIN NATURALE

Casi particolari di  $R \bowtie S$

- $U_R = U_S$ 
  - Join naturale = intersezione
- $U_R \cap U_S = \emptyset$ 
  - Join naturale = prodotto cartesiano

Se si usano gli stessi nome per gli attributi in chiave e chiave esterna il join naturale permette di navigare in modo naturale le associazioni

## JOIN NATURALE - ESEMPIO

- Individuare codice e nome dei clienti e titolo e anno dei film di Quentin Tarantino da lui noleggiati

$\Pi_{\text{codCli, nome, titolo, anno}}(\text{Cliente} \bowtie \text{Noleggio} \bowtie \text{Video} \bowtie \sigma_{\text{regista}='quentin tarantino'}(\text{Film}))$

codCli	nome	titolo	anno
6635	paola	le iene	1992
6635	paola	pulp fiction	1994
6642	marco	pulp fiction	1994

# DIVISIONE

44

# DIVISIONE

- Si chiama così perché è *un'inversa* del prodotto cartesiano

$$R = (R \times S) \div S$$

- Serve a selezionare le tuple che su un sottoinsieme degli attributi (il divisore) compaiono con tutte le possibili combinazioni di valori
  - i film che sono disponibili in ogni formato video
  - i clienti che hanno affittato tutti i film di un certo regista
- Si definisce togliendo le tuple per cui manca almeno un completamento

# DIVISIONE

In realtà la definizione è corretta solo se  $S$  ha schema  $A_n, \dots, A_k$  per qualche  $n \in (1, k)$

Esercizio proposto: correggere la formula definitoria (hint: riordinare gli attributi di  $R$ )

Date due relazioni

- $R$  con attributi  $A_1, \dots, A_k$ ,
- $S$  tale che  $U_R \supset U_S$

Detto  $D = \text{purge}(A_1, \dots, A_k, U_S)$  i nomi di attributi in  $A_1, \dots, A_k$  che **non** compaiono in  $S$

$$R \div S = \Pi_D(R) - \Pi_D((\Pi_D(R) \times S) - R)$$

tuple su  $D$  che hanno tutti i possibili completamenti in  $S$

tuple su  $D$  completate in tutti i modi possibili rispetto a  $S$

tuple su  $D$  per cui manca almeno un completamento in  $S$

In altre parole

- $t \in R \div S$  se e solo se  $\forall s \in S \exists r_s \in R$  tale che  $r_s[U_R - U_S] = t$  e  $r_s[U_S] = s$
- $t \in R \div S$  se e solo se  $\{t\} \times S \subseteq R$ 
  - usando gli isomorfismi di associatività del prodotto cartesiano per fare quadrare gli schemi

# DIVISIONE - ESEMPIO

	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
✓	k	a	a
✓	j	a	b
✓	j	a	a
	k	b	b
	m	c	b
✓	m	a	a
✓	k	a	b

R

A <sub>1</sub>
k
j

$R \div S$

Manca per m

A <sub>2</sub>	A <sub>3</sub>
a	a
a	b

S

In SQL come si esprime?

Non c'è un costrutto *ad hoc*

Si può usare l'espressione di definizione

Spesso ci sono modi più furbi di ottenere lo stesso risultato usando il potere espressivo ulteriore di SQL (operazioni sui dati, count, group by...)

# DIVISIONE – ESEMPIO

Determinare i codici dei clienti che hanno noleggiato tutti i film di Tim Burton

## 1. Film di Tim Burton

$$TB = \Pi_{\text{titolo, regista}}(\sigma_{\text{regista}='tim\ burton'}(\text{Film}))$$

titolo	regista
edward mani di forbice	tim burton
nightmare before christmas	tim burton
ed wood	tim burton
mars attacks	tim burton
il mistero di sleepy hollow	tim burton
big fish	tim burton
la sposa cadavere	tim burton
la fabbrica di cioccolato	tim burton



# DIVISIONE – ESEMPIO

## 2. Film noleggiati dai clienti

$$NC = \Pi_{\text{codCli}, \text{titolo}, \text{regista}}(\text{Noleggio} \bowtie \text{Video})$$

codCli	titolo	regista
6635	underground	emir kusturica
6635	edward mani di forbice	tim burton
6635	nightmare before christmas	tim burton
6635	ed wood	tim burton
6642	underground	emir kusturica
6635	mars attacks	tim burton
6635	il mistero di sleepy hollow	tim burton
6642	nightmare before christmas	tim burton
6642	ed wood	tim burton
6635	la sposa cadavere	tim burton
6635	la fabbrica di cioccolato	tim burton
6635	big fish	tim burton
6635	le iene	quentin tarantino
6642	mars attacks	tim burton
6610	mediterraneo	gabriele salvatores
6610	underground	emir kusturica
6610	big fish	tim burton
6642	pulp fiction	quentin tarantino
6610	io non ho paura	gabriele salvatores
6610	edward mani di forbice	tim burton
6642	io non ho paura	gabriele salvatores
6610	nightmare before christmas	tim burton
6635	pulp fiction	quentin tarantino
6635	nirvana	gabriele salvatores
6642	la fabbrica di cioccolato	tim burton
6642	big fish	tim burton

# DIVISIONE – ESEMPIO

codCli	titolo	regista
6635	underground	emir kusturica
6635	edward mani di forbice	tim burton
6635	nightmare before christmas	tim burton
6635	ed wood	tim burton
6642	underground	emir kusturica
6635	mars attacks	tim burton
6635	il mistero di sleepy hollow	tim burton
6642	nightmare before christmas	tim burton
6642	ed wood	tim burton
6635	la sposa cadavere	tim burton
6635	la fabbrica di cioccolato	tim burton
6635	big fish	tim burton
6635	le iene	quentin tarantino
6642	mars attacks	tim burton
6610	mediterraneo	gabriele salvatores
6610	underground	emir kusturica
6610	big fish	tim burton
6642	pulp fiction	quentin tarantino
6610	io non ho paura	gabriele salvatores
6610	edward mani	tim burton
6642	io non ho paura	gabriele salvatores
6610	nightmare before christmas	tim burton
6635	pulp fiction	quentin tarantino
6635	nirvana	gabriele salvatores
6642	la fabbrica di cioccolato	tim burton
6642	big fish	tim burton

Manca  
→ 6642 non sarà nel risultato

Manca  
→ 6610 non sarà nel risultato

NC ÷ TB

# ESEMPI

51

# ALGEBRA RELAZIONALE – ESEMPIO

Selezionare i codici dei clienti che hanno noleggiato film **drammatici**

1. Codici clienti che hanno noleggiato almeno un film drammatico

$$R = \Pi_{\text{codCli}}(\text{Noleggio} \bowtie \text{Video} \bowtie \sigma_{\text{genere}='drammatico'}(\text{Film}))$$

Selezionare i codici dei clienti che hanno noleggiato **sia** film drammatici **che** film horror

1. Codici clienti che hanno noleggiato almeno un film drammatico: fatto in R
2. Codici clienti che hanno noleggiato almeno un film horror

$$S = \Pi_{\text{codCli}}(\text{Noleggio} \bowtie \text{Video} \bowtie \sigma_{\text{genere}='horror'}(\text{Film}))$$

3. Risultato: **R**  $\cap$  **S**

# ALGEBRA RELAZIONALE - ESEMPIO

Selezionare i codici dei clienti che non hanno **mai** noleggiato film horror

1. Codici di tutti i clienti

$$R = \Pi_{\text{codCli}}(\text{Cliente})$$

2. Codici clienti che hanno noleggiato almeno un film horror

$$S = \Pi_{\text{codCli}}(\text{Noleggio} \bowtie \text{Video} \bowtie \sigma_{\text{genere}='horror'}(\text{Film}))$$

3. Risultato: **R - S**

# ALGEBRA RELAZIONALE – ESEMPIO

Selezionare i codici dei clienti che hanno noleggiato film di almeno due generi diversi

1. Coppie cliente/genere film noleggiati

$$R = \Pi_{\text{codCli, genere}} (\text{Noleggio} \bowtie \text{Video} \bowtie \text{Film})$$

2. Coppie di coppie cliente/genere film noleggiati

$$S = \rho_{\text{codCli, genere} \leftarrow c, g} (R) \times R$$

3. Selezione delle coppie relative a stesso cliente e generi diversi

$$T = \sigma_{\text{codCli} = c \wedge \text{genere} \neq g} (S)$$

4. Proiezione sul codice cliente

$$\Pi_{\text{codCli}}(T)$$

Ci sono **tutte** le coppie anche quelle relative a clienti diversi o stesso cliente e stesso genere

Analogamente usando il join naturale

$$\Pi_{\text{codCli}}(\sigma_{\text{genere} \neq g} (\rho_{\text{genere} \leftarrow g} (R) \bowtie R)) = \mathbf{T????}$$

# ALGEBRA RELAZIONALE - ESEMPIO

Selezionare titolo e regista del film con la valutazione più alta

1. Titolo e regista di tutti i film

$$R = \Pi_{\text{titolo, regista}}(\text{Film})$$

1. Titolo, regista e valutazione di tutti i film

$$S = \Pi_{\text{titolo, regista, valutaz}}(\text{Film})$$

3. Coppie di tuple in S in cui la valutazione del primo è (strettamente) minore di quella del secondo

$$T = S \bowtie_{\text{valutaz} < v} \rho_{\text{titolo, regista, valutaz} \leftarrow t, r, v}(S)$$

4. Proiezione su titolo e regista

$$H = \Pi_{\text{titolo, regista}}(T)$$

titolo e regista (il primo)  
vanno esclusi dal risultato

5. Risultato **R-H**

# ALGEBRA RELAZIONALE – SINTESI

Op.	Funzionalità	Cond.	Semantica
$\Pi_A$	$\mathcal{R}(U) \rightarrow \mathcal{R}(A)$	$A \subseteq U$	$\Pi_A(R) = \{t[A] \mid t \in R\}$
$\sigma_F$	$\mathcal{R}(U) \rightarrow \mathcal{R}(U)$	$A(F) \subseteq U$	$\sigma_F(R) = \{t \mid t \in R \wedge F(t)\}$
$\times$	$\mathcal{R}(U) \times \mathcal{R}(V) \rightarrow \mathcal{R}(U \cup V)$	$U \cap V = \emptyset$	$R_1 \times R_2 = \{t_1 \cdot t_2 \mid t_1 \in R_1 \wedge t_2 \in R_2\}$
$\cup$	$\mathcal{R}(U) \times \mathcal{R}(U) \rightarrow \mathcal{R}(U)$		$R_1 \cup R_2 = \{t \mid t \in R_1 \vee t \in R_2\}$
$-$	$\mathcal{R}(U) \times \mathcal{R}(U) \rightarrow \mathcal{R}(U)$		$R_1 - R_2 = \{t \mid t \in R_1 \wedge t \notin R_2\}$
$\cap$	$\mathcal{R}(U) \times \mathcal{R}(U) \rightarrow \mathcal{R}(U)$		$R_1 \cap R_2 = \{t \mid t \in R_1 \wedge t \in R_2\}$
$\bowtie_F$	$\mathcal{R}(U) \times \mathcal{R}(V) \rightarrow \mathcal{R}(U \cup V)$	$U \cap V = \emptyset$	$R_1 \bowtie_F R_2 = \{t_1 \cdot t_2 \mid t_1 \in R_1 \wedge t_2 \in R_2 \wedge F(t_1 \cdot t_2)\}$
$\bowtie$	$\mathcal{R}(U) \times \mathcal{R}(V) \rightarrow \mathcal{R}(U \cup V)$		$R_1 \bowtie R_2 = \{t \mid t[U] \in R_1 \wedge t[V] \in R_2\}$
$\div$	$\mathcal{R}(U) \times \mathcal{R}(V) \rightarrow \mathcal{R}(U \setminus V)$	$V \subset U$	$R_1 \div R_2 = \{t \mid \forall t_2 \in R_2 \exists t_1 \in R_1 \text{ t.c. } t_1[U \setminus V] = t, t_1[V] = t_2\}$