

*Corso di Laurea in Ingegneria Gestionale
Sapienza Università di Roma*

Corso di Basi di Dati

A.A. 2019/2020

3 – Algebra Relazionale

Tiziana Catarci

I linguaggi di interrogazione e aggiornamento

- ▶ Le **basi di dati** vengono utilizzate per rappresentare le informazioni di interesse per applicazioni che **gestiscono dati**.
- ▶ L'utilizzo di linguaggi per la specifica delle operazioni di **interrogazione** ed **aggiornamento** sui dati stessi è una **componente essenziale** delle basi di dati e quindi di ciascun modello dei dati.

Linguaggi di interrogazione relazionale

- ▶ **Linguaggi di interrogazione (LI)**: permettono il **reperimento** di dati da una base di dati.
- ▶ Il modello relazionale supporta LI semplici e potenti:
 - ▶ Forte base formale basata sulla logica.
 - ▶ Ottimizzazione.
- ▶ Linguaggi di interrogazione **diversi da** linguaggi di programmazione!
 - ▶ I LI non sono necessariamente “Turing completi”.
 - ▶ I LI non sono fatti per essere usati in calcoli complessi.
 - ▶ I LI supportano un accesso semplice ed efficiente a grandi insiemi di dati.

Linguaggi formali di interrogazione relazionale

- ▶ Due linguaggi di interrogazione matematici formano la base per i linguaggi “reali” (es. SQL) e per l’implementazione:
 - ▶ **Algebra relazionale**: linguaggio procedurale (specifica come viene generato il risultato), utilissimo per rappresentare i piani di esecuzione.
 - ▶ **Calcolo relazionale**: permette agli utenti di descrivere le **proprietà del risultato**, piuttosto che il modo in cui calcolarlo (non operativo, dichiarativo). Questo linguaggio è basato sul calcolo dei predicati del primo ordine.
- ▶ Capire l’algebra e il calcolo è la chiave per la comprensione dell’SQL e dell’elaborazione delle interrogazioni!

L'algebra relazionale

- ▶ **Algebra Relazionale** : linguaggio **procedurale** (specifica **come** *viene generato* il risultato) di **interrogazione** associato al modello relazionale.
- ▶ Attraverso l'algebra relazionale, le operazioni complesse vengono specificate descrivendo il **procedimento da seguire** per ottenere la soluzione.
- ▶ L'algebra relazionale è costituita da un insieme di operatori (unari o binari):
 - ▶ definiti su relazioni;
 - ▶ che producono una relazione come risultato;
 - ▶ e possono essere composti per formulare interrogazioni complesse.

L'algebra relazionale

- ▶ **Operatori** dell'algebra relazionale:
 - ▶ Insiemistici (unione, intersezione, differenza)
 - ▶ Ridenominazione
 - ▶ Prodotto cartesiano
 - ▶ Selezione
 - ▶ Proiezione
 - ▶ Join (naturale, equi-join, condizionale o theta-join)
 - ▶ Divisione

Operatori insiemistici

- ▶ Una relazione è un *insieme di tuple omogenee* (cioè definite sugli *stessi attributi*).
- ▶ E' possibile applicare gli operatori insiemistici **solo** a relazioni per cui valga la proprietà di **compatibilità rispetto all'unione**:
 - ▶ le relazioni in ingresso hanno lo stesso numero di campi.
 - ▶ campi corrispondenti delle due relazioni, presi in ordine da sinistra a destra, hanno lo stesso dominio (la compatibilità non si verifica con i nomi degli attributi).
- ▶ Se due relazioni R1 e R2 sono compatibili rispetto all'unione, ma hanno i nomi degli attributi differenti, la relazione che si ottiene come risultato eredita per convenzione i nomi di R1.



ATTENZIONE : Si rischia di ottenere un insieme di tuple disomogenee...

Operatori insiemistici - Unione

Laureati

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38

Dirigenti

Matricola	Cognome	Età
9297	Neri	56
7432	Neri	39
9824	Verdi	38

U

Laureati U Dirigenti

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38
9297	Neri	56

- L'unione di due relazioni r_1 e r_2 è indicata con $r_1 \cup r_2$ e contiene tutte le tuple (**prese una sola volta**) presenti in r_1 , in r_2 oppure in entrambe.
- Lo schema del risultato è identico allo schema di r_1 .



Operatori insiemistici - Intersezione

Laureati

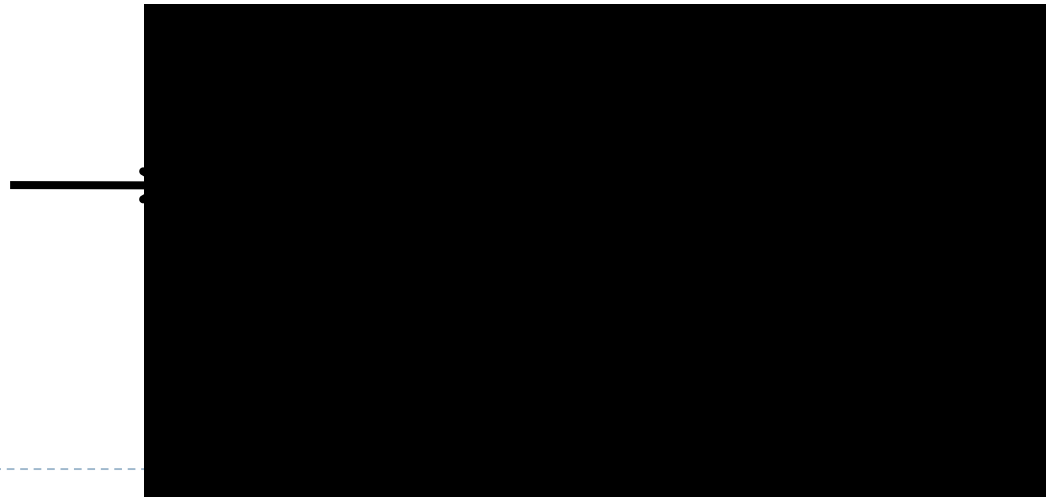
Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38

Dirigenti

Matricola	Cognome	Età
9297	Neri	56
7432	Neri	39
9824	Verdi	38



Matricola	Cognome	Età
7432	Neri	39
9824	Verdi	38



Operatori insiemistici - Differenza

Laureati

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38

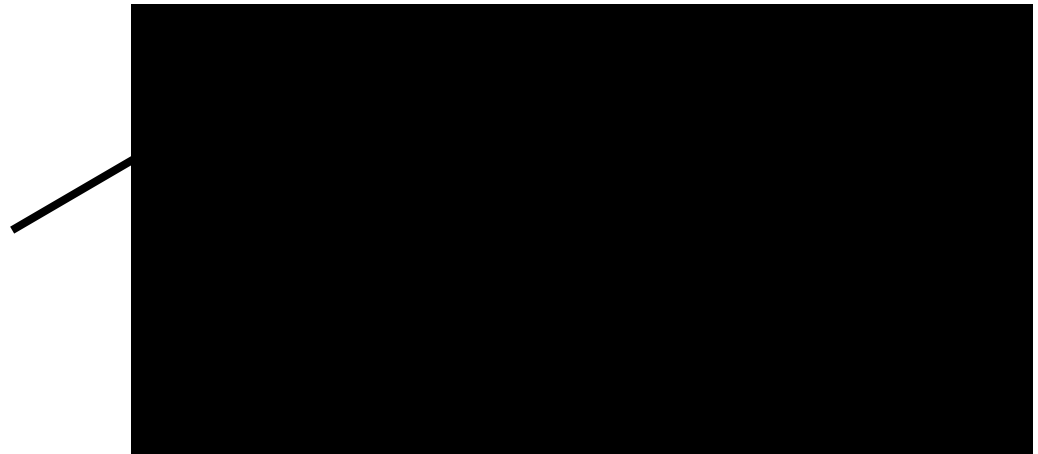
Dirigenti

Matricola	Cognome	Età
9297	Neri	56
7432	Neri	39
9824	Verdi	38



Laureati - Dirigenti

Matricola	Cognome	Età
7274	Rossi	37



Un'unione sensata...ma impossibile

Paternità

Padre	Figlio
Francesco	Maurizio
Francesco	Carla
Giuseppe	Maria

U

Maternità

Madre	Figlio
Adriana	Maurizio
Adriana	Paolo
Eleonora	Maria

Paternità U Maternità ?? →

Padre	Figlio
Francesco	Maurizio
Francesco	Carla
Giuseppe	Maria
Adriana	Maurizio
Adriana	Paolo
Eleonora	Maria

L'unione rispetta la proprietà di compatibilità rispetto all'unione, ma il risultato è disomogeneo...

La ridenominazione 1\3

- ▶ Sintassi dell'operatore : $\rho(\mathbf{R}(\mathbf{F}),\mathbf{E})$
 - ▶ \mathbf{E} : espressione arbitraria di algebra relazionale (*es. Paternità U Maternità*).
 - ▶ \mathbf{R} : istanza di relazione che rappresenta il risultato dell'esecuzione di \mathbf{E} (contiene le stesse tuple di \mathbf{E} , ma con alcuni campi rinominati).
 - ▶ \mathbf{F} : lista di “ridenominazione” nella forma:
 - ▶ *Vecchio nome* \rightarrow *Nuovo nome*
 - ▶ *Posizione* \rightarrow *Nuovo nome*
- ▶ $\rho(\mathbf{C}(\mathbf{Padre} \rightarrow \mathbf{Genitore}), \mathbf{Paternità} \cup \mathbf{Maternità})$
oppure
- ▶ $\rho(\mathbf{C}(1 \rightarrow \mathbf{Genitore}), \mathbf{Paternità} \cup \mathbf{Maternità})$

Genitore	Figlio
Francesco	Maurizio
Francesco	Carla
Giuseppe	Maria
Adriana	Maurizio
Adriana	Paolo
Eleonora	Maria

$\mathbf{C}(\mathbf{Genitore:String}, \mathbf{Figlio:String})$



La ridenominazione 2\3

Per ottenere la relazione Genitore-Figlio, si può anche scrivere alternativamente...

Paternità

Padre	Figlio
Francesco	Maurizio
Francesco	Carla
Giuseppe	Maria

$\rho((Padre \rightarrow Genitore), Paternità)$



Genitore	Figlio
Francesco	Maurizio
Francesco	Carla
Giuseppe	Maria

Maternità

Madre	Figlio
Adriana	Maurizio
Adriana	Paolo
Eleonora	Maria

$\rho((Madre \rightarrow Genitore), Maternità)$



Genitore	Figlio
Adriana	Maurizio
Adriana	Paolo
Eleonora	Maria

$\rightarrow \rho((Padre \rightarrow Genitore), Paternità) \cup \rho((Madre \rightarrow Genitore), Maternità) \leftarrow$

La ridenominazione 3\3

Impiegati

Cognome	Ufficio	Stipendio
Rossi	Roma	55
Neri	Milano	64

Operai

Cognome	Fabbrica	Salario
Bruni	Monza	45
Verdi	Latina	55

$\rho((Ufficio \rightarrow Sede, Stipendio \rightarrow Retribuzione), Impiegati)$

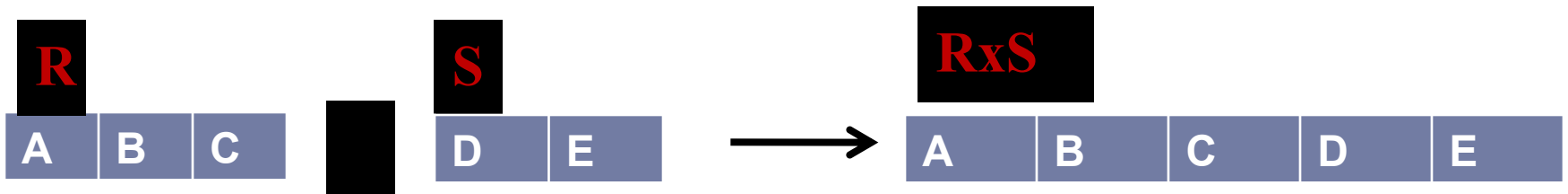
U

$\rho((Fabbrica \rightarrow Sede, Salario \rightarrow Retribuzione), Operai)$

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

Il prodotto cartesiano 1\3

- ▶ E' anch'esso un operatore insiemistico.
- ▶ Per essere utilizzato **non richiede che sia valida** la proprietà di *compatibilità rispetto all'unione*.
- ▶ Sintassi dell'operatore : **R x S**
- ▶ Restituisce un'istanza di relazione il cui schema contiene tutti i campi di R (nell'ordine originale) seguiti da tutti i campi di S (nell'ordine originale).



Il prodotto cartesiano 2\3

- contiene una tupla $\langle r,s \rangle$ per ogni coppia di tuple $r \in R$ e $s \in S$ (perciò, ciascuna riga di R è accoppiata con ciascuna di riga di S).

ATTENZIONE : Se due relazioni hanno degli attributi con nomi in comune, i campi corrispondenti nel prod.cartesiano non hanno nome (per convenzione) e ci si può riferire a loro solo attraverso la posizione che occupano nello schema...

Impiegati

Impiegato	Codice
Rossi	A
Neri	B
Bianchi	B

Reparti

Capo	Codice
Mori	A
Bruni	B

Impiegati Reparti

Impiegato	(Codice)	Capo	(Codice)
Rossi	A	Mori	A
Rossi	A	Bruni	B
Neri	B	Mori	A
Neri	B	Bruni	B
Bianchi	B	Mori	A
Bianchi	B	Bruni	B

→ conflitto di nomi



si può risolvere con una ridenominazione...

Il prodotto cartesiano 3\3

$\rho(\text{nuovaRelazione}(2 \rightarrow \text{CodImpiegato}, 4 \rightarrow \text{CodReparto}), \text{Impiegati} \times \text{Reparti})$



In questo caso è necessario sfruttare la notazione posizionale...

nuovaRelazione

Impiegato	CodImpiegato	Capo	CodReparto
Rossi	A	Mori	A
Rossi	A	Bruni	B
Neri	B	Mori	A
Neri	B	Bruni	B
Bianchi	B	Mori	A
Bianchi	B	Bruni	B

→ conflitto di nomi risolto

Selezione 1\2

- ▶ Operatore unario per **selezionare righe da una relazione**.
- ▶ **Sintassi** : $\sigma_{\text{condizione}}(\mathbf{R})$
 - ▶ condizione di selezione = espressione booleana del tipo:
 - ▶ *attributo* **OP** *costante* oppure *attributo1* **OP** *attributo2*
 - ▶ **OP** = { < , <= , > , >= , = , ≠ }
 - ▶ Espandibile con i connettivi logici \mathbf{V} , $\mathbf{\wedge}$
- ▶ **Semantica** :
 - ▶ La relazione risultato ha gli stessi attributi della relazione in ingresso e contiene quelle tuple che soddisfano la condizione specificata (senza duplicati).

Esempio

Tutti gli Impiegati con codice = A

$\sigma_{\text{Codice}='A'}(\text{Impiegati})$

Impiegati

Impiegato	Codice
Rossi	A
Neri	B
Bianchi	B

Impiegato	Codice
Rossi	A

Selezione 2\2

▶ Esempio

- ▶ Tutti i Laureati con Cognome = 'Rossi' ed Età > 37

$\sigma_{\text{Cognome}='Rossi' \wedge \text{Età}>37}(\text{Laureati})$

Laureati

Matricola	Cognome	Università	Età
7274	Rossi	Roma	37
7432	Rossi	Roma	39
9824	Roma	Roma	38

Matricola	Cognome	Università	Età
7432	Rossi	Roma	39

▶ Esempio

- ▶ Tutti i Laureati con lo stesso Cognome dell'Università in cui hanno studiato

$\sigma_{\text{Cognome}=\text{Università}}(\text{Laureati})$

Matricola	Cognome	Università	Età
9824	Roma	Roma	38

Proiezione

- ▶ Operatore unario per **estrarre colonne da una relazione**.
- ▶ Sintassi : $\Pi_{\text{lista_di_Attributi}}(\mathbf{R})$
- ▶ Semantica :
 - ▶ la relazione risultato ha i soli attributi contenuti in *lista_di_attributi* e contiene le tuple ristrette agli attributi nella lista (senza duplicati).

▶ Esempio

- ▶ **Cognome e Università di tutti i laureati**

Laureati

Matricola	Cognome	Università	Età
7274	Rossi	Roma	37
7432	Rossi	Roma	39
9824	Roma	Roma	38

ATTENZIONE :
gli attributi non contenuti
nella lista vengono
proiettati fuori.

$\Pi_{\text{Cognome, Università}}(\mathbf{Laureati})$

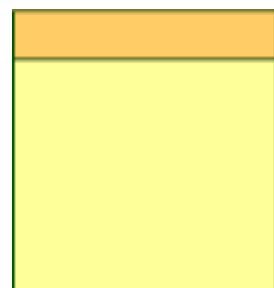


Cognome	Università
Rossi	Roma
Roma	Roma

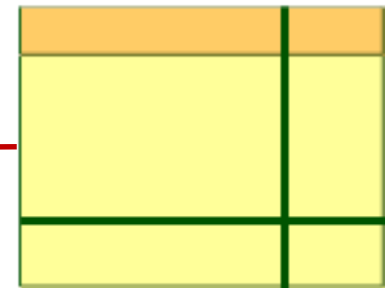
Selezione e Proiezione 1\2

- ▶ Gli operatori di selezione e proiezione svolgono funzioni che potremmo definire **ortogonali** (o complementari).
- ▶ Sono entrambe definite su un operando (una singola istanza di relazione) e producono come risultato una porzione dell'operando.

Proiezione: Produce un risultato cui contribuiscono tutte le tuple, ma su un sottoinsieme degli attributi. Pertanto genera **decomposizione verticale**.

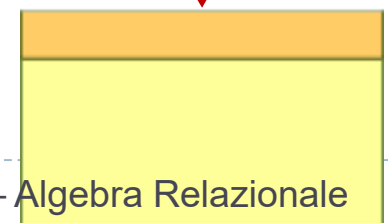


proiezione



selezione

Selezione: Produce un sottoinsieme delle tuple su tutti gli attributi. Pertanto genera **decomposizione orizzontale**.



Selezione e Proiezione 2\2

- ▶ Gli operatori di selezione e proiezione si possono combinare efficientemente.

- ▶ **Esempio**

- ▶ **Cognome e Università dei laureati con più di 37 anni**

Laureati

Matricola	Cognome	Università	Età
7274	Rossi	Roma	37
7432	Rossi	Roma	39
9824	Roma	Roma	38

Cognome	Università
Rossi	Roma
Roma	Roma

$$\Pi_{\text{Cognome, Università}} (\sigma_{\text{Età} > 37} (\text{Laureati}))$$

Join condizionale (o theta-join)

▶ **JOIN** = Operatore molto usato per **combinare informazioni** tra due o più relazioni.

▶ **JOIN condizionale:**

▶ Sintassi e semantica:

▶ $R_1 \bowtie_c R_2$ equivale a $\sigma_c(R_1 \times R_2)$

▶ La *condizione di JOIN* è identica (nella forma) alla *condizione di selezione*.

Un ***JOIN condizionale*** tra R_1 e R_2 è fondamentalmente una **SELECT** applicata sul prodotto cartesiano tra R_1 e R_2

R₁

Impiegato	Stipendio
Rossi	20
Neri	20
Bianchi	30

R₂

Capo	Stipendio
Mori	10
Bruni	20

R₁ $\bowtie_{R_1.Stipendio > 20}$ **R₂**

Impiegato	(Stipe ndio)	Capo	(Stipe ndio)
Bianchi	30	Mori	10
Bianchi	30	Bruni	20

Equi-Join

- ▶ Un caso speciale di JOIN condizionale dove la condizione è composta solo da uguaglianze (eventualmente connesse da \wedge).

R₁

Impiegato	Stipendio
Rossi	20
Neri	20
Bianchi	30

R₂

Capo	Stipendio
Mori	10
Bruni	20

R₁



R₁.Stipendio=R₂.Stipendio

R₂

Impiegato	Stipendio	Capo
Rossi	20	Bruni
Neri	20	Bruni

- ▶ Lo schema di un risultato di un EQUI-JOIN contiene **tutti gli attributi di R₁** e quegli attributi di R₂ che **non compaiono** nella condizione di JOIN. Quindi c'è solo una copia dei campi per i quali è specificata l'uguaglianza.
 - ▶ Solo un attributo Stipendio appare nel risultato (mantenerli entrambi sarebbe ridondante).

Join naturale 1\2

- ▶ E' un EQUI-JOIN in cui le uguaglianze sono specificate su tutti gli attributi aventi lo stesso nome in R1 e R2.
- ▶ Le tuple del risultato sono ottenute combinando le tuple degli operandi con **valori uguali sugli attributi comuni**.

R₁

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

R₂

Capo	Reparto
Mori	A
Bruni	B

R₁ ⋈ R₂

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

- ▶ Se le due relazioni non hanno attributi in comune, il JOIN naturale si riduce ad essere un semplice prodotto cartesiano.

Join naturale 2\2

▶ Esempio : Un JOIN non completo

R₁

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

R₂

Capo	Reparto
Mori	B
Bruni	C

R₁ ⋈ R₂

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

▶ Esempio : Un JOIN vuoto

R₁

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

R₂

Capo	Reparto
Mori	D
Bruni	C

R₁ ⋈ R₂

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

Esercizio - Massimo e Minimo Assoluto

- ▶ Dato $R(\text{Impiegato}, \text{Stipendio})$, trovare il minimo stipendio in R .

R

Impiegato	Stipendio
Rossi	10
Neri	20
Bianchi	30

Esercizio - Massimo e Minimo Assoluto

- ▶ Dato $R(\text{Impiegato}, \text{Stipendio})$, trovare il minimo stipendio in R .

Nella seconda parte dell'interrogazione vengono trovati tutti quei valori di **Stipendio** che non sono il minimo. Per far questo viene fatto un join tra la relazione R e se stessa, ridenominando tutti gli attributi della copia di R . La condizione di join seleziona quei valori dell'attributo **Stipendio** che sono maggiori di almeno un valore della sua controparte rinominata.

$\Pi_{\text{Stipendio}}(R) -$

$\Pi_{\text{Stipendio}}(R \bowtie_{\text{Stipendio} > \text{Stip}} R)$

$(\rho(R1(1 \rightarrow \text{Imp}, 2 \rightarrow \text{Stip}), R))$

In questo modo vengono mantenute tutte le tuple tranne quella in cui l'attributo **Stipendio** assume il valore minore. Quindi per il "principio di complementarità" sottraendo dall'insieme iniziale l'insieme delle tuple dove Stipendio non è il minimo, otteniamo proprio il valore minimo che cercavamo.

R

Impiegato	Stipendio
Rossi	10
Neri	20
Bianchi	30

→

Stipendio
10

Esercizio - Massimo e Minimo Relativo

- ▶ Dato $R(\text{Studente}, \text{Esame}, \text{Voto})$, trovare il massimo voto per ogni studente in R .

R

Studente	Esame	Voto
Rossi	Analisi	18
Rossi	Basi di Dati	20
Bianchi	Analisi	30

Esercizio - Massimo e Minimo Relativo

- ▶ Dato $R(\text{Studente}, \text{Esame}, \text{Voto})$, trovare il massimo voto per ogni studente in R .

E' molto simile al caso precedente. Il join in questo caso seleziona tutti i valori minimi di **Voto** per ogni valore dell'attributo **Studente**.

$\Pi_{\text{Studente}, \text{Voto}}(R) -$

$\Pi_{\text{Studente}, \text{Voto}}(R \bowtie$

$\text{Voto} < \text{Vot AND Studente} = \text{Stud}$

$(\rho(R1(1 \rightarrow \text{Stud}, 2 \rightarrow \text{Exam}, 3 \rightarrow \text{Vot}), R)))$

R

Studente	Esame	Voto
Rossi	Analisi	18
Rossi	Basi di Dati	20
Bianchi	Analisi	30



Studente	Voto
Rossi	20
Bianchi	30

Esercizio - Omonimi

- ▶ **Dato $R(\text{Nome}, \text{Cognome}, \text{CF})$, trovare le persone omonime (stesso nome e cognome, ma diverso CF).**

R

Nome	Cognome	CF
Marco	Rossi	AAAAA
Marco	Rossi	BBBBBB
Andrea	Bianchi	CCCCC

Esercizio - Omonimi

- ▶ Dato $R(\text{Nome}, \text{Cognome}, \text{CF})$, trovare le persone omonime (stesso nome e cognome, ma diverso CF).

Viene ancora fatto un join tra la relazione R e se stessa, con però tutti gli attributi ridenominati. Il join permette di mantenere tutte quelle tuple in cui i valori degli attributi **Nome** e **Cognome** sono uguali alla loro controparte ridenominata, mentre il valore dell'attributo **CF** è diverso.

$\Pi_{\text{Nome}, \text{Cognome}, \text{CF}}(R \bowtie$

$\text{Nome}=\text{Nom} \text{ AND } \text{Cognome}=\text{Cog} \text{ AND } \text{CF} \neq \text{Cod}$

$(\rho(R1(1 \rightarrow \text{Nom}, 2 \rightarrow \text{Cog}, 3 \rightarrow \text{Cod}), R)))$

R

Nome	Cognome	CF
Marco	Rossi	AAAAA
Marco	Rossi	BBBBBB
Andrea	Bianchi	CCCCC



Nome	Cognome	CF
Marco	Rossi	AAAAA
Marco	Rossi	BBBBB

Divisione

- ▶ Non supportata come operatore primitivo, ma utile per alcuni tipi di interrogazione.
- ▶ Sia A una relazione con due campi, x e y ; sia B una relazione con il solo campo y :
- ▶ $A/B = \{\langle x \rangle \mid \text{per ogni } \langle y \rangle \in B, \langle x, y \rangle \in A\}$
 - ▶ cioè, A/B contiene tutte le tuple x tali che per ogni tupla y in B , ci sia una tupla xy in A .

ESEMPIO : Trovare gli **Impiegati** *che lavorano in tutti i reparti.*

R₁

Impiegato	Reparto
Rossi	A
Rossi	B
Bianchi	B

R₂

Reparto
A
B

R₁ / R₂

Impiegato
Rossi

Esempio di divisione A/B

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

pno
p2
p4

B2

pno
p1
p2
p4

B3

sno
s1
s2
s3
s4

A/B1

sno
s1
s4

A/B2

sno
s1

A/B3

Esprimere una divisione usando operatori di base

- ▶ La divisione non è un'operazione essenziale; solo un'utile scorciatoia.
 - ▶ Questo discorso varrebbe anche per i JOIN, ma dato il loro frequente utilizzo, i sistemi li implementano esplicitamente.
- ▶ Idea: per A/B calcolare tutti i valori x che non sono interdetti da qualche valore y in B .
 - ▶ Un valore x è 'interdetto' se associandogli valori y da B si ottiene una tupla xy che non è in A .
 - ▶ cioè, A/B contiene tutte le tuple x tali che per ogni tupla y in B , ci sia una tupla xy in A .
- ▶ Valori x interdetti: $\Pi_x((\Pi_x(A) \times B) - A)$
- ▶ A/B : $\Pi_x(A) - \text{tutte le tuple interdette}$

Condizioni di selezione e valori NULL

Impiegati

Matricola	Cognome	Università	Età
7274	Rossi	Roma	37
7432	Rossi	Roma	41
9824	Roma	Roma	NULL

$\sigma_{Eta > 40}(\text{Impiegati})$???

La condizione è vera solo per valori **NON NULLI**

- ▶ Per riferirsi a valori NULLI esistono forme apposite di condizioni:

IS NULL
IS NOT NULL

$\sigma_{Eta > 40} \vee Eta \text{ IS NULL}(\text{Impiegati})$

Matricola	Cognome	Università	Età
7432	Rossi	Roma	41
9824	Roma	Roma	NULL

Esempi – Relazioni di riferimento

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

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

Trovare *Matricola, Nome, Età, Stipendio* degli impiegati che guadagnano più di 40 milioni.

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



Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Esempio 2

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

Trovare *Matricola, Nome, Età* degli impiegati che *guadagnano più di 40 milioni*.

Matricola	Nome	Età
7309	Rossi	34
5698	Bruni	43
4076	Mori	45
8123	Lupi	46

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



Esempio 3 – 1\4

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

Trovare le *Matricole* dei capi degli impiegati che *guadagnano più di 40 milioni*.



Una *buona tecnica* per ricavare le espressioni dell'algebra relazionale è quella di *procedere per passi*, ricordando che **i risultati di interrogazioni intermedie sono sempre relazioni**.

Esempio 3 – 2\4

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

PASSO 1 : si cercano tutti gli impiegati che guadagnano più di 40 milioni.

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

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Esempio 3 – 3\4

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

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

PASSO 2 : si associano tutti gli impiegati che guadagnano più di 40 milioni ai loro capi, sfruttando un equi-join con $Supervisione.Impiegato = Matricola$

Supervisione

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

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

Nome	Età	Stipendio	Impiegato	Capo
Rossi	34	45	7309	5698
Bruni	43	42	5698	4076
Mori	45	50	4076	8123

Esempio 3 – 4\4

Supervisione \bowtie **Impiegato=Matricola** $\sigma_{Stipendio>40}$ (**Impiegati**)

Nome	Età	Stipendio	Impiegato	Capo
Rossi	34	45	7309	5698
Bruni	43	42	5698	4076
Mori	45	50	4076	8123

PASSO 3 : proietto la relazione ottenuta solo sull'attributo *Capo*.

Π_{capo} (**Supervisione** \bowtie **Impiegato=Matricola** $\sigma_{Stipendio>40}$ (**Impiegati**))

Capo
5698
4076
8123

Esempio 4

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

Trovare *nome* e *stipendio* dei capi degli impiegati che *guadagnano più di 40 milioni*.

$\Pi_{nome, stipendio}(\dots)$

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

$\Pi_{capo}(\text{Supervisione} \bowtie_{\text{Impiegato}=\text{Matricola}}$

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

Nome	Stipendio
Bruni	42
Mori	50
Lupi	60

Esempio 5

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

Si può sfruttare
l'operatore di
differenza...

Trovare le *matricole* dei capi i cui
impiegati guadagnano **tutti** più di 40
milioni.

$\Pi_{\text{capo}}(\text{Supervisione}) -$

$\Pi_{\text{capo}}(\text{Supervisione} \bowtie_{\text{Impiegato=Matricola}}$

$(\sigma_{\text{Stipendio} \leq 40}(\text{Impiegati}))$)

Capo
8123

Esempio 6

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

Trovare gli *impiegati* che *guadagnano più del proprio capo*, mostrando *matricola, nome e stipendio dell'impiegato e del capo*.

$\Pi_{\text{Matricola, Nome, Stipendio, MatrC, NomeC, StipC}}($

$\sigma_{\text{Stipendio} > \text{StipC}}($

$\rho((\text{Matricola} \rightarrow \text{MatrC}, \text{Nome} \rightarrow \text{NomeC}, \text{Stipendio} \rightarrow \text{StipC}, \text{Età} \rightarrow \text{EtàC}), \text{Impiegati})$

$\bowtie_{\text{MatrC}=\text{Capo}}$

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



MatrC	NomeC	StipC	Matricola	Nome	Stipendio
5698	Bruni	42	7309	Rossi	45