

Algebra relazionale

Tratto da:

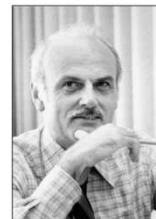
Atzeni, Ceri, Fraternali, Paraboschi, Torlone
Basi di dati *Quinto edizione*
McGraw-Hill Education, 2018
Capitolo 3: *Algebra e Calcolo relazionale*

1

Edgar F. Codd, 1923-2003

Modello Relazionale

- La storia delle banche dati relazionali è la storia di una scienza e rivoluzione tecnologica.
- La rivoluzione scientifica iniziò nel 1970 da Edgar (Ted) F. Codd presso l'IBM San Jose Research Laboratory (ora IBM Almaden Research Center).
- Codd ha introdotto il modello relazionale e due linguaggi di interrogazioni di basi di dati: [Algebra relazionale](#) e [Calcolo relazionale](#).



2

2

Linguaggi di interrogazione per basi di dati relazionali

- **Dichiarativi**
 - specificano le proprietà del risultato ("che cosa")
- **Procedurali**
 - specificano le modalità di generazione del risultato ("come")

3

3

Linguaggi di interrogazione

- **Algebra relazionale**: procedurale
- **Calcolo relazionale**:
dichiarativo (teorico)
SQL (Structured Query Language):
parzialmente dichiarativo (reale)
- **QBE** (Query by Example):
dichiarativo (reale)

4

4

Che caratteristiche deve avere un linguaggio per basi di dati?

- Deve essere definito ad **alto livello** per garantire indipendenza fisica dei dati
- Deve essere abbastanza **espressivo** per permettere ad esprimere varie interrogazioni di una db che potrebbero servire in pratica
- Deve essere **implementabile in modo efficiente**

5

5

Algebra relazionale (AR)

- **5 operatori di base**
 - *Proiezione*
 - *Selezione*
 - *Unione*
 - *Differenza*
 - *Prodotto cartesiano*
- Combinando questi semplici operatori possiamo esprimere diverse tipologie di query (anche complesse)

6

6

Due versioni di AR:

- *unnamed* (notazione posizionale)
- *named* (notazione non posizionale)
 - In questa variante si aggiunge ai cinque operatori prima menzionati anche l'operatore di *Ridenominazione*

7

7

Operatori derivati

- Intersezione
- Join e le sue varianti
 - Theta-join
 - Join naturale
 - Join esterni
 - Semijoin
- Divisione

8

8

Operatori specifici, introdotti per gestire
le tabelle (relazioni):
proiezione e selezione

9

9

Paziente

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Fedi	Camilla	19/09/1976	femmina	CZ
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Vasto	Claudia	06/06/2013	femmina	CZ
Militare	Enzo	06/10/1945	maschio	RC

Restituire cognome e nome dei nostri pazienti

π (Paziente)
\$1 \$2

π (Paziente)
Cognome Nome

10

10

Paziente

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Fedi	Camilla	19/09/1976	femmina	CZ
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Vasto	Claudia	06/06/2013	femmina	CZ
Militare	Enzo	06/10/1945	maschio	RC

Restituire le provincie di residenza dei nostri pazienti

\mathcal{T}_5 (Paziente)

$\mathcal{T}_{ProvinciaR}$ (Paziente)

ProvinciaR
CS
CZ
VV
RC

11

11

Paziente

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Fedi	Camilla	19/09/1976	femmina	CZ
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Vasto	Claudia	06/06/2013	femmina	CZ
Militare	Enzo	06/10/1945	maschio	RC

Selezionare i dati solo su femmine:

\mathcal{C} (Paziente)

$\$4 = \text{'femmina'}$

$\mathcal{C}_{\text{Sesso} = \text{'femmina'}}$ (Paziente)

12

12

Paziente

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Fedi	Camilla	19/09/1976	femmina	CZ
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Vasto	Claudia	06/06/2013	femmina	CZ
Militare	Enzo	06/10/1945	maschio	RC

Selezionare i dati solo su femmine nate dal 2000 in poi:

6 (Paziente)
 $\$4 = \text{'femmina' AND}$
 $\$3 > \#31/12/1999\#$

in alternativa $\$3 \geq \#1/01/2000\#$

13

13

Paziente

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Fedi	Camilla	19/09/1976	femmina	CZ
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Vasto	Claudia	06/06/2013	femmina	CZ
Militare	Enzo	06/10/1945	maschio	RC

Selezionare i dati dei pazienti che non sono residenti in provincia di Catanzaro.

6 (Paziente)
 $\$5 \neq \text{'CZ'}$

in alternativa $\$5 \neq \text{'CZ'}$

14

14

Paziente

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Fedi	Camilla	19/09/1976	femmina	CZ
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Vasto	Claudia	06/06/2013	femmina	CZ
Militare	Enzo	06/10/1945	maschio	RC

Selezionare i dati dei pazienti che non sono residenti in provincia di Catanzaro.

Risultato:

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Militare	Enzo	06/10/1945	maschio	RC

15

15

Paziente

Cognome	Nome	DataN	Sesso	ProvinciaR
Cita	Francesco	06/06/2000	maschio	CS
Fedi	Camilla	19/09/1976	femmina	CZ
Gelo	Alfredo	19/10/1968	maschio	VV
Polletti	Carmine	28/06/2002	maschio	CS
Vasto	Claudia	06/06/2013	femmina	CZ
Militare	Enzo	06/10/1945	maschio	RC

Restituire **cognome e nome dei pazienti che non sono residenti in provincia di Catanzaro.**

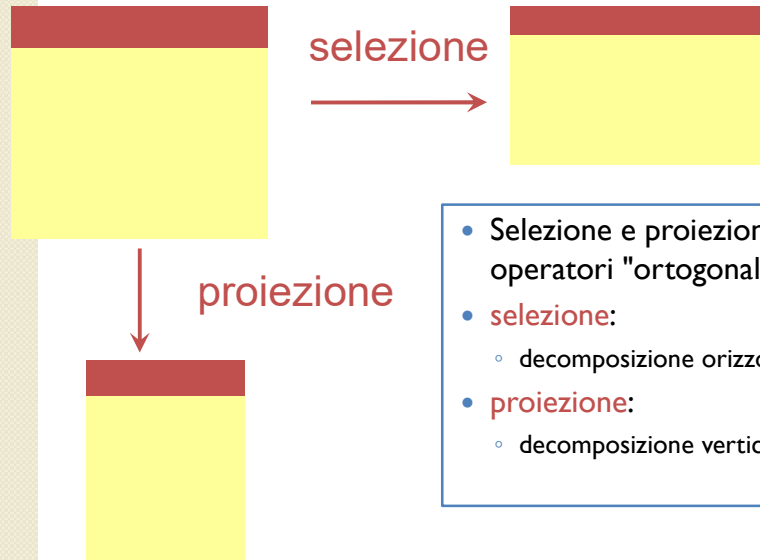
$\mathcal{G}_1 \left(\begin{matrix} \text{Paziente} \\ \$5! = \text{"CZ"} \end{matrix} \right)$
 $\$1 \2

~~$\mathcal{G}_1 \left(\begin{matrix} \text{Paziente} \\ \text{Provincia} \end{matrix} \right)$~~
~~Provincia~~ ~~Cognome~~ ~~Nome~~

16

16

Selezione e proiezione



- Selezione e proiezione sono operatori "ortogonali"
- **selezione:**
 - decomposizione orizzontale
- **proiezione:**
 - decomposizione verticale

17

17

Proiezione, sintassi e semantica

- sintassi

$\pi_{\text{ListaAttributi}}$ (Operando)

- semantica

- il risultato contiene le ennuple ottenute da tutte le ennuple dell'operando ristrette agli attributi nella lista

18

18

Proiezione

- operatore monadico
- produce un risultato che
 - ha parte degli attributi dell'operando
 - contiene ennuple cui contribuiscono tutte le ennuple dell'operando

19

19

Cardinalità delle proiezioni

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

20

20

Selezione, sintassi e semantica

- sintassi

$\sigma_{\text{Condizione}}$ (*Operando*)

- *Condizione*: espressione booleana (come quelle dei vincoli di ennupla)

- semantica

- il risultato contiene le ennuple dell'operando che soddisfano la condizione

21

21

Selezione

- operatore monadico
- produce un risultato che
 - ha lo stesso schema dell'operando
 - contiene un sottoinsieme delle ennuple dell'operando,
 - quelle che soddisfano una condizione

22

22

- combinando selezione e proiezione, possiamo estrarre informazioni da **una** relazione
- non possiamo però correlare informazioni presenti in relazioni diverse, né informazioni in ennuple diverse di una stessa relazione

23

23

Classici Operatori insiemistici:
unione e differenza
(anche *intersezione*, che in AR
è un operatore derivato)

24

24

Operatori insiemistici

- le relazioni sono insiemi
 - (nelle relazioni non devono essere enuple duplicate)
- i risultati debbono essere relazioni
- è possibile applicare unione, intersezione, differenza solo a relazioni che hanno la stessa struttura, cioè che sono definite sugli stessi attributi

25

25

Esempio

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

1. Visualizzare laureati e specialisti.
2. Visualizzare laureati che sono specialisti.
3. Visualizzare laureati che non sono specialisti.

26

26

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

27

27

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

28

28

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

29

29

Esempio di tabelle con struttura simile, ma che hanno attributi nominati diversamente

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Creare una sola relazione che riporta i genitori e i loro figli:

- AR posizionale (*unnamed*): **Paternità** \cup **Maternità**
non considera nomi delle colonne
- AR non posizionale (*named*): \cup **non è applicabile**

30

30

Ridenominazione

- operatore monadico (con un argomento)
- "modifica lo schema" lasciando inalterata l'istanza dell'operando

31

31

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

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

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

32

32

Paternità		$\rho_{\text{Genitore} \leftarrow \text{Padre}}$ (Paternità)	
Padre	Figlio	Genitore	Figlio
Adamo	Abele	Adamo	Abele
Adamo	Caino	Adamo	Caino
Abramo	Isacco	Abramo	Isacco
Maternità		$\rho_{\text{Genitore} \leftarrow \text{Madre}}$ (Maternità)	
Madre	Figlio	Genitore	Figlio
Eva	Abele	Eva	Abele
Eva	Set	Eva	Set
Sara	Isacco	Sara	Isacco

33

$\rho_{\text{Genitore} \leftarrow \text{Padre}}$ (Paternità)		$\rho_{\text{Genitore} \leftarrow \text{Padre}}$ (Paternità)	
Genitore	Figlio	$\rho_{\text{Genitore} \leftarrow \text{Madre}}$ (Maternità)	
Adamo	Abele		
Adamo	Caino	$\rho_{\text{Genitore} \leftarrow \text{Madre}}$ (Maternità)	
Abramo	Isacco		
$\rho_{\text{Genitore} \leftarrow \text{Madre}}$ (Maternità)		$\rho_{\text{Genitore} \leftarrow \text{Madre}}$ (Maternità)	
Genitore	Figlio	Genitore	Figlio
Eva	Abele	Adamo	Abele
Eva	Set	Adamo	Caino
Sara	Isacco	Abramo	Isacco
		Eva	Abele
		Eva	Set
		Sara	Isacco

34

Impiegati

Cognome	Ufficio	Stipendio
Rossi	Roma	55
Neri	Milano	64

Operai

Cognome	Fabbrica	Salario
Bruni	Monza	45
Verdi	Latina	55

$\rho_{\text{Sede, Retribuzione}} \leftarrow \text{Ufficio, Stipendio (Impiegati)}$
 $\rho_{\text{Sede, Retribuzione}} \leftarrow \text{Fabbrica, Salario (Operai)}$

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

35

35

Come correlare le informazioni presenti in relazioni diverse?

Medico

IdM
Cognome
Nome

CartellaClinica

Id
Paziente
Medico
DataR
ValoriF

Paziente

IdP
Cognome
Nome
DataN
Sesso
ProvinciaR

36

36

Come correlare le informazioni presenti in relazioni diverse?

Operatore base: di **prodotto cartesiano**

A	B		C	D	=	A	B	C	D
a1	b1	x	b1	d1	=	a1	b1	b1	d1
a2	b2		b2	d2		a1	b1	b2	d2
						a2	b2	b1	d1
						a2	b2	b2	d2

37

37

Come correlare le informazioni presenti in relazioni diverse?

Operatore derivato: **join** (\bowtie) e le sue varianti

A	B		C	D	=	A	B	C	D
a1	b1	$\bowtie_{B=C}$	b1	d1	=	a1	b1	b1	d1
a2	b2		b2	d2		a2	b2	b2	d2

38

38

Come correlare le informazioni presenti in relazioni diverse?

Operatore base: di **prodotto cartesiano**

Operatori derivati: **join** e le sue varianti

39

39

Join naturale (intuizione):

- Utilizza attributi comuni per correlare le relazioni
- Nel risultato non vengono duplicati le colonne con lo stesso nome

Busta		Compito	
Numero	Candidato	Numero	Voto
1	Mario Rossi	1	25
2	Nicola Russo	2	13
3	Mario Bianchi	3	27

Busta ⋈ Compito

Numero	Candidato	Voto
1	Mario Rossi	25
2	Nicola Russo	13
3	Mario Bianchi	27

40

40

Join naturale

- operatore binario (generalizzabile)
- produce un risultato
 - sull'unione degli attributi degli operandi
 - con ennuple costruite ciascuna a partire da una ennupla di ognuno degli operandi
- $R_1(X_1), R_2(X_2)$
- $R_1 \bowtie R_2$ è una relazione su $X_1 \cup X_2$

$\{ t \text{ su } X_1 \cup X_2 \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2 \text{ con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$

41

41

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 ennupla contribuisce al risultato:
 - join **completo**

42

42

Un 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

43

43

Un join vuoto

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

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

44

44

Un join completo, con $n \times m$ ennuple

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

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

45

45

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 esiste vincolo di integrità referenziale fra **B** (in R_1) e R_2 (in tal caso **B** è chiave in R_2):

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

- se **B** è chiave in R_2 (e non sappiamo se esiste vincolo di integrità referenziale fra **B** (in R_1) e R_2):

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

46

46

Join, possibile perdita dei dati

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

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

- alcune ennuple non contribuiscono al risultato: vengono "tagliate fuori"

47

47

Join esterno

- Il join **esterno** estende, con valori nulli, le ennuple che verrebbero tagliate fuori da un join (**interno**)
- Esiste in tre versioni:
 - **Sinistro (left join)** ⋈ :
mantiene tutte le ennuple del primo operando, estendendole con valori nulli, se necessario
 - **Destro (right join)** ⋈ :
... del secondo operando ...
 - **Completo (full join)** ⋈ :
... di entrambi gli operandi ...

48

48

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

Impiegati ⌋ Reparti		
Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

49

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

Impiegati ⌋ Reparti		
Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

50

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

Impiegati ⋈ Reparti

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

51

Semijoin (\bowtie)(\bowtie)	
• Operatore su due relazioni $R_1(X_1), R_2(X_2)$	
• Left semijoin: \bowtie	
◦ Restituisce una relazione su X_1 , con le ennuple di R_1 che contribuiscono al join con R_2	
◦ $R_1 \bowtie R_2$ è una relazione su X_1	
$\{ t \mid t \in R_1 \text{ ed esiste } t_2 \in R_2$	
$\text{con } t[X_1 \cap X_2] = t_2[X_1 \cap X_2] \}$	
◦ $R_1 \bowtie R_2 = \pi_{X_1}(R_1 \bowtie R_2)$	
• Right semijoin: \bowtie	

52

Left Semijoin

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

Impiegati \bowtie Reparti

Impiegato	Reparto
Neri	B
Bianchi	B

53

53

Join e proiezioni

- $R_1(X_1), R_2(X_2)$

$$\pi_{X_1}(R_1 \bowtie R_2) \subseteq R_1$$

- $R(X), X = X_1 \cup X_2$

$$(\pi_{X_1}(R)) \bowtie (\pi_{X_2}(R)) \supseteq R$$

54

54

Join e proiezioni

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

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Impiegato	Reparto	Reparto	Capo
Neri	B	B	Mori
Bianchi	B		

$$\pi_{X_1}(R_1 \bowtie R_2) \subseteq R_1$$

55

55

Proiezioni e join

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Bruni
Verdi	A	Bini

Impiegato	Reparto	Reparto	Capo
Neri	B	B	Mori
Bianchi	B	B	Bruni
Verdi	A	A	Bini

$$(\pi_{X_1}(R)) \bowtie (\pi_{X_2}(R)) \supseteq R$$

Impiegato	Reparto	Capo
Neri	B	Mori
Neri	B	Bruni
Bianchi	B	Mori
Bianchi	B	Bruni
Verdi	A	Bini

56

56

Prodotto cartesiano

- un join naturale su relazioni senza attributi in comune coincide con il prodotto cartesiano
- contiene sempre un numero di ennuple pari al prodotto delle cardinalità degli operandi (le ennuple sono tutte combinabili)

57

57

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

58

58

Theta-join, equi-join

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

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

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

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

- La condizione **C** è spesso una congiunzione (**AND**) di atomi di confronto $A_1 \vartheta A_2$ dove ϑ è uno degli operatori di confronto ($=, >, <, \dots$)
- Se l'operatore di confronto nel theta-join è sempre l'uguaglianza ($=$) allora si parla di **equi-join**

59

59

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

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

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

60

60

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

Impiegati ⋈ Reparti

61

Join naturale ed equi-join

Impiegati		Reparti	
Impiegato	Reparto	Reparto	Capo

Impiegati ⋈ Reparti =

$$\pi_{\text{Impiegato, Reparto, Capo}} [(\text{Impiegati} \bowtie_{\text{Reparto=Codice}} \rho_{\text{Codice} \leftarrow \text{Reparto}} (\text{Reparti}))]$$

62

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"

63

63

Un'equivalenza importante

- Push selections (se A è attributo di R_2)
$$\sigma_{A=10}(R_1 \bowtie R_2) = R_1 \bowtie \sigma_{A=10}(R_2)$$
- Riduce in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione)

64

64

Selezione con valori nulli

Impiegati

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

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

- la condizione atomica è vera solo per valori non nulli

65

65

Un risultato non desiderabile

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

- Perché? Perché le selezioni vengono valutate separatamente!
- Ma anche

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

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

66

66

Selezione con valori nulli: soluzione

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

- 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)

67

67

- Quindi:

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

68

68

Viste (relazioni derivate)

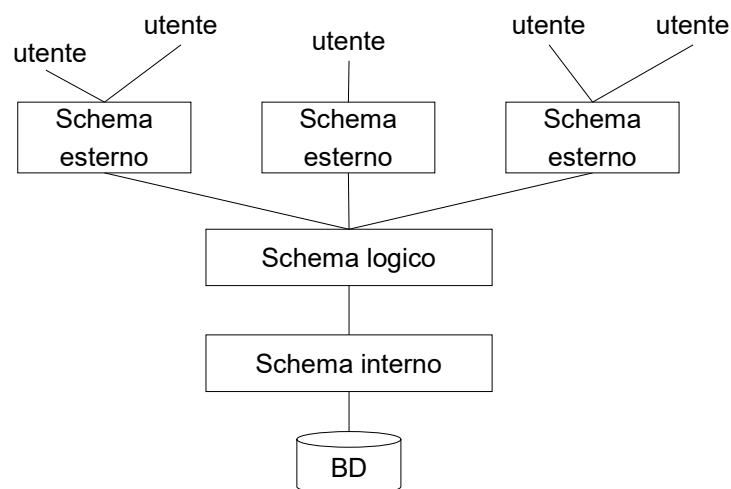
- Rappresentazioni diverse per gli stessi dati (schema esterno)
- **Relazioni derivate:**
 - relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- **Relazioni di base:** contenuto autonomo
- Le relazioni derivate possono essere definite su altre derivate, ma ...



69

69

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



70

70

Viste, esempio

Afferenza	Direzione	
	Impiegato	Reparto
	Rossi	A
	Neri	B
	Bianchi	B

Direzione	
Reparto	Capo
A	Mori
B	Bruni

- una vista:

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

71

71

Viste virtuali e materializzate

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

72

72

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



73

73

Viste virtuali

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

74

74

Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

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

viene eseguita come

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

75

75

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 è 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

76

76

Viste e aggiornamenti, attenzione

Afferenza		Direzione	
Impiegato	Reparto	Reparto	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Verdi	A	C	Bruni

Supervisione	Impiegato	Capo
	Rossi	Mori
	Neri	Bruni
	Verdi	Mori

- Vogliamo inserire, nella vista, il fatto che Lupi ha come capo Bruni; oppure che Belli ha come capo Falchi; come facciamo?

77

77

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!
- Ben pochi aggiornamenti sono ammissibili sulle viste

78

78

Operatore di divisione

- Studenti che hanno sostenuto **tutti** gli esami sostenuti dallo studente Mario Rossi.
- Studenti iscritti a **tutti** i corsi tenuti dal Prof. Neri Mario.
- Clienti che hanno comprato **tutti** i tipi di pesto di Giovanni Rana (cioè avente marca 'Rana').
- Utenti che hanno visionato **ogni** film con attore Alberto Sordi su Netflix.

Definition: Date la relazione R di arità r e la relazione S di arità s , con $r > s$.

Il **quoziente (o divisione) $R \div S$** è la relazione di arità $r - s$ che consiste di tutte le tuple (a_1, \dots, a_{r-s}) tali che per ogni tupla (b_1, \dots, b_s) in S, abbiamo $(a_1, \dots, a_{r-s}, b_1, \dots, b_s)$ in R.

79

79

Operatore di divisione

- Studenti che hanno sostenuto tutti gli esami sostenuti dallo studente Mario Rossi.
 - Supponiamo di aver definito le viste:
 - **R(Matricola, CodC)** che riporta per ciascun studente CodC (codice corso) per il quale ha sostenuto esame
 - **S(CodC)** che riporta CodS di tutti gli esami sostenuti dallo studente Mario Rossi
 - Allora il risultato desiderato si ottiene facendo **$R \div S$** che restituisce solo quei valori del primo campo di R (campo Matricola) per i quali ci sono riportati tutti valori di CodC che troviamo nella relazione S.

80

80

Operatore di divisione

R		S	$R \div S$
Matricola	CodC	CodC	Matricola
123	1	1	123
123	2	2	
123	3		
124	1		
124	3		



$R \div S = \pi_{\text{Matricola}}(R) - \text{"tuples in } \pi_{\text{Matricola}}(R) \text{ per i quali manca qualche valore di CodC che troviamo nella relazione } S"$

$$= \pi_{\text{Matricola}}(R) - \pi_{\text{Matricola}} [\pi_{\text{Matricola}}(R) \times S - R].$$

81