

**Laurea in Informatica  
A.A. 2024-2025**

**Corso "Base di Dati"**

**L'Algebra Relazionale**

**Prof. Massimiliano de Leoni**



**UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA**

# Linguaggi per basi di dati

- Definizione dello schema (relazioni, attributi, chiavi, ecc...)
  - **DDL**: data definition language
- Operazioni sui dati: interrogazione. ("query") aggiornamento
  - **DML**: data manipulation language

# Linguaggi di interrogazione per basi di dati relazionali

- **Dichiarativi**
  - Specificano le proprietà del risultato ("che cosa")
  - Parzialmente dichiarativo: **SQL**
- **Procedurali**
  - Specificano le modalità di generazione del risultato ("come")
  - Vediamo oggi e domani: **Algebra Relazionale**

# Algebra relazionale

- Insieme di operatori
  - su relazioni
  - che producono relazioni
  - e possono essere composti

# Operatori insiemistici

- le relazioni sono insiemi
- i risultati debbono essere relazioni
- è possibile applicare **unione**, **intersezione**, **differenza** solo a relazioni definite sugli stessi attributi

# Unione

## Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

## Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

## Laureati $\cup$ Specialisti

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45
9297	Neri	33

# Intersezione

## Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

## Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

## Laureati $\cap$ Specialisti

Matricola	Nome	Età
7432	Neri	54
9824	Verdi	45

# Differenza

## Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

## Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

## Laureati – Specialisti

Matricola	Nome	Età
7274	Rossi	42



# Un'Unione Sensata ma Impossibile

## Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

## Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Paternità  $\cup$  Maternità

??

# Ridenominazione

(Operatore  $\rho$ )

"Modifica lo schema" lasciando inalterata l'istanza dell'operando

## Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

## $\rho_{\text{Genitore} \leftarrow \text{Padre}}$ (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

# Come fare con la «Unione Sensata ma (inizialmente) impossibile»

## Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

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

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

## Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

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

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

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

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

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



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

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

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

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

# Un esempio con molteplici ridenominazioni

Impiegati

Cognome	Ufficio	Stipendio
Rossi	Roma	55
Neri	Milano	64

Operai

Cognome	Fabbrica	Salario
Bruni	Monza	45
Verdi	Latina	55

$\rho$  Sede, Retribuzione  $\leftarrow$  Ufficio, Stipendio (Impiegati)  $\cup$   
 $\rho$  Sede, Retribuzione  $\leftarrow$  Fabbrica, Salario (Operai)

Cognome	Sede	Retribuzione
Rossi	Roma	55
Neri	Milano	64
Bruni	Monza	45
Verdi	Latina	55

# Selezione

(Operatore  $\sigma$ )

- Risultato:
  - stesso schema
  - sottoinsieme delle tuple
  - tuple che soddisfano una condizione

## Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Neri	Napoli	64

# Selezione

(Operatore  $\sigma$ )

- Risultato:
  - stesso schema
  - sottoinsieme delle tuple
  - tuple che soddisfano una condizione

Impiegati

$\sigma_{\text{Stipendio} > 60}$  (Impiegati)

Matricola	Cognome	Filiale	Stipendio
7898	Rossi	Roma	55
5998	Neri	Milano	64
8558	Milano	Milano	44
5698	Neri	Napoli	64

## Esempio: Impiegati che guadagnano più di 60 e lavorano a Milano

$\sigma_{\text{Stipendio} > 60 \text{ AND Filiale} = \text{'Milano'}} (\text{Impiegati})$

Impiegati

Matricola	Cognome	Filiale	Stipendio
7200	Dei	Roma	55
5998	Neri	Milano	64
8550	Milano	Milano	44
5000	Neri	Napoli	31



# Esempio: Impiegati che hanno lo stesso nome della filiale presso cui lavorano

$$\sigma_{\text{Cognome} = \text{Filiale}} (\text{Impiegati})$$

Impiegati

Matricola	Cognome	Filiale	Stipendio
7888	Rossi	Roma	55
5000	Noni	Milano	31
9553	Milano	Milano	44
5000	Noni	Napoli	31

# Selezione e proiezione

- Operatori "ortogonali"
- **Selezione**: Decomposizione Orizzontale
  - ➔ Eliminazione di Tuple
- **Proiezione**: Decomposizione Verticale
  - ➔ Eliminazione di Attributi/Colonne

# Proiezione

(Operatore  $\pi$ )

- Risultato:
  - ha parte degli attributi della relazione
  - sottoinsieme delle tuple (duplicati eliminati)

**Impiegati**

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

Duplicato da eliminare

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Neri	Napoli	64

# Esempio: Matricola e Cognome di Tutti gli impiegati

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

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Neri	Napoli	55
5998	Neri	Milano	64
9553	Rossi	Roma	44
5698	Rossi	Roma	64

# Esempio: Matricola e Cognome di Tutti gli impiegati

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

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Neri	Napoli	55
5998	Neri	Milano	64
9553	Rossi	Roma	44
5698	Rossi	Roma	64

# Cardinalità delle proiezioni

- Una proiezione contiene
  - contiene al più tante tuple quante l'operando,
  - ma può contenerne di meno
- Se  $X$  è una superchiave di  $R$ , allora  $\pi_X(R)$  contiene esattamente tante tuple quante  $R$

# Selezione e proiezione

- Combinando selezione e proiezione, possiamo estrarre interessanti informazioni da una relazione

# Esempio: matricola e cognome di quelli che guadagnano > 50

Impiegati  $\pi_{(Matricola, Cognome)} (\sigma_{Stipendio > 50} (Impiegati))$

Matricola	Cognome	Filiale	Stipendio
7309	Neri	Napoli	55
5998	Neri	Milano	64
8558	Rossi	Roma	64
5698	Rossi	Roma	64



# Join

(Operatore  $\bowtie$ )

- Operatore Binario su due relazioni A e B
- Risultato:
  - l'unione degli attributi degli operandi
  - tuple costruite come  $A \times B$  mantenendo quelle con valori uguali su attributi uguali

# Join: Example

Voto

Candidati

Numero	Voto
1	25
2	13
3	27
4	28

Numero	Candidato
1	Mario Rossi
2	Nicola Russo
3	Mario Bianchi
4	Remo Neri

Numero	Candidato	Voto
1	Mario Rossi	25
2	Nicola Russo	13
3	Mario Bianchi	27
4	Remo Neri	28

Voto ⋈ Candidati

# Join, sintassi e semantica

- Date due relazioni  $R_1(X_1)$ ,  $R_2(X_2)$
- $R_1 \bowtie R_2$  è una relaz. su  $X_1X_2$  (eq.  $X_1 \cup X_2$ )

$$\{ t \text{ su } X_1X_2 \mid t[X_1] \in R_1 \text{ e } t[X_2] \in R_2 \}$$

dove  $t[X_1]$  indica la proiezione su  $X_1$ , cioè  $\pi_{X_1}(R)$

# Join Completo

- Date due relazioni  $R_1(X_1)$ ,  $R_2(X_2)$
- $R_1 \bowtie R_2$  è una relaz. su  $X_1X_2$  (eq.  $X_1 \cup X_2$ )

$$\{ t \text{ su } X_1X_2 \mid t[X_1] \in R_1 \text{ e } t[X_2] \in R_2 \}$$

dove  $t[X_1]$  indica la proiezione su  $X_1$ , cioè  $\pi_{X_1}(R)$

# Join Completo

Impiegato	Reparto	Reparto	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

Ogni tupla contribuisce al risultato:  
la proiezione sugli attributi di ogni  
relazione R coinvolta restituisce R

## Join Non Completo

Impiegato	Reparto	Reparto	Capo
Rossi	A	B	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Qualche tupla non contribuisce al risultato:  
la proiezione sugli attributi di ogni  
relazione R coinvolta non restituisce R

# Un join vuoto

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
D	Mori
C	Bruni

Impiegato	Reparto	Capo
-----------	---------	------

# Un join con $n \times m$ tuple

Impiegato	Reparto
Rossi	B
Neri	B

Reparto	Capo
B	Mori
B	Bruni

Impiegato	Reparto	Capo
Rossi	B	Mori
Rossi	B	Bruni
Neri	B	Mori
Neri	B	Bruni



# Cardinalità del join

- $R_1(A,B)$  ,  $R_2(B,C)$

- In generale

$$0 \leq |R_1 \bowtie R_2| \leq |R_1| \times |R_2|$$

- se  $B$  è chiave in  $R_2$

$$0 \leq |R_1 \bowtie R_2| \leq |R_1|$$

- se  $B$  è chiave in  $R_2$  ed esiste vincolo di integrità referenziale fra  $B$  in  $R_1$  e  $R_2$ :

$$|R_1 \bowtie R_2| = |R_1|$$

se l'attributo  $B$  in  $R_1$  non può assumere valore nullo

## Il «problema» del join non completo

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

- Le tuple non contribuiscono al risultato: vengono "tagliate fuori"

# Join esterno

- Il join **esterno** estende, con valori nulli, le tuple che verrebbero tagliate fuori da un join (**interno**)
- esiste in tre versioni:
  - sinistro, destro, completo

# Join esterno

- **sinistro**: mantiene tutte le tuple del primo operando, estendendole con valori nulli, se necessario
- **destro**: ... del secondo operando ...
- **completo**: ... di entrambi gli operandi ...

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati ⋈<sub>LEFT</sub> Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Reparto	Capo
B	Mori
C	Bruni

## Impiegati $\bowtie_{\text{RIGHT}}$ Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Reparto	Capo
B	Mori
C	Bruni

## Impiegati $\bowtie_{\text{FULL}}$ Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

# Semijoin

- Operatore su due relazioni  $R_1(X_1)$ ,  $R_2(X_2)$
- Restituisce una relazione su  $X_1$ , con le tuple di  $R_1$  che contribuiscono al join con  $R_2$

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Reparto	Capo
B	Mori
C	Bruni

## Impiegati SEMI⋈ Reparti

Impiegato	Reparto
Neri	B
Bianchi	B



# Semijoin come Proiezione e Join

Date due  
relazioni

$R_1(A,B)$  ,  $R_2$   
 $(B,C)$

$R_1 \text{ SEMI} \bowtie R_2 =$   
 $\pi_{A,B} (R_1 \bowtie R_2)$

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Reparto	Capo
B	Mori
C	Bruni

## Impiegati SEMI $\bowtie$ Reparti

Impiegato	Reparto
Neri	B
Bianchi	B

# Prodotto cartesiano

- Un join naturale su relazioni  $R_1(X_1)$ ,  $R_2(X_2)$  senza attributi in comune ( $X_1 \cap X_2 = \emptyset$ ) coincide col prodotto cartesiano

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Codice	Capo
A	Mori
B	Bruni

## Impiegati ⋈ Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
Rossi	A	B	Bruni
Neri	B	A	Mori
Neri	B	B	Bruni
Bianchi	B	A	Mori
Bianchi	B	B	Bruni

# Theta-join

- Il prodotto cartesiano, in pratica, ha senso (quasi) solo se seguito da selezione:

$$\sigma_{\text{Condizione}} (R_1 \bowtie R_2)$$

- L'operazione viene chiamata **theta-join** e indicata con

$$R_1 \bowtie_{\text{Condizione}} R_2$$

AND e OR di atomi di confronto  $A_1 \vartheta A_2$  dove  $\vartheta$  è uno degli operatori di confronto ( $=, >, <, \dots$ )

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati ⋈<sub>Reparto=Codice</sub> Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
<del>Rossi</del>	<del>A</del>	<del>B</del>	<del>Bruni</del>
<del>Neri</del>	<del>B</del>	<del>A</del>	<del>Mori</del>
Neri	B	B	Bruni
<del>Bianchi</del>	<del>B</del>	<del>A</del>	<del>Mori</del>
Bianchi	B	B	Bruni

## Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

## Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati  $\bowtie_{\text{Reparto}=\text{Codice}}$  Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni

Se l'operatore di confronto nel theta-join è sempre l'uguaglianza (=) allora si parla di **equi-join**, il più interessante.

# Esempi: Basi di Dati di partenza

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Supervisione

Impiegato	Capo
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

# Esempio 1: matricola, nome ed età degli impiegati che guadagnano più di 40

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Supervisione

Impiegato	Capo
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

$\pi_{\text{Matricola, Nome, Età}} (\sigma_{\text{Stipendio} > 40}(\text{Impiegati}))$



## Esempio 2: Capi degli impiegati che guadagnano più di 40

Impiegati	Matricola	Nome	Età	Stipendio
	7309	Rossi	34	45
	5998	Bianchi	37	38
	9553	Neri	42	35
	5698	Bruni	43	42
	4076	Mori	45	50
	8123	Lupi	46	60

Supervisione	Impiegato	Capo
	7309	5698
	5998	5698
	9553	4076
	5698	4076
	4076	8123

$\pi_{\text{Capo}} (\text{Supervisione} \bowtie_{\text{Impiegato=Matricola}} (\sigma_{\text{Stipendio}>40}(\text{Impiegati})))$

# Esempio 3: Matricola, nome e stipendio degli impiegati che guadagnano più dei capi / 1

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Impiegato	Capo	Supervisione
7309	5698	
5998	5698	
9553	4076	
5698	4076	
4076	8123	

Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5998	Bianchi	37	38	5998	5698
9553	Neri	42	35	9553	4076
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123

Supervisione  $\bowtie$  Impiegato=Matricola Impiegati

# Esempio 3: Matricola, nome e stipendio degli impiegati che guadagnano più dei capi / 2

Impiegati	Matricola	Nome	Età	Stipendio	Impiegato	Capo	Supervisione
	7309	Rossi	34	45	7309	5698	
	5998	Bianchi	37	38	5998	5698	
	9553	Neri	42	35	9553	4076	
	5698	Bruni	43	42	5698	4076	
	4076	Mori	45	50	4076	8123	
	8123	Lupi	46	60			

$\rho_{\text{MatrC, NomeC, StipC, EtàC}} \leftarrow \text{Matr, Nome, Stip, Età} (\text{Impiegati})$

MatrC	NomeC	EtàC	StipC
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

# Esempio 3: Matricola, nome e stipendio degli impiegati che guadagnano più dei capi / 3

Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5998	Bianchi	37	38	5998	5698
9553	Neri	42	35	9553	4076
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123

$\rho_{\text{MatrC, NomeC, StipC, EtàC}} \leftarrow \text{Matr, Nome, Stip, Età}(\text{Impiegati})$

MatrC	NomeC	EtàC	StipC
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Supervisione  $\bowtie$   $\text{Impiegato=Matricola}$  Impiegati

Matricola	Nome	Età	Stipendio	Impiegato	Capo	MatrC	NomeC	EtàC	StipC
7309	Rossi	34	45	7309	5698	5698	Bruni	43	42
5998	Bianchi	37	38	5998	5698	5698	Bruni	43	42
9553	Neri	42	35	9553	4076	4076	Mori	45	50
5698	Bruni	43	42	5698	4076	4076	Mori	45	50
4076	Mori	45	50	4076	8123	8123	Lupi	46	60

$\rho_{\text{MatrC, NomeC, StipC, EtàC}} \leftarrow \text{Matr, Nome, Stip, Età}(\text{Impiegati})$

$\bowtie \text{MatrC=Capo}$

$(\text{Supervisione } \bowtie \text{Impiegato=Matricola Impiegati))$

## Esempio 3: Matricola, nome e stipendio degli impiegati che guadagnano più dei capi / 4

Matricola	Nome	Età	Stipendio	Impiegato	Capo	MatrC	NomeC	EtàC	StipC
7309	Rossi	44	45	7309	5098	5098	Bruni	43	42
8888	Bianchi	45	38	8888	8888	8888	Bianchi	45	38
8888	Bianchi	45	38	8888	8888	8888	Bianchi	45	38
8888	Bianchi	45	38	8888	8888	8888	Bianchi	45	38
8888	Bianchi	45	38	8888	8888	8888	Bianchi	45	38

$\Pi_{\text{Matricola, Nome, Stipendio}}$   
 $(\sigma_{\text{Stipendio} > \text{StipC}}($   
 $\rho_{\text{MatrC, NomeC, StipC, EtàC} \leftarrow \text{Matr, Nome, Stip, Età}(\text{Impiegati})$   
 $\bowtie \text{MatrC} = \text{Capo}$   
 $(\text{Supervisione} \bowtie \text{Impiegato} = \text{Matricola} \text{ Impiegati)))$

# Esempio 4: Matricola dei capi i cui impiegati guadagnano tutti più di 40

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Supervisione

Impiegato	Capo
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

Implementato come  
«Restituisci tutti i capi,  
tranne quelli per cui  
c'è almeno un impiegato  
che guadagna < 40

$\pi_{\text{Capo}}(\text{Supervisione}) -$

$\pi_{\text{Capo}}(\text{Supervisione}) \bowtie \text{Impiegato} = \text{Matricola}(\sigma_{\text{Stipendio} \leq 40}(\text{Impiegati}))$

# Equivalenza di Espressioni

- In Algebra, due espressioni sono **equivalenti** se producono lo stesso risultato per ogni valore. Per es:

$$X (Y + Z) = XY + XZ$$

- In Algebra Relazionale, due espressioni sono **equivalenti** se producono lo stesso risultato qualunque sia l'istanza attuale della relazioni della base di dati

# Equivalenza in Algebra Relazione: Perché importante?

- L'equivalenza è importante in pratica perché i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "costose"
- Per esempio:
  - Date due relazioni  $R_1(X_1)$  e  $R_2(X_2)$  con l'attributo  $A \in X_2$
  - Sono equivalenti:

$$\sigma_{A=10}(R_1 \bowtie R_2) \equiv R_1 \bowtie \sigma_{A=10}(R_2)$$



# Equivalenza in Algebra Relazione: Riduzione della Computazione

- Consideriamo
  1.  $\sigma_{A=10}(R_1 \bowtie R_2)$
  2.  $R_1 \bowtie \sigma_{A=10}(R_2)$
- Assumiamo che le tuple di  $R_2$  con  $A=10$  è 10%
- Numero massimo di righe create (temporaneamente) nei due casi
  1.  $|R_1| \times |R_2|$
  2.  $|R_1| \times 0.1 \times |R_2|$

# Equivalenza di espressioni

- Due espressioni sono **equivalenti** se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati
- L'equivalenza è importante in pratica perché i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "costose"

# Alcune equivalenze / 1

- Data una relazione  $R(Z)$  dove  $X, Y \subseteq Z$  e  $X \cap Y = \emptyset$  :
  1.  $\sigma_{C1 \text{ AND } C2} (R) \equiv \sigma_{C1} \sigma_{C2} (R)$
  2.  $\pi_X (\pi_{XY} (R)) \equiv \pi_X (R)$

## Alcune equivalenze / 2

- Date due relazioni  $R_1(X_1)$  e  $R_2(X_2)$  con  $Y_1 \subset X_1$ ,  $Y_2 \subset X_2$ :
  3.  $\sigma_{C1}(R_1 \bowtie R_2) \equiv R_1 \bowtie \sigma_{C1}(R_2)$   
se C1 condizione su  $X_2$
  4.  $\pi_{X_1 Y_2}(R_1 \bowtie R_2) \equiv \pi_{X_1 Y_2}(R_1 \bowtie \pi_{Y_2}(R_2))$   
se  $X_2 - Y_2$  non coinvolti del join
- In linguaggio naturale:
  3. Selezione prima del join
  4. Anticipazione della proiezione su  $Y_2$  se  $(X_2 - Y_2)$  non nel join e non nell'output

## Alcune equivalenze / 3

- Date due relazioni  $R_1(X_1)$  e  $R_2(X_2)$ :

5.  $\sigma_C(R_1 \bowtie R_2) \equiv R_1 \bowtie_C R_2$

6.  $\sigma_C(R_1 \cup R_2) \equiv \sigma_C(R_1) \cup \sigma_C(R_2)$

7.  $\sigma_C(R_1 - R_2) \equiv \sigma_C(R_1) - \sigma_C(R_2)$

- In linguaggio naturale:

5. Selezione su C allo stesso tempo del join

6. Selezione prima dell'unione

7. Selezione prima della differenza

# Esempio: Capi con impiegati con meno di 40 anni

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Impiegato	Capo	Supervisione
7309	5698	
5998	5698	
9553	4076	
5698	4076	
4076	8123	

$\pi_{\text{Capo}} (\sigma_{(\text{Impiegato}=\text{Matricola AND Et\`a}<40)} (\text{Supervisione} \bowtie \text{Impiegati}))$

**Super-inefficiente!!**

# Esempio: Capi con impiegati con meno di 40 anni / 2

$\pi_{\text{Capo}} (\sigma_{(\text{Impiegato}=\text{Matricola} \text{ AND } \text{Età}<40)} (\text{Supervisione} \bowtie \text{Impiegati}))$

Regola 1:  $\pi_{\text{Capo}}$

$(\sigma_{\text{Età}<40} \sigma_{\text{Impiegato}=\text{Matricola}} (\text{Supervisione} \bowtie \text{Impiegati}))$

Regola 5:  $\pi_{\text{Capo}}$

$(\sigma_{\text{Età}<40} (\text{Supervisione} \bowtie_{\text{Impiegato}=\text{Matricola}} \text{Impiegati}))$

Regola 4:  $\pi_{\text{Capo}} (\pi_{\text{Matricola}} (\sigma_{\text{Età}<40} (\text{Impiegati})))$

$\bowtie_{\text{Impiegato}=\text{Matricola}} \text{Supervisione}$

# Nota

- I DBMS usano regole di equivalenza per ottimizzare le interrogazioni
  - **Obiettivo:** Relazioni «Intermedie» con il minimo numero di tuple ed attributi
  - **Osservazione:** I DBMS non eseguono veramente le interrogazioni come vengono formulate
- Questo corso non si preoccupa dell'ottimizzazione: i DBMS lo faranno!



# Selezione con valori nulli / 1

## 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**

# Selezione con valori nulli / 2

## Impiegati

Matricola	Cognome	Filiale	Età
7000	Rossi	Roma	32
5998	Neri	Milano	45
6550	Bianchi	Milano	NULL

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

**la condizione atomica è vera solo per valori non nulli**

# Selezione con valori nulli / 3

## Impiegati

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

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

# Selezione con valori nulli: Soluzione

- La condizione  $\sigma_{\text{Età} > 40} (\text{Impiegati})$  è vera solo per valori non nulli
- Per riferirsi ai valori nulli esistono forme apposite di condizioni:

IS NULL  
IS NOT NULL

# Selezione con valori nulli: Soluzione

## Impiegati

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

$$\begin{aligned} & \sigma_{\text{Età} > 30}(\text{Persone}) \cup \sigma_{\text{Età} \leq 30}(\text{Persone}) \cup \sigma_{\text{Età IS NULL}}(\text{Persone}) \\ & = \\ & \sigma_{\text{Età} > 30 \vee \text{Età} \leq 30 \vee \text{Età IS NULL}}(\text{Persone}) \\ & = \\ & \text{Persone} \end{aligned}$$

## Impiegati

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

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

## Impiegati

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

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

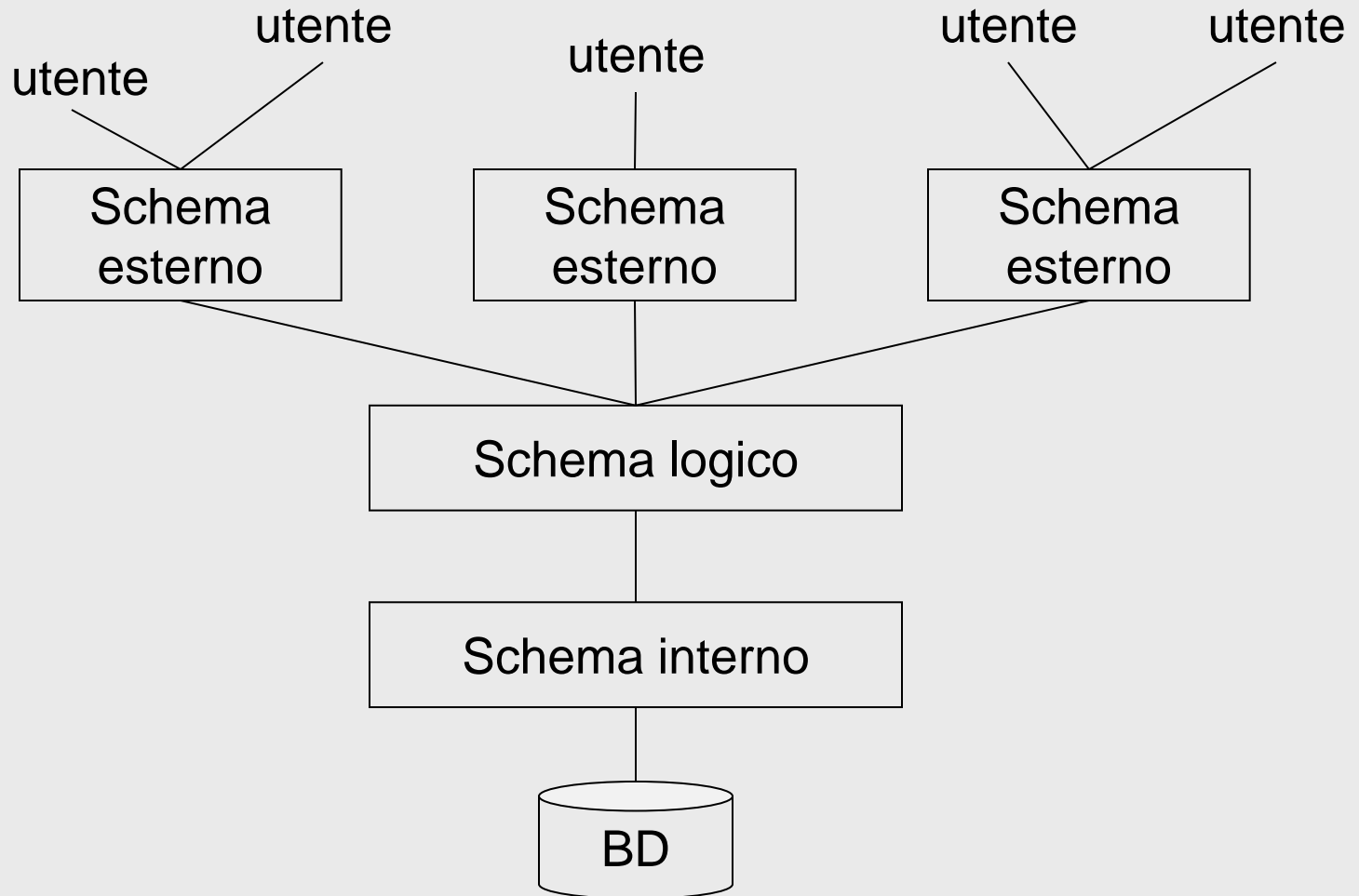
# Viste (relazioni derivate)

- Rappresentazioni diverse per gli stessi dati
- Relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)

Nota. **Relazioni di base:** contenuto autonomo



# 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:

Supervisione =

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

# 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
    - 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 (o quasi)

# Interrogazioni sulle viste: Esempio

Tutti gli impiegati con capo «Bruni»

Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Direzione

Reparto	Capo
A	Mori
B	Bruni

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

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

viene eseguita come

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

# 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

# Aggiornamenti tramite le Viste:

## Caso OK

### Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Verdi	A
Lupi	C

### Direzione

Reparto	Capo
A	Mori
B	Bruni
C	Bruni
C	Bruni

Supervisione = Afferenza  $\bowtie$  Direzione

Impiegato	Reparto	Capo
Rossi	A	Mori
Neri	B	Bruni
Verdi	A	Mori
Lupi	C	Bruni

Cambi reversabili  
sulle relazioni di partenza

Supponiamo di voler  
aggiungere la riga verde



# Aggiornamenti tramite le Viste:

## Caso NOK

### Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Verdi	A
Lupi	??

### Direzione

Reparto	Capo
A	Mori
B	Bruni
C	Bruni
??	Bruni

Supervisione =  $\Pi_{\text{Impiegato, Capo}} (\text{Afferenza} \bowtie \text{Direzione})$

Impiegato	Capo
Rossi	Mori
Neri	Bruni
Verdi	Mori
Lupi	Bruni

Cambi non reversabili  
sulle relazioni di partenza

Supponiamo di voler  
aggiungere la riga verde

# Viste e aggiornamenti

- **Obiettivo:** modificare le relazioni di base modificando la vista.
- Le relazioni di basi vengono aggiornate, in modo tale che la vista sia coerente.
- Possibile in pochi casi (es. quando il join è completo)

# Una convenzione e notazione alternativa per i join

L'approccio usato in SQL

- Ignoriamo il join naturale su attributi con nomi uguali
- Per "riconoscere" attributi con lo stesso nome possiamo mettere il nome della relazione davanti
- Usiamo viste per ridenominare relazioni

## Esempio 3 Rivisitato : Matricola, nome e stipendio degli impiegati che guadagnano più dei capi / 1

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Impiegato	Capo
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

Supervisione

Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5998	Bianchi	37	38	5998	5698
9553	Neri	42	35	9553	4076
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123

$$\text{SupImp} = (\text{Supervisione} \bowtie_{\text{Impiegato=Matricola}} \text{Impiegati})$$

## Esempio 3 Rivisitato : Matricola, nome e stipendio degli impiegati che guadagnano più dei capi / 2

Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5998	Bianchi	37	38	5998	5698
9553	Neri	42	35	9553	4076
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

SupImp = (Supervisione  $\bowtie$ <sub>Impiegato=Matricola</sub> Impiegati)

SupImp

Impiegati

Matricola	Nome	Età	Stipendio	Impiegato	Capo	Matricola	Nome	Età	Stipendio
7309	Rossi	34	45	7309	5698	5698	Bruni	43	42
5998	Bianchi	37	38	5998	5698	5698	Bruni	43	42
9553	Neri	42	35	9553	4076	4076	Mori	45	50
5698	Bruni	43	42	5698	4076	4076	Mori	45	50
4076	Mori	45	50	4076	8123	8123	Lupi	46	60

SupImp  $\bowtie$ <sub>Capo=Impiegati.Matricola</sub> Impiegati

## Esempio 3 Rivisitato : Matricola, nome e stipendio degli impiegati che guadagnano più dei capi / 3

Matricola	Nome	Capo	Stipendio	Impiegato	Capo	Matricola	Nome	Capo	Stipendio
7309	Rossi	4	45	7309	5	98	5	98	4
8558	Mani	3	35	8558	4	78	4	78	5
1078	Mani	3	35	1078	3	28	3	28	5

$\pi_{\text{SupImp.matricola, SupImp.Nome, SupImp.Stipendio}}($

$\sigma_{\text{SupImp.Stipendio} > \text{Impiegato.Stipendio}} (\text{SupImp} \bowtie_{\text{Capo} = \text{Impiegati.Matricola}} \text{Impiegati})$

)

# Riferimenti

- Sezione 3.1 del libro



# Esercizio 1

- Si consideri la seguente base di dati dei diversi servizi sociali in città diverse di Italia:  
SERVIZI\_SOCIALI (Citta, Servizio, Anno, Spesa)  
POSIZIONE (Citta, Regione, Abitanti)
- Non è possibile avere due città con lo stesso nome.
- Una tupla (Padova, Babysitting, 2019, 30000) in SERVIZI SOCIALI indica che Padova ha speso nel 2019 la cifra di 30000€ per il servizio sociale Babysitting.

Restituire le regioni che forniscono i servizi di babysitting

$$\pi_{\text{REGIONE}} ((\sigma_{\text{SERVIZIO}=\text{"Babysitting"}} (\text{SERVIZI\_SOCIALI}))$$
$$\bowtie_{\text{SERVIZI\_SOCIALI.CITTA}=\text{POSIZIONE.CITTA}} \text{POSIZIONE})$$





## Esercizio 2

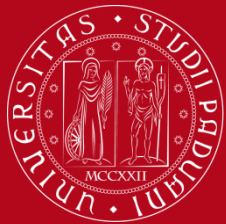
- Si consideri la seguente base di dati dei diversi servizi sociali in città diverse di Italia:  
SERVIZI\_SOCIALI (Citta, Servizio, Anno, Spesa)  
POSIZIONE (Citta, Regione, Abitanti)
- Non è possibile avere due città con lo stesso nome.
- Una tupla (Padova, Babysitting, 2019, 30000) in SERVIZI SOCIALI indica che Padova ha speso nel 2019 la cifra di 30000€ per il servizio sociale Babysitting.

Restituire le città che forniscono almeno due servizi sociali

S1=SERVIZI\_SOCIALI

S2=SERVIZI\_SOCIALI

$\pi_{S1.CITTA} (S1 \bowtie_{S1.CITTA=S2.CITTA \text{ AND } S1.SERVIZIO \neq S2.SERVIZIO} S2)$



# Esercizio 3

## (Esame 30 Giugno 2021)

- Si consideri la seguente base di dati dei diversi servizi sociali in città diverse di Italia:  
SERVIZI\_SOCIALI (Citta, Servizio, Anno, Spesa)  
POSIZIONE (Citta, Regione, Abitanti)
- Non è possibile avere due città con lo stesso nome.
- Una tupla (Padova, Babysitting, 2019, 30000) in SERVIZI SOCIALI indica che Padova ha speso nel 2019 la cifra di 30000€ per il servizio sociale Babysitting.

Restituire le città che forniscono al più un servizio sociale

S1=SERVIZI\_SOCIALI

S2=SERVIZI\_SOCIALI

$\pi_{CITTA} (S1) \setminus$

$\pi_{S1.CITTA} (S1 \bowtie_{S1.CITTA=S2.CITTA \text{ AND } S1.SERVIZIO \neq S2.SERVIZIO} S2)$



## Esercizio 4

- Si consideri la seguente base di dati dei diversi servizi sociali in città diverse di Italia:  
SERVIZI\_SOCIALI (Città, Servizio, Anno, Spesa)  
POSIZIONE (Città, Regione, Abitanti)
- Non è possibile avere due città con lo stesso nome.
- Una tupla (Padova, Babysitting, 2019, 30000) in SERVIZI SOCIALI indica che Padova ha speso nel 2019 la cifra di 30000€ per il servizio sociale Babysitting.

Restituire la città con almeno 200000 abitanti che ha speso di meno nel servizio di babysitting nel 2024, tra tutte le città di almeno 200000 abitanti.

$$S1 = \pi_{\text{POSIZIONE.CITTA, SPESA}} (\sigma_{\text{ABITANTI} \geq 200000} (\text{POSIZIONE}))$$

$$\bowtie_{\text{SERVIZI\_SOCIALI.CITTA} = \text{POSIZIONE.CITTA}}$$

$$(\sigma_{\text{SERVIZIO} = \text{'Babysitting'} \text{ AND } \text{ANNO} = 2024} (\text{SERVIZI\_SOCIALI}))$$

$$S2 = S1$$

$$\pi_{\text{CITTA}} (S1) \setminus \pi_{S1.CITTA} (S1 \bowtie_{S1.SPESA > S2.SPESA} S2)$$