

# Operatori derivati dell'algebra relazionale

Prof. Alfredo Pulvirenti  
Prof. Salvatore Alaimo

(Atzeni-Ceri Capitolo 3)

# JOIN incompleti

- Nel caso in cui alcuni valori tra gli attributi comuni non coincidono

$r_1$

Employee	Department
Smith	sales
Black	production
White	production

$r_2$

Department	Head
production	Mori
purchasing	Brown

$r_1 \bowtie r_2$

Employee	Department	Head
Black	production	Mori
White	production	Mori

Quindi, alcune n-uple non partecipano al JOIN (*dangling* n-uple)

## JOIN vuoti, un caso estremo

- Potrebbe anche succedere che nessuna n-upla trovi il corrispettivo

$r_1$

Employee	Department
Smith	sales
Black	production
White	production

$r_2$

Department	Head
marketing	Mori
purchasing	Brown

$r_1 \bowtie r_2$

Employee	Department	Head

# L'altro caso estremo del JOIN

- Ogni n-upla di  $R_1$  si combina con ogni n-upla di  $R_2$

$r_1$	Employee	Project
	Smith	A
	Black	A
	White	A

$r_2$	Project	Head
	A	Mori
	A	Brown

$r_1 \bowtie r_2$	Employee	Project	Head
	Smith	A	Mori
	Black	A	Brown
	White	A	Mori
	Smith	A	Brown
	Black	A	Mori
	White	A	Brown

- Cardinalità del risultato è il prodotto delle cardinalità

## OUTER JOIN (Giunzione esterna)

- Una variante del JOIN per mantenere nel risultato le n-uple che non partecipano al JOIN
- Gli attributi delle *dangling* n-uple vengono riempiti con NULL
- Tre varianti:
  - **Left:** solo dangling n-uple del primo operando
  - **Right:** solo dangling n-uple del secondo operando
  - **Full:** n-uple da entrambi gli operandi

# Giunzione Esterna

- La giunzione esterna è la giunzione naturale estesa con tutte le n-uple che non appartengono alla giunzione naturale, completate con valori NULL per gli attributi mancanti.
- Siano R ed S definite sugli insiemi di attributi XY e YZ rispettivamente.

$$\begin{aligned} R \overleftrightarrow{\bowtie} S = & (R \bowtie S) \cup \\ & ((R - \pi_{XY}(R \bowtie S)) \times \{Z = NULL\}) \cup \\ & (\{X = NULL\} \times (S - \pi_{YZ}(R \bowtie S))) \end{aligned}$$

## Altre Giunzioni Esterne

- Nelle giunzioni esterne sinistre e destre si aggiungono solo le parti sinistre e destre.
- Siano  $R$  ed  $S$  definite sugli insiemi di attributi  $XY$  e  $YZ$  rispettivamente.

- Definiamo Giunzione Esterna Sinistra:

$$R \overleftarrow{\bowtie} S = (R \bowtie S) \cup ((R - \pi_{XY}(R \bowtie S)) \times \{Z = NULL\})$$

- Definiamo Giunzione Esterna Destra:

$$R \overrightarrow{\bowtie} S = (R \bowtie S) \cup (\{X = NULL\} \times (S - \pi_{YZ}(R \bowtie S)))$$

# Esempio di NATURAL OUTER JOIN

$r_1$

Employee	Department
Smith	sales
Black	production
White	production

$r_2$

Department	Head
production	Mori
purchasing	Brown

$r_1 \bowtie_{\text{LEFT}} r_2$

Employee	Department	Head
Smith	Sales	NULL
Black	production	Mori
White	production	Mori

$r_1 \bowtie_{\text{RIGHT}} r_2$

Employee	Department	Head
Black	production	Mori
White	production	Mori
NULL	purchasing	Brown

$r_1 \bowtie_{\text{FULL}} r_2$

Employee	Department	Head
Smith	Sales	NULL
Black	production	Mori
White	production	Mori
NULL	purchasing	Brown



# Proprieta' del JOIN

- Il JOIN e'
  - Commutativo:  $R \bowtie S = S \bowtie R$
  - Associativo:  $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
- Quindi possiamo avere sequenze di JOIN senza rischio di ambiguita:

$$R \bowtie S \bowtie T \dots$$

# Esempio di JOIN multipli

$r_1$

Employee	Department
Smith	sales
Black	production
Brown	marketing
White	production

$r_2$

Department	Division
production	A
marketing	B
purchasing	B

$r_2$

Division	Head
A	Mori
B	Brown

$r_1 \bowtie r_2 \bowtie r_3$

Employee	Department	Division	Head
Black	production	A	Mori
Brown	marketing	B	Brown
White	production	A	Mori

## Prodotto cartesiano a partire dal JOIN

- Il JOIN è definito anche se non ci sono attributi comuni fra le relazioni
- In questo caso, non essendoci vincoli sulle tuple da selezionare, vengono selezionate tutte le tuple dalle relazioni del JOIN e quindi **otteniamo un prodotto cartesiano**

# Esempio di prodotto cartesiano generato dal JOIN

Employees

Employee	Project
Smith	A
Black	A
Black	B

Projects

Code	Name
A	Venus
B	Mars

Employees ⋈ Projects

Employee	Project	Code	Name
Smith	A	A	Venus
Black	A	A	Venus
Black	B	A	Venus
Smith	A	B	Mars
Black	A	B	Mars
Black	B	B	Mars

# Intersezione a partire dalla Natural Join

- Dati due relazioni definite sulla stessa lista di attributi, allora il natural join coincide con l'intersezione delle due relazioni.

# Semi-giunzione(Semi-join)

- Siano  $R$  con attributi  $XY$  ed  $S$  con attributi  $YZ$
- $R \bowtie S$  è una relazione di attributi  $XY$  costituita da tutte le  $n$ -uple di  $R$  che partecipano a  $R \bowtie S$ .
- La semi-giunzione e' derivata perché

$$R \bowtie S = \pi_{XY}(R \bowtie S)$$

# Studenti × Esami

Nome	Matricola	Indirizzo	Telefono
Mario Rossi	123456	Via Etnea 1	222222
Ugo Bianchi	234567	Via Roma 2	333333
Teo Verdi	345678	Via Torino 3	444444

Corso	Matricola	Voto
Architettura	123456	30
Programmazione	234567	18
Architetture	234567	27

## Studenti × Esami

Nome	Matricola	Indirizzo	Telefono
Mario Rossi	123456	Via Etnea 1	222222
Ugo Bianchi	234567	Via Roma 2	333333

## Unione Esterna

- Siano  $R$  ed  $S$  due relazioni definite sugli insiemi di attributi  $XY$  e  $YZ$  allora
- L'unione esterna

$$R \overset{\leftrightarrow}{\cup} S =$$

$$R \times \{Z = NULL\} \cup \{X = NULL\} \times S$$

- si ottiene estendendo le due tabelle con le colonne dell'altra con valori nulli e si fa l'unione.



# Esempio di Unione Esterna

A	B	C	D
X1	Y	Z	X
X2	Y	Z	X
X3	Y	W	X
X4	Y	W	X

**R**

B	C	D	E
Y	Z	X	Y1
Y	Z	X	M1
Y	W	X	Y2
Y	W	X	M2

**S**

$R \cup S$

A	B	C	D	E
X1	Y	Z	X	NULL
X2	Y	Z	X	NULL
X3	Y	W	X	NULL
X4	Y	W	X	NULL
NULL	Y	Z	X	Y1
NULL	Y	Z	X	M1
NULL	Y	W	X	Y2
NULL	Y	W	X	M2

# Selezione con valori nulli

## Impiegati

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

$\sigma_{\text{Età} > 40}$  (Impiegati)

- la condizione atomica è vera solo per valori non nulli

# Un risultato non desiderabile

$$\sigma_{Età>30}(Persone) \cup \sigma_{Età\leq 30}(Persone) \neq Persone$$

- Perché?
  - Perché le selezioni vengono valutate separatamente!

- Ma anche

$$\sigma_{Età>30 \vee Età\leq 30}(Persone) \neq Persone$$

- Perché?
  - Perché anche le condizioni atomiche vengono valutate separatamente!

# Selezione con valori nulli: soluzione

$\sigma_{\text{Età}} > 40$  (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

- si potrebbe usare (ma non serve) una "logica a tre valori" (vero, falso, sconosciuto)

# LOGICA A 3 VALORI

$p$	$q$	$p \text{ and } q$	$p \text{ or } q$	$\text{not } p$
T	T	T	T	F
T	F	F	T	F
T	U	U	T	F
F	F	F	F	T
F	U	F	U	T
U	U	U	U	U

- Quindi:

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

=

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

=

*Persone*

## Impiegati

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

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

## Quoziente (divisione)

- **Divisione:** Siano  $XY$  gli attributi di  $R$  ed  $Y$  quelli di  $S$ , allora

$$R \div S = \{w \mid \{w\} \times S \subseteq R\}$$

### Esercizio

Dimostrare che il quoziente è un operatore derivato.



# Esempio

La divisione serve a rispondere a query del tipo:  
trova **TUTTE** le n-uple di  $R$  associate a **TUTTE** le  
n-uple di  $S$ .

Esempio

$$\{'DB,' PROG'\} = \pi_{corso}(\sigma_{corso='DB' \vee corso='PROG'}(Esami))$$

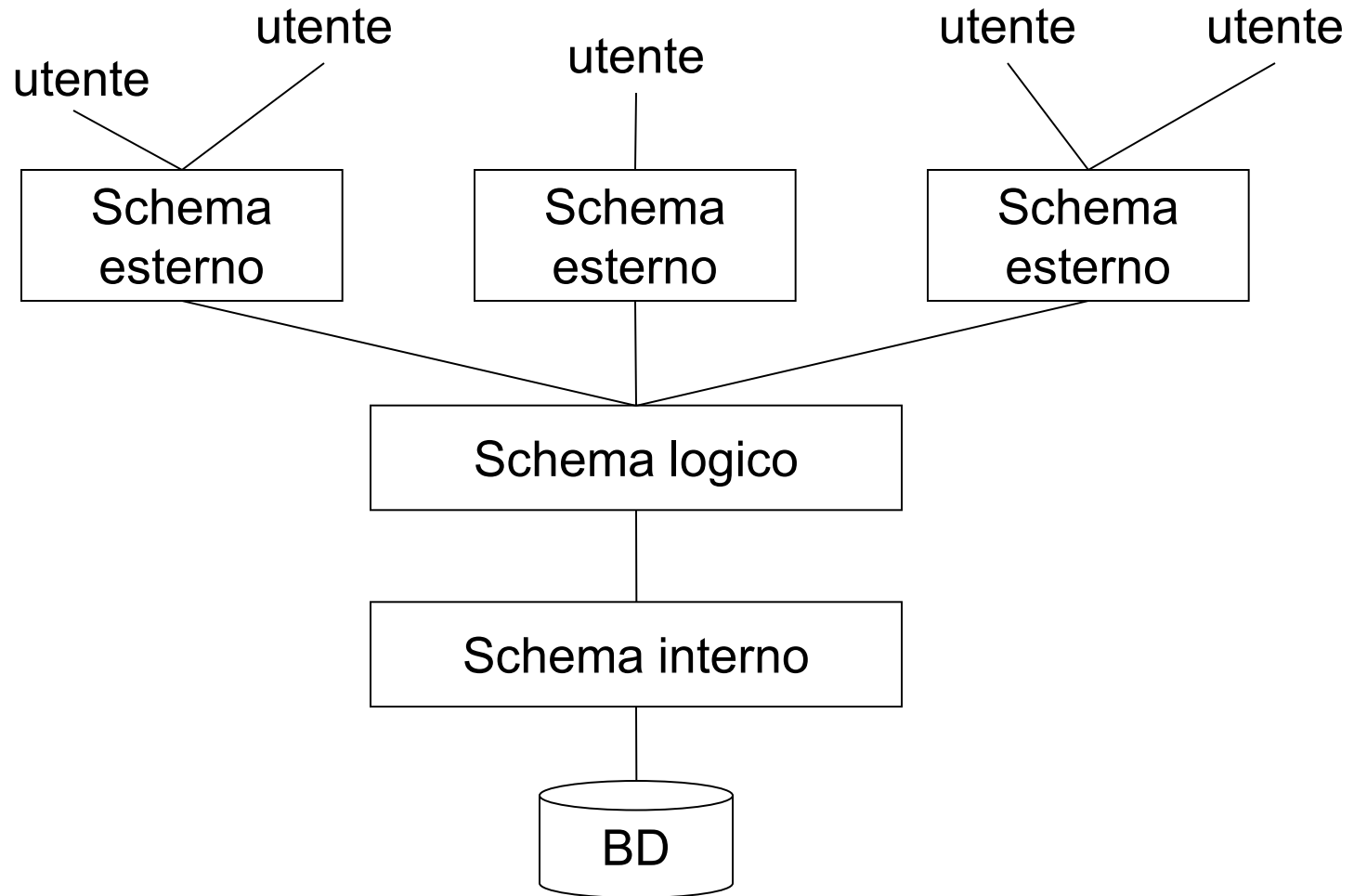
$$\pi_{matricola, corso}(Esami) \div \{'DB,' PROG'\}$$

Le matricole di studenti che hanno superato DB  
e PROG.

# 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

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



# Viste, esempio

Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Direzione

Reparto	Capo
A	Mori
B	Bruni

- Una vista:

$$\text{SUPERVISIONE} = \pi_{\text{Impiegato}, \text{Capo}}(\text{Afferenza} \bowtie \text{Direzione})$$

# Viste virtuali e materializzate

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

# 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

# Viste virtuali

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

# Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

$\sigma_{\text{Capo}='Leoni'}$  (Supervisione)

viene eseguita come

$\sigma_{\text{Capo}='Leoni'}(\pi_{\text{Impiegato, Capo}}(\text{Afferenza} \bowtie \text{Direzione}))$



# 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

# Viste come strumento di programmazione

- Trovare gli impiegati che hanno lo stesso capo di Rossi

- Senza vista:

$\pi_{\text{Impiegato}} ((\text{Afferenza} \bowtie \text{Direzione}) \bowtie \delta_{\text{ImpR,RepR} \leftarrow \text{Imp,Reparto}} (\sigma_{\text{Impiegato}='Rossi'} (\text{Afferenza} \bowtie \text{Direzione})))$

- Con la vista:

$\pi_{\text{Impiegato}} (\text{Supervisione} \bowtie \delta_{\text{ImpR,RepR} \leftarrow \text{Imp,Reparto}} (\sigma_{\text{Impiegato}='Rossi'} (\text{Supervisione})))$