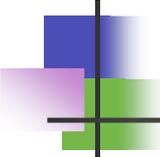


# Linguaggi di interrogazione

---

- Un'interrogazione è una funzione che data una base di dati produce una relazione su un dato schema
- Linguaggi di interrogazione
  - **Procedurali**
    - specificano il procedimento di generazione del risultato ("come ottengo il risultato")
    - Algebra relazionale
  - **Dichiarativi**
    - specificano le proprietà del risultato ("che cosa è il risultato")
    - Calcolo relazionale

1



# Algebra relazionale

---

- E' basata su un insieme di **operatori** che sono definiti su relazioni e che producono relazioni
  - operatori **insiemistici** (unione, intersezione, differenza)
  - **ridenominazione, selezione, proiezione**
  - l'operatore di **join** (join naturale, prodotto cartesiano, theta-join)
- Si possono combinare gli operatori di base per ottenere interrogazioni anche complesse

2

# Unione, Intersezione, differenza

- Questi operatori hanno senso solo se si applicano fra relazioni con lo stesso schema (con gli stessi attributi)

❶ L'**unione** di due relazioni  $r_1$  e  $r_2$  è la relazione  $r = r_1 \cup r_2$  che contiene le tuple che appartengono ad  $r_1$  oppure ad  $r_2$ , oppure ad entrambe.

| Matricola | Nome   | Cognome |
|-----------|--------|---------|
| A80010012 | Mario  | Rossi   |
| A80010200 | Gianni | Longhi  |
| A80010111 | Fabio  | Verdi   |

LAUREATI

| Matricola | Nome  | Cognome |
|-----------|-------|---------|
| A82010007 | Marco | Beni    |
| A82010345 | Anna  | Bianchi |

SPECIALIZZATI

| Matricola | Nome   | Cognome |
|-----------|--------|---------|
| A80010012 | Mario  | Rossi   |
| A80010200 | Gianni | Longhi  |
| A80010111 | Fabio  | Verdi   |
| A82010007 | Marco  | Beni    |
| A82010345 | Anna   | Bianchi |

STUDENTI

3

# Intersezione e differenza

- L'**intersezione** di due relazioni  $r_1(X)$  e  $r_2(X)$  definite su un insieme di attributi  $X$  è la relazione  $r(X) = r_1(X) \cap r_2(X)$  contenente le tuple che appartengono sia a  $r_1(X)$  che a  $r_2(X)$
- La **differenza** di due relazioni  $r_1(X)$  e  $r_2(X)$  definite su un insieme di attributi  $X$  è la relazione  $r(X) = r_1(X) - r_2(X)$  contenente le tuple che appartengono a  $r_1(X)$  ma non a  $r_2(X)$

| Matricola | Nome   | Cognome |
|-----------|--------|---------|
| 3456575   | Mario  | Rossi   |
| 3432334   | Gianni | Longhi  |
| 3223343   | Fabio  | Verdi   |

SPECIALIZZATI

| Matricola | Nome   | Cognome |
|-----------|--------|---------|
| 3456575   | Anna   | Verdi   |
| 3223342   | Isa    | Belli   |
| 3432334   | Gianni | Longhi  |
| 3222122   | Fabio  | Feltri  |

LAUREATI

| Matricola | Nome   | Cognome |
|-----------|--------|---------|
| 3432334   | Gianni | Longhi  |

SPECIALIZZATI  $\cap$  LAUREATI

| Matricola | Nome  | Cognome |
|-----------|-------|---------|
| 3456575   | Mario | Rossi   |
| 3223343   | Fabio | Verdi   |

SPECIALIZZATI - LAUREATI

4

# Ridenominazione

- Modifica il nome di un sottoinsieme degli attributi di una relazione lasciando inalterato il contenuto delle relazione (modifica lo schema)
- L'operatore ha per argomento una relazione  $r(X)$  e indica la corrispondenza fra i nomi vecchi e nuovi di un sottoinsieme di attributi di  $X$

$$\rho_{A_1, A_2, \dots, A_n \leftarrow B_1, B_2, \dots, B_n}(r)$$

dove  $B_1, B_2, \dots, B_n \in X$

5

## Un esempio

IMPIEGATI

| <u>Matricola</u> | Cognome | Agenzia | Stipendio |
|------------------|---------|---------|-----------|
| 2321112          | Belli   | Milano  | 55        |
| 3432334          | Longhi  | Roma    | 45        |



| <u>Matricola</u> | Cognome | Sede   | Retribuzione |
|------------------|---------|--------|--------------|
| 2321112          | Belli   | Milano | 55           |
| 3432334          | Longhi  | Roma   | 45           |

$\rho_{Sede, Retribuzione \leftarrow Agenzia, Stipendio}(\text{IMPIEGATI})$

OPERAI

| <u>Matricola</u> | Cognome | Fabbrica | Salario |
|------------------|---------|----------|---------|
| 4434556          | Sordi   | Monza    | 35      |
| 4568797          | Bianchi | Viterbo  | 33      |



| <u>Matricola</u> | Cognome | Sede    | Retribuzione |
|------------------|---------|---------|--------------|
| 4434556          | Sordi   | Monza   | 35           |
| 4568797          | Bianchi | Viterbo | 33           |

$\rho_{Sede, Retribuzione \leftarrow Fabbrica, Salario}(\text{OPERAI})$



| <u>Matricola</u> | Cognome | Sede    | Retribuzione |
|------------------|---------|---------|--------------|
| 2321112          | Belli   | Milano  | 55           |
| 3432334          | Longhi  | Roma    | 45           |
| 4434556          | Sordi   | Monza   | 35           |
| 4568797          | Bianchi | Viterbo | 33           |



$\rho_{Sede, Retribuzione \leftarrow Agenzia, Stipendio}(\text{IMPIEGATI}) \cup \rho_{Sede, Retribuzione \leftarrow Fabbrica, Salario}(\text{OPERAI})$

6

# Selezione

- L'operatore di **selezione**  $\sigma_{cond}(R)$  produce una relazione con lo stesso schema di R, che contiene le tuple di R che soddisfano la condizione *cond*
- Le condizioni di selezione possono prevedere confronti fra attributi e fra attributi e costanti e possono essere costruite combinando condizioni più semplici con i connettivi logici  $\vee$  (or),  $\wedge$  (and) e  $\neg$  (not)

IMPIEGATI(Cognome, Nome, Età, Stipendio)

$\sigma_{Età < 30 \text{ AND } Stipendio > 4.000.000}(\text{IMPIEGATI})$



Vengono selezionati i record relativi ad impiegati con età minore di 30 anni e stipendio superiore a 4 milioni

7

# Esempi di selezione

| Cognome | Nome    | Età | Stipendio |
|---------|---------|-----|-----------|
| Rossi   | Alberto | 25  | 2.000.000 |
| Verdi   | Luca    | 40  | 4.500.000 |
| Billi   | Paolo   | 28  | 4.100.000 |
| Luti    | Anna    | 29  | 5.000.000 |

IMPIEGATI

| Cognome | Nome  | Età | Stipendio |
|---------|-------|-----|-----------|
| Billi   | Paolo | 28  | 4.100.000 |
| Luti    | Anna  | 29  | 5.000.000 |

$\sigma_{Età < 30 \text{ AND } Stipendio > 4.000.000}(\text{IMPIEGATI})$

| Cognome | Nome    | Città di nascita | Residenza |
|---------|---------|------------------|-----------|
| Rossi   | Alberto | Firenze          | Firenze   |
| Verdi   | Luca    | Siena            | Pisa      |
| Billi   | Paolo   | Pisa             | Lucca     |
| Luti    | Anna    | Prato            | Prato     |

CITTADINI

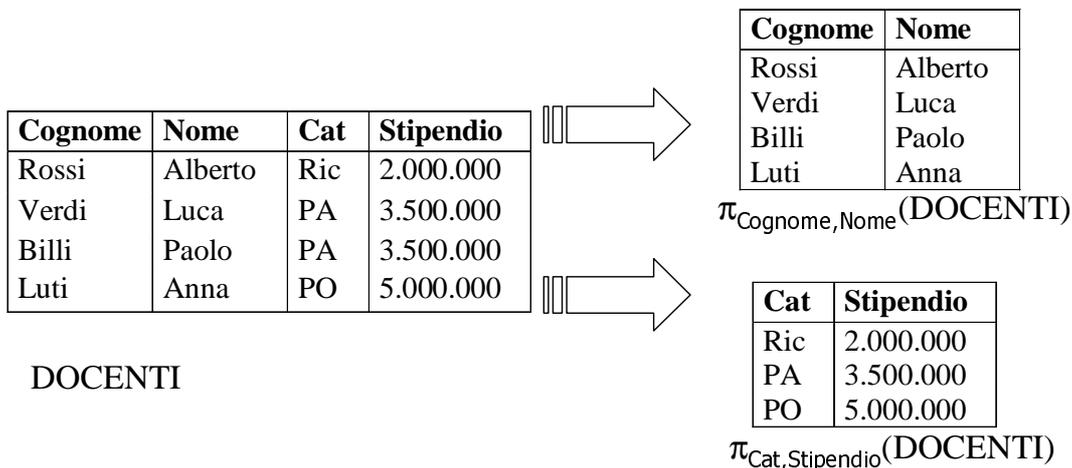
| Cognome | Nome    | Città di nascita | Residenza |
|---------|---------|------------------|-----------|
| Rossi   | Alberto | Firenze          | Firenze   |
| Luti    | Anna    | Prato            | Prato     |

$\sigma_{Città \text{ di } Nascita = Residenza}(\text{CITTADINI})$

8

# Proiezione

- Dati una relazione  $R(X)$  e un sottoinsieme  $Y$  degli attributi in  $X$ , la **proiezione** di  $R$  su  $Y$   $\pi_Y(R)$  è l'insieme delle tuple su  $Y$  ottenute dalle tuple di  $R$  considerando solo i valori su  $Y$



9

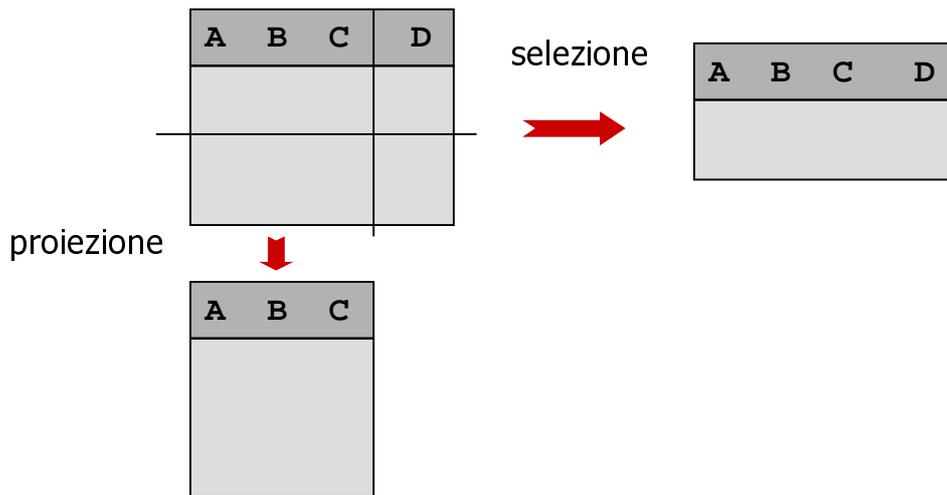
# Cardinalità della proiezione

- una proiezione
  - contiene al più tante tuple quante l'operando
  - può contenerne di meno (si eliminano i duplicati)
- se  $X$  è una superchiave di  $R$ , allora  $\pi_X(R)$  contiene esattamente tante tuple quante  $R$

10

# Selezione e proiezione

- Sono operatori "ortogonali"
  - selezione** - decomposizione orizzontale
  - proiezione** - decomposizione verticale



11

# Combinazione di $\sigma$ e $\pi$

- Combinando selezione e proiezione si possono estrarre informazioni da una relazione

| Cognome | Nome    | Cat | Stipendio |
|---------|---------|-----|-----------|
| Rossi   | Alberto | Ric | 2.000.000 |
| Verdi   | Luca    | PA  | 3.500.000 |
| Billi   | Paolo   | PA  | 3.500.000 |
| Luti    | Anna    | PO  | 5.000.000 |

Docenti

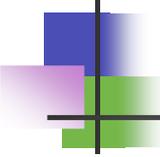
seleziona Nome e Cognome  
dei Professori Associati

$\pi_{\text{Cognome, Nome}}(\sigma_{\text{Cat}='PA'}(\text{docenti}))$

| Cognome | Nome  |
|---------|-------|
| Verdi   | Luca  |
| Billi   | Paolo |

Professori Associati

12

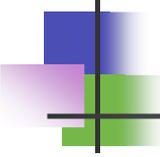


## Limitazione di $\sigma$ e $\pi$

---

- Combinando selezione e proiezione, possiamo estrarre informazioni da **una** relazione
- Sono infatti operatori con un solo argomento
- non possiamo però correlare informazioni presenti in relazioni diverse
- Si introduce l'operatore di **JOIN** che permette di correlare dati in relazioni diverse

13



## Join

---

- E' l'operatore che permette di correlare dati contenuti in relazioni diverse confrontando i valori contenuti in esse
- Utilizza il fatto che il modello relazionale è basato su valori
- Esistono due varianti principali dell'operatore
  - **Join naturale**
  - **theta-join**

14

## Join naturale

- E' un operatore che correla dati in relazioni diverse sulla base di

**valori uguali in attributi con lo stesso nome**

- La relazione risultante
  - ha per attributi l'unione degli attributi delle relazioni di partenza
  - le sue tuple sono ottenute combinando le tuple delle relazioni con valori uguali sugli attributi comuni

$$r_1(X_1) \bowtie r_2(X_2) = \{t \text{ su } X_1 \cup X_2 \mid t[X_1] \in r_1 \text{ e } t[X_2] \in r_2\}$$

15

## Verifica della definizione...

$$r(X) = r_1(X_1) \bowtie r_2(X_2) = \{t \text{ su } X_1 \cup X_2 \mid t[X_1] \in r_1 \text{ e } t[X_2] \in r_2\}$$

$$X_1 = \{A_1, \dots, A_k, C_1, \dots, C_m\}$$

$$X_2 = \{B_1, \dots, B_r, C_1, \dots, C_m\}$$

$$X = X_1 \cup X_2 = \{A_1, \dots, A_k, B_1, \dots, B_r, C_1, \dots, C_m\}$$

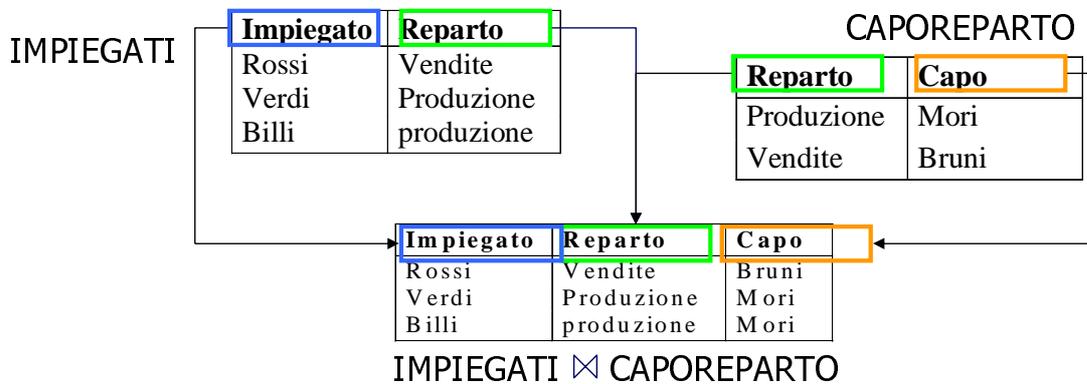
$$t[X_1] \in r_1 \text{ e } t[X_2] \in r_2 \Rightarrow$$

$$t[C_1, \dots, C_m] = t_1[C_1, \dots, C_m] = t_2$$

$$[C_1, \dots, C_m]$$

16

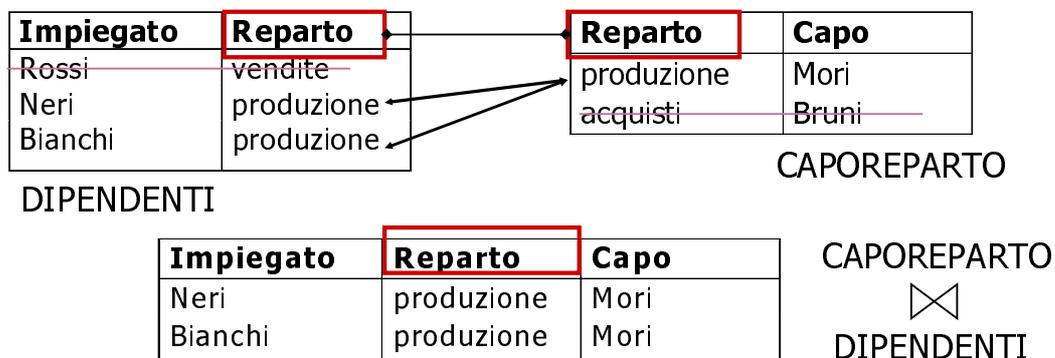
# Esempio di join naturale



- In questo caso si ha un join **completo**
  - $\forall t_1 \in r_1 \exists t \in r_1 \bowtie r_2$  tale che  $t[X_1]=t_1$
  - $\forall t_2 \in r_2 \exists t \in r_1 \bowtie r_2$  tale che  $t[X_2]=t_2$

17

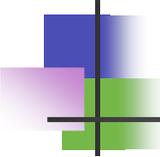
# Join con tuple "pendenti"



La prima tupla della relazione DIPENDENTI e la seconda della relazione CAPOREPARTO non generano nessuna tupla nel join

18





## Cardinalità del join

---

- In generale

$$0 \leq |r_1 \bowtie r_2| \leq |r_1| \times |r_2|$$

- Se il join è completo  $|r_1 \bowtie r_2| \geq \max(|r_1|, |r_2|)$
- Se  $X_1 \cap X_2$  contiene una chiave di  $r_2$  allora

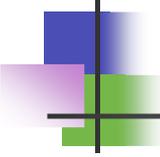
$$|r_1 \bowtie r_2| \leq |r_1|$$

Ogni tupla di  $r_1$  si combina al più con una tupla di  $r_2$

- Se  $X_1 \cap X_2$  contiene una chiave di  $r_2$  e sussiste il vincolo di integrità referenziale fra  $X_1 \cap X_2$  in  $r_1$  e la chiave di  $r_2$ , allora

$$|r_1 \bowtie r_2| = |r_1|$$

21



## Join esterno

---

- Il join tralascia le tuple di una relazione che non hanno corrispondenza nell'altra relazione
- Il join **esterno (outer join)** estende, con valori nulli, le tuple che verrebbero tagliate fuori da un join **(interno)**
- esiste in tre versioni:
  - **sinistro** - mantiene tutte le tuple del primo operando estendendole con valori nulli se necessario
  - **destro** - estende le tuple del secondo operando
  - **completo** - estende le tuple di entrambi gli operandi
- E' previsto da SQL

22

# Esempi di join esterno

| Impiegato | Reparto    |
|-----------|------------|
| Rossi     | vendite    |
| Neri      | produzione |
| Bianchi   | produzione |

Dipendenti

| Reparto    | Capo  |
|------------|-------|
| produzione | Mori  |
| acquisti   | Bruni |

Caporeparto

Dipendenti  $\bowtie$  LEFT Caporeparto

| Impiegato | Reparto    | Capo |
|-----------|------------|------|
| Rossi     | vendite    | NULL |
| Neri      | produzione | Mori |
| Bianchi   | produzione | Mori |

Dipendenti  $\bowtie$  RIGHT Caporeparto

| Impiegato | Reparto    | Capo  |
|-----------|------------|-------|
| Neri      | produzione | Mori  |
| Bianchi   | produzione | Mori  |
| NULL      | acquisti   | Bruni |

Dipendenti  $\bowtie$  FULL Caporeparto

| Impiegato | Reparto    | Capo  |
|-----------|------------|-------|
| Rossi     | vendite    | NULL  |
| Neri      | produzione | Mori  |
| Bianchi   | produzione | Mori  |
| NULL      | acquisti   | Bruni |

23

# Proprietà del Join naturale

- Il join è **commutativo** e **associativo**

$$r_1 \bowtie r_2 = r_2 \bowtie r_1$$

$$r_1 \bowtie (r_2 \bowtie r_3) = (r_1 \bowtie r_2) \bowtie r_3$$

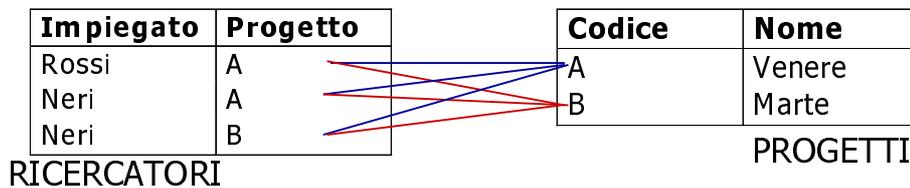
- Se  $r_1$  e  $r_2$  sono definite sullo stesso insieme di attributi, il join coincide con l'intersezione

$$r_1(X) \bowtie r_2(X) = r_1(X) \cap r_2(X)$$

- Se  $r_1$  e  $r_2$  sono definite su insiemi di attributi disgiunti ( $X_1 \cap X_2 = \emptyset$ ), il join diventa un "prodotto cartesiano" (concatenazione di tutte le tuple in  $r_1$  e con tutte le tuple in  $r_2$ )

24

# Join e *prodotto cartesiano*



| Impiegato | Progetto | Codice | Nome   |
|-----------|----------|--------|--------|
| Rossi     | A        | A      | Venere |
| Neri      | A        | A      | Venere |
| Neri      | B        | A      | Venere |
| Rossi     | A        | B      | Marte  |
| Neri      | A        | B      | Marte  |
| Neri      | B        | B      | Marte  |

RICERCATORI  
 $\bowtie$   
 PROGETTI

25

# Il theta-join

- E' un operatore derivato. Corrisponde ad effettuare il prodotto cartesiano delle due relazioni seguito da una selezione basata su una condizione F.
- Se  $r_1$  e  $r_2$  non hanno attributi in comune

$$r_1 \bowtie_F r_2 = \sigma_F(r_1 \bowtie r_2)$$

Selezione delle tuple che soddisfano F

Tutte le possibili combinazioni delle tuple

- E' importante dal punto di vista pratico perché è utilizzato nei sistemi di basi di dati esistenti

26

# Perchè theta-join?

- La condizione **F** è spesso una espressione che lega con operatori logici (AND, OR, NOT) atomi di confronto

$$A_1 \vartheta A_2$$

dove  $\vartheta$  è uno degli operatori di confronto (=, >, <, ...)

- se l'operatore è sempre l'uguaglianza (=) allora si parla di **equi-join**
- Il join naturale può essere ottenuto combinando ridenominazione equi-join e proiezione

$$\pi_{ABC}(r_1 \bowtie_{B=B'}(\rho_{B' \leftarrow B}(r_2)))$$

con  $r_1(AB)$  e  $r_2(BC)$



27

# Un esempio di theta-join

| Impiegato | Progetto |
|-----------|----------|
| Rossi     | A        |
| Neri      | A        |
| Neri      | B        |

RICERCATORI

| Codice | Nome   |
|--------|--------|
| A      | Venere |
| B      | Marte  |

PROGETTI

RICERCATORI



PROGETTI

| Impiegato | Progetto | Codice | Nome   |
|-----------|----------|--------|--------|
| Rossi     | A        | A      | Venere |
| Neri      | A        | A      | Venere |
| Neri      | B        | A      | Venere |
| Rossi     | A        | B      | Marte  |
| Neri      | A        | B      | Marte  |
| Neri      | B        | B      | Marte  |

| Impiegato | Progetto | Codice | Nome   |
|-----------|----------|--------|--------|
| Rossi     | A        | A      | Venere |
| Neri      | A        | A      | Venere |
| Neri      | B        | B      | Marte  |

RICERCATORI



PROGETTI

Progetto=Codice

28

# Interrogazioni in algebra relazionale

- Un'interrogazione a una base di dati produce una serie di record che soddisfano i criteri richiesti
- E' una funzione che applicata ad un'istanza di basi di dati produce una relazione ovvero dei dati organizzati come tuple di una relazione
- Le interrogazioni si possono rappresentare con gli operatori dell'algebra relazionale che forniscono una procedura per calcolare il risultato

29

## Un esempio di interrogazione

| <u>Matricola</u> | <u>Nome</u>   | <u>Età</u> | <u>Stipendio</u> |
|------------------|---------------|------------|------------------|
| 101              | Marco Rossi   | 23         | 1.500            |
| 103              | Paolo Bianchi | 34         | 2.680            |
| 105              | Anna Falchi   | 33         | 1.700            |
| 110              | Gaia Belli    | 36         | 2.500            |
| 134              | Luca Forti    | 27         | 2.500            |
| 145              | Sonia Melli   | 23         | 1.500            |
| 149              | Mario Mori    | 33         | 1.800            |
| 153              | Bruno Bruni   | 35         | 1.500            |
| 155              | Filippo Mei   | 30         | 2.500            |

IMPIEGATI

| <u>Capo</u> | <u>Impiegato</u> |
|-------------|------------------|
| 103         | 101              |
| 103         | 105              |
| 103         | 145              |
| 110         | 103              |
| 110         | 149              |
| 134         | 153              |

SUPERVISIONE

*Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 1.700*

30

# I passi dell'interrogazione [1]

- Si selezionano gli impiegati che guadagnano più di 1.700

$\sigma_{\text{Stipendio} > 1.700}(\text{IMPIEGATI})$

| Matricola | Nome          | Età | Stipendio |
|-----------|---------------|-----|-----------|
| 103       | Paolo Bianchi | 34  | 2.680     |
| 110       | Gaia Belli    | 36  | 2.500     |
| 134       | Luca Forti    | 27  | 2.500     |
| 149       | Mario Mori    | 33  | 1.800     |
| 155       | Filippo Mei   | 30  | 2.500     |

- Si associano gli impiegati trovati con i rispettivi capi

$\text{SUPERVISIONE} \bowtie_{\text{Impiegato}=\text{Matricola}} \sigma_{\text{Stipendio} > 1.700}(\text{IMPIEGATI})$

| Matricola | Nome          | Età | Stipendio | Capo | Impiegato |
|-----------|---------------|-----|-----------|------|-----------|
| 103       | Paolo Bianchi | 34  | 2.680     | 110  | 103       |
| 149       | Mario Mori    | 33  | 1.800     | 110  | 149       |

31

# I passi dell'interrogazione [2]

- Si estrae la matricola dei capi

$\pi_{\text{Capo}} (\text{SUPERVISIONE} \bowtie_{\text{Impiegato}=\text{Matricola}} \sigma_{\text{Stipendio} > 1.700}(\text{IMPIEGATI}))$

| Capo |
|------|
| 110  |

- Si associa la matricola con la relazione IMPIEGATI per ottenere le informazioni sui capi e poi si proietta sugli attributi richiesti (Nome, Stipendio)

$\pi_{\text{Nome, Stipendio}} (\text{IMPIEGATI} \bowtie_{\text{Matricola}=\text{Capo}} \pi_{\text{Capo}} (\text{SUPERVISIONE} \bowtie_{\text{Impiegato}=\text{Matricola}} \sigma_{\text{Stipendio} > 1.700}(\text{IMPIEGATI))))$

| Nome       | Stipendio |
|------------|-----------|
| Gaia Belli | 2.500     |

32

# Un'altra interrogazione [1]

*Trovare gli impiegati che guadagnano più del rispettivo capo, mostrando matricola, nome e stipendio di entrambi*

- Si associa la matricola del capo ad ogni impiegato

SUPERVISIONE ⋈<sub>Impiegato=Matricola</sub> IMPIEGATI

- Si associa la matricola del capo con la riga corrispondente della relazione IMPIEGATI (ridenominando i campi)

$\rho_{\text{MatrCapo, NomeCapo, StipCapo, Et\`aCapo}} \leftarrow \text{Matricola, Nome, Stipendio, Et\`a} (\text{IMPIEGATI})$

⋈<sub>MatrCapo=Capo</sub> (SUPERVISIONE ⋈<sub>Impiegato=Matricola</sub> IMPIEGATI)

33

# Un'altra interrogazione [2]

IMPIEGATI

| Nome          | Et\`a | Stipendio | Matricola |
|---------------|-------|-----------|-----------|
| Marco Rossi   | 23    | 1.500     | 101       |
| Paolo Bianchi | 34    | 2.680     | 103       |
| Anna Falchi   | 33    | 1.700     | 105       |
| Gaia Belli    | 36    | 2.500     | 110       |
| Luca Forti    | 27    | 2.500     | 134       |
| Sonia Melli   | 23    | 1.500     | 145       |
| Mario Mori    | 33    | 1.800     | 149       |
| Bruno Bruni   | 35    | 1.500     | 153       |
| Filippo Mei   | 30    | 2.500     | 155       |

SUPERVISIONE

| Impiegato | Capo |
|-----------|------|
| 101       | 103  |
| 103       | 103  |
| 105       | 103  |
| 145       | 103  |
| 103       | 110  |
| 149       | 110  |
| 153       | 134  |



| Matricola | Nome          | Et\`a | Stipendio | Impiegato | Capo |
|-----------|---------------|-------|-----------|-----------|------|
| 101       | Marco Rossi   | 23    | 1.500     | 101       | 103  |
| 103       | Paolo Bianchi | 34    | 2.680     | 103       | 110  |
| 105       | Anna Falchi   | 33    | 1.700     | 105       | 103  |
| 145       | Sonia Melli   | 23    | 1.500     | 145       | 103  |
| 149       | Mario Mori    | 33    | 1.800     | 149       | 110  |
| 153       | Bruno Bruni   | 35    | 1.500     | 153       | 134  |

SUPERVISIONE ⋈<sub>Impiegato=Matricola</sub> IMPIEGATI

34

# Un'altra interrogazione [3]

$\rho_{\text{MatrCapo, NomeCapo, StipCapo, Et\grave{a}Capo} \leftarrow \text{Matricola, Nome, Stipendio, Et\grave{a}} (\text{IMPIEGATI})$

| NomeCapo      | Et\grave{a}Capo | StipCapo | MatrCapo |
|---------------|-----------------|----------|----------|
| Marco Rossi   | 23              | 1.500    | 101      |
| Paolo Bianchi | 34              | 2.680    | 103      |
| Anna Falchi   | 33              | 1.700    | 105      |
| Gaia Belli    | 36              | 2.500    | 110      |
| Luca Forti    | 27              | 2.500    | 134      |
| Sonia Melli   | 23              | 1.500    | 145      |
| Mario Mori    | 33              | 1.800    | 149      |
| Bruno Bruni   | 35              | 1.500    | 153      |
| Filippo Mei   | 30              | 2.500    | 155      |

| Capo | Matricola | Nome          | Et\grave{a} | Stipendio | Impiegato |
|------|-----------|---------------|-------------|-----------|-----------|
| 103  | 101       | Marco Rossi   | 23          | 1.500     | 101       |
| 110  | 103       | Paolo Bianchi | 34          | 2.680     | 103       |
| 103  | 105       | Anna Falchi   | 33          | 1.700     | 105       |
| 103  | 145       | Sonia Melli   | 23          | 1.500     | 145       |
| 110  | 149       | Mario Mori    | 33          | 1.800     | 149       |
| 134  | 153       | Bruno Bruni   | 35          | 1.500     | 153       |

SUPERVISIONE  $\bowtie$  Impiegato=Matricola IMPIEGATI

| NomeCapo      | Et\grave{a}Capo | StipCapo | MatrCapo | Capo | Matricola | Nome          | Et\grave{a} | Stipendio | Impiegato |
|---------------|-----------------|----------|----------|------|-----------|---------------|-------------|-----------|-----------|
| Paolo Bianchi | 34              | 2.680    | 103      | 103  | 101       | Marco Rossi   | 23          | 1.500     | 101       |
| Gaia Belli    | 36              | 2.500    | 110      | 110  | 103       | Paolo Bianchi | 34          | 2.680     | 103       |
| Paolo Bianchi | 34              | 2.680    | 103      | 103  | 105       | Anna Falchi   | 33          | 1.700     | 105       |
| Paolo Bianchi | 34              | 2.680    | 103      | 103  | 145       | Sonia Melli   | 23          | 1.500     | 145       |
| Gaia Belli    | 36              | 2.500    | 110      | 110  | 149       | Mario Mori    | 33          | 1.800     | 149       |
| Luca Forti    | 27              | 2.500    | 134      | 134  | 153       | Bruno Bruni   | 35          | 1.500     | 153       |

$\rho_{\text{MatrCapo, NomeCapo, StipCapo, Et\grave{a}Capo} \leftarrow \text{Matricola, Nome, Stipendio, Et\grave{a}} (\text{IMPIEGATI})$

$\bowtie$  MatrCapo=Capo (SUPERVISIONE  $\bowtie$  Impiegato=Matricola IMPIEGATI)

35

# Un'altra interrogazione [4]

*Trovare gli impiegati che guadagnano **più del rispettivo capo**, mostrando matricola, nome e stipendio di entrambi*

- Si selezionano le righe che soddisfano il criterio  $\text{Stipendio} > \text{StipCapo}$

$\sigma_{\text{Stipendio} > \text{StipCapo}} (\rho_{\text{MatrCapo, NomeCapo, StipCapo, Et\grave{a}Capo} \leftarrow \text{Matricola, Nome, Stipendio, Et\grave{a}} (\text{IMPIEGATI}))$

$\bowtie$  MatrCapo=Capo (SUPERVISIONE  $\bowtie$  Impiegato=Matricola IMPIEGATI)

- Si estraggono gli attributi che interessano con la proiezione

$\pi_{\text{Matricola, Nome, Stipendio, MatrCapo, NomeCapo, StipCapo}} ()$

36

# Un'altra interrogazione [5]

$\rho_{\text{MatrCapo, NomeCapo, StipCapo, Et\`aCapo} \leftarrow \text{Matricola, Nome, Stipendio, Et\`a}}(\text{IMPIEGATI})$

$\bowtie_{\text{MatrCapo=Capo}}(\text{SUPERVISIONE} \bowtie_{\text{Impiegato=Matricola}} \text{IMPIEGATI})$

| NomeCapo      | Et\`aCapo | StipCapo | MatrCapo | Capo | Matricola | Nome          | Et\`a | Stipendio | Impiegato |
|---------------|-----------|----------|----------|------|-----------|---------------|-------|-----------|-----------|
| Paolo Bianchi | 34        | 2.680    | 103      | 103  | 101       | Marco Rossi   | 23    | 1.500     | 101       |
| Gaia Belli    | 36        | 2.500    | 110      | 110  | 103       | Paolo Bianchi | 34    | 2.680     | 103       |
| Paolo Bianchi | 34        | 2.680    | 103      | 103  | 105       | Anna Falchi   | 33    | 1.700     | 105       |
| Paolo Bianchi | 34        | 2.680    | 103      | 103  | 145       | Sonia Melli   | 23    | 1.500     | 145       |
| Gaia Belli    | 36        | 2.500    | 110      | 110  | 149       | Mario Mori    | 33    | 1.800     | 149       |
| Luca Forti    | 27        | 2.500    | 134      | 134  | 153       | Bruno Bruni   | 35    | 1.500     | 153       |

$\sigma_{\text{Stipendio} > \text{StipCapo}}()$

| NomeCapo   | Et\`aCapo | StipCapo | MatrCapo | Capo | Matricola | Nome          | Et\`a | Stipendio | Impiegato |
|------------|-----------|----------|----------|------|-----------|---------------|-------|-----------|-----------|
| Gaia Belli | 36        | 2.500    | 110      | 110  | 103       | Paolo Bianchi | 34    | 2.680     | 103       |

37

# Equivalenza di espressioni

- Due espressioni sono equivalenti se producono lo stesso risultato
  - $E_1 \equiv_R E_2$  se  $E_1(r) = E_2(r)$  per ogni istanza  $r$  della basi di dati con schema  $R$
  - $E_1 \equiv E_2$  se  $E_1 \equiv_R E_2$  per ogni schema  $R$
- Le trasformazioni di equivalenza sono utilizzate per ottimizzare le operazioni da effettuare per produrre il risultato di una interrogazione
  - Ottimizzazione della dimensione dei risultati intermedi

38

# Trasformazioni di equivalenza 1

- **Atomizzazione delle selezioni**

$$\sigma_{F_1 \wedge F_2} \equiv \sigma_{F_1}(\sigma_{F_2}(E))$$

- **Idempotenza delle proiezioni**

$$\pi_X(E) \equiv \pi_X(\pi_{XY}(E))$$

- **Anticipazione della selezione rispetto al join**

$$\sigma_F(E_1 \bowtie E_2) \equiv E_1 \bowtie \sigma_F(E_2)$$

se la condizione F fa riferimento solo ad attributi di  $E_2$

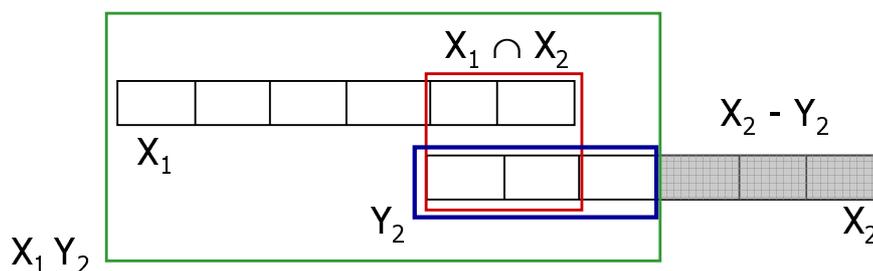
39

# Trasformazioni di equivalenza 2

- **Anticipazione della proiezione rispetto al join**

$$\pi_{X_1 Y_2}(E_1 \bowtie E_2) \equiv E_1 \bowtie \pi_{Y_2}(E_2)$$

- $E_1$  è definita sugli attributi  $X_1$  e  $E_2$  su  $X_2$
- $Y_2 \subseteq X_2$
- $(X_1 \cap X_2) \subseteq Y_2$  - gli attributi coinvolti nel join sono tutti in  $Y_2$



40

## Trasformazioni di equivalenza 3

- Eliminazione degli attributi "superflui"

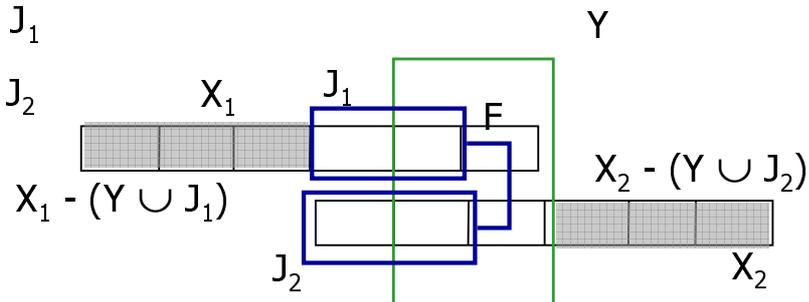
Si possono eliminare subito da ciascuna relazione gli attributi che non compaiono nel risultato finale e non sono coinvolti nel join

$$\pi_Y(E_1 \bowtie_F E_2) \equiv \pi_Y(\pi_{Y_1}(E_1) \bowtie_F \pi_{Y_2}(E_2))$$

Detti  $J_1 \subseteq X_1$  e  $J_2 \subseteq X_2$  gli attributi coinvolti nel join

- $Y_1 = (X_1 \cap Y) \cup J_1$

- $Y_2 = (X_2 \cap Y) \cup J_2$



41

## Trasformazioni di equivalenza 4

- Selezione in un prodotto cartesiano

$$\sigma_F(E_1 \bowtie E_2) \equiv E_1 \bowtie_F E_2$$

- Distributività della selezione

$$\sigma_F(E_1 \cup E_2) \equiv \sigma_F(E_1) \cup \sigma_F(E_2)$$

$$\sigma_F(E_1 - E_2) \equiv \sigma_F(E_1) - \sigma_F(E_2)$$

- Distributività della proiezione

$$\pi_X(E_1 \cup E_2) \equiv \pi_X(E_1) \cup \pi_X(E_2)$$

42

# Trasformazioni di equivalenza 5

- Distributività del join

$$E \bowtie (E_1 \cup E_2) \equiv (E \bowtie E_1) \cup (E \bowtie E_2)$$

- Selezioni con espressioni composte

$$\sigma_{F_1 \vee F_2}(R) \equiv \sigma_{F_1}(R) \cup \sigma_{F_2}(R)$$

$$\sigma_{F_1 \wedge F_2}(R) \equiv \sigma_{F_1}(R) \cap \sigma_{F_2}(R) \equiv \sigma_{F_1}(\sigma_{F_2}(R)) \quad \sigma_{F_2}(\sigma_{F_1}(R))$$

$$\sigma_{F_1 \wedge \neg F_2}(R) \equiv \sigma_{F_1}(R) - \sigma_{F_2}(R)$$

43

# Algebra con valori nulli

- La presenza di valori nulli crea problemi nel calcolo dei predicati applicati sugli attributi
- Una proposizione può essere vera (V), falsa (F) o sconosciuta (U) - **logica a 3 valori**

Età > 33

| Matricola | Nome          | Età  |
|-----------|---------------|------|
| 103       | Paolo Bianchi | 34   |
| 110       | Gaia Belli    | 36   |
| 134       | Luca Forti    | NULL |
| 149       | Mario Mori    | 33   |
| 155       | Filippo Mei   | 30   |

??

V  
V  
U  
F  
F



| Matricola | Nome          | Età |
|-----------|---------------|-----|
| 103       | Paolo Bianchi | 34  |
| 110       | Gaia Belli    | 36  |

Impiegati

$\sigma_{Età > 33}(\text{Impiegati})$

44

# Logica a 3 valori

- Occorre estendere gli operatori AND, OR e NOT alla logica a 3 valori

| NOT |   | AND | F | U | V | OR | F | U | V |
|-----|---|-----|---|---|---|----|---|---|---|
| F   | V | F   | F | F | F | F  | F | U | V |
| U   | U | U   | F | U | U | U  | U | U | V |
| V   | F | V   | F | U | V | V  | V | V | V |

- Tuttavia ci sono dei problemi....

45

# Un risultato non desiderabile

| <u>Matricola</u> | Nome          | Età  |
|------------------|---------------|------|
| 103              | Paolo Bianchi | 34   |
| 110              | Gaia Belli    | 36   |
| 134              | Luca Forti    | NULL |
| 149              | Mario Mori    | 33   |
| 155              | Filippo Mei   | 30   |

Impiegati

| <u>Matricola</u> | Nome          | Età |
|------------------|---------------|-----|
| 103              | Paolo Bianchi | 34  |
| 110              | Gaia Belli    | 36  |
| 149              | Mario Mori    | 33  |
| 155              | Filippo Mei   | 30  |

| <u>Matricola</u> | Nome          | Età |
|------------------|---------------|-----|
| 103              | Paolo Bianchi | 34  |
| 110              | Gaia Belli    | 36  |
| 149              | Mario Mori    | 33  |
| 155              | Filippo Mei   | 30  |

$$\sigma_{Età > 33}(\text{Impiegati}) \cup \sigma_{Età \leq 33}(\text{Impiegati})$$

$$\sigma_{Età > 33 \vee Età \leq 33}(\text{Impiegati})$$

≠ Impiegati

46

# Due nuove condizioni atomiche

- Si introducono le condizioni atomiche
  - **A IS NULL**  
vero per una tupla t se t[A]=NULL
  - **A IS NOT NULL**  
vero per una tupla t se t[A]≠NULL
- Questa soluzione è disponibile in SQL
- Il problema scompare....

$$\sigma_{\text{Età}>33 \vee \text{Età}\leq 33 \vee \text{Età IS NULL}}(\text{Impiegati}) = \text{Impiegati}$$

47

# Viste

- Le **relazioni derivate** permettono rappresentazioni diverse per gli stessi dati
- Si fa riferimento allo **schema esterno** dell'architettura dei DBMS
  - **Relazioni derivate**  
Relazioni il cui contenuto dipende dal contenuto di altre relazioni
  - **Relazioni di base**  
Relazioni il cui contenuto è autonomo
- Le relazioni derivate possono dipendere anche da altre relazioni derivate a patto che non ci sia circolarità

48

# Tipologie di viste

- **Viste materializzate**

Sono relazioni derivate effettivamente memorizzate nella base di dati

- immediatamente disponibili per le interrogazioni
- ridondanti
- appesantiscono gli aggiornamenti
- non sono supportate dai DBMS

- **Relazioni virtuali (viste)**

Relazioni definite per mezzo di funzioni (espressioni del linguaggio di interrogazione)

Vengono ricalcolate all'occorrenza

# Viste: un esempio

| Impiegato     | Reparto |
|---------------|---------|
| Paolo Bianchi | A       |
| Gaia Belli    | B       |
| Mario Mori    | B       |
| Filippo Mei   | C       |

| Reparto | Capo       |
|---------|------------|
| A       | Gino Bruni |
| B       | Cesare Teo |
| C       | Anna Livio |

Afferenza

Direzione

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

Le interrogazioni sono effettuate **sostituendo** la definizione della vista nell'espressione che rappresenta l'interrogazione

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

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

# Motivazioni delle viste

- Un utente *vede* solo le parti rilevanti della base di dati
- L'utente vede solo ciò che è autorizzato a vedere
- Utilizzando le viste si può semplificare la scrittura di interrogazioni (espressioni complesse e sottoespressioni ripetute)
- Un vista può permettere l'utilizzo di programmi esistenti su schemi ristrutturati
- L'utilizzo di viste non influisce sull'efficienza delle interrogazioni

51

# Programmazione e viste

*Trovare gli impiegati che hanno lo stesso capo di Mario Mori*

- Senza vista

$$\pi_{\text{Impiegato}}((\text{Afferenza} \bowtie \text{Direzione}) \bowtie \rho_{\text{ImpR,RepR} \leftarrow \text{Impiegato,Reparto}}(\sigma_{\text{Impiegato}='Mario Mori'}(\text{Afferenza} \bowtie \text{Direzione})))$$

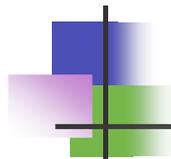
capo di Mario Mori

- Con la vista

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

$$\rho_{\text{ImpR} \leftarrow \text{Impiegato}}(\sigma_{\text{Impiegato}='Mario Mori'}(\pi_{\text{Impiegato}}(\text{Supervisione} \bowtie \text{Supervisione})))$$

52



# Viste e aggiornamenti

---

- "Aggiornare una vista"
  - modificare le relazioni di base in modo che la vista, "ricalcolata" rispecchi l'aggiornamento
- L'aggiornamento sulle relazioni di base corrispondente a quello specificato sulla vista deve essere univoco
- In generale però non è univoco....
  - Se aggiorno Supervisione inserendo una coppia (Impiegato, Capo) come posso aggiornare Afferenza e Direzione se non specifico il reparto? Dipende dai dati già presenti...
- Ben pochi aggiornamenti sono ammissibili sulle viste