

# Capitolo 2 - Algebra Relazionale

---

Appunti di **Giuseppe Pitruzzella** - Corso di *Database* @ DMI, UniCt

## 1. Notazione

TABELLA O RELAZIONE		
Attributo 1	Attributo 2	Attributo 3
Tipo di Dato - Valore	Tipo di Dato - Valore	Tipo di Dato - Valore

Una riga di valori di una relazione è detta *tupla*. L'insieme delle righe di valori di una relazione è detto *corpo*.

## 2. Operatori

Notazione

Tabella o relazione		
Attributo o campo dato (dominio)	Attributo o campo dato (dominio)	Attributo o campo dato (dominio)
(Tipo di dato) Valore	(Tipo di dato) Valore	(Tipo di dato) Valore
(Tipo di dato) Valore	(Tipo di dato) Valore	(Tipo di dato) Valore

Testata o schema  
Tupla  
Corpo

Unione (U)

Impiegati		
Numero	Cognome	Età
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Direttori		
Numero	Cognome	Età
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Impiegati U Direttori		
Numero	Cognome	Età
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38
9297	O'Malley	56

Intersezione (∩)

Impiegati		
Numero	Cognome	Età
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Direttori		
Numero	Cognome	Età
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Impiegati ∩ Direttori		
Numero	Cognome	Età
7432	O'Malley	39
9824	Darkes	38

Differenza (-)

Impiegati		
Numero	Cognome	Età
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Direttori		
Numero	Cognome	Età
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Impiegati – Direttori		
Numero	Cognome	Età
7274	Robinson	37

Divisione (/)

Voli	
Codice	Data
001	01/07/2011
001	07/09/2011
001	10/10/2011
002	01/07/2011
002	10/10/2011
002	25/12/2011

Linee	
Codice	
001	
002	

Voli/Linee	
Data	
01/07/2011	
10/10/2011	

Data una tabella t1 con attributi a1 e a2, e una tabella t2 con attributi a1, la divisione t1/t2 restituisce gli attributi a2 in comune con le tuple di a1.

Ridenominazio  
ne (δ)  
[delta]

Paternità	
Padre	Figlio
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael

Maternità	
Madre	Figlio
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

$\delta_{Padre \rightarrow Genitore}(Paternità) \cup \delta_{Madre \rightarrow Genitore}(Maternità)$	
Genitore	Figlio
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

Prodotto  
cartesiano (×)

Impiegati	
Impiegato	Progetto
Smith	A
Black	A
Black	B

Progetti	
Codice	Nome
A	Venus
B	Mars

Impiegati × Progetti			
Impiegato	Progetto	Codice	Nome
Smith	A	A	Venus
Black	A	A	Venus
Black	B	A	Venus
Smith	A	B	Mars
Black	A	B	Mars
Black	B	B	Mars

Selezione (σλ)  
[sigma\_lambda]

Selezione

Impiegati			
Cognome	Nome	Età	Salario
Smith	Mary	25	5000
Black	Lucy	40	30000
Verdi	Nico	22	4500

$\sigma_{Età < 30 \wedge Salary > 4000}(Impiegati)$			
Cognome	Nome	Età	Salario
Smith	Mary	25	5000
Verdi	Nico	22	4500

Proiezione (π)

Proiezione

Impiegati			
Cognome	Nome	Dipartimento	Capo
Smith	Mary	Vendite	De Rossi

$\pi_{Cognome, Nome}(Impiegati)$	
Cognome	Nome
Smith	Mary

	Black Verdi	Lucy Nico	Vendite Personale	De Rossi Fox		Black Verdi	Lucy Nico	
--	----------------	--------------	----------------------	-----------------	--	----------------	--------------	--

### Natural-JOIN ( $\bowtie$ )

Le n-uple del risultato sono ottenute combinando le n-uple di Paternità e Maternità che hanno gli stessi valori negli attributi con lo stesso nome

Paternità		Maternità		Paternità $\bowtie$ Maternità		
Padre	Figlio	Madr e	Figlio	Padre	Figlio	Madre
Adam	Cain	Eve	Cain	Adam	Cain	Eve
Adam	Abel	Eve	Seth	Abhra m	Isaac	Sarah
Abraha m	Isaac	Sarah	Isaac	Abhra m	Ishmae l	Hagar
Abraha m	Ishmae l	Hagar	Ishmae l			

### Theta-JOIN ( $\bowtie_{x(f)y}$ )

Viene specificato un predicato per la selezione delle n-uple. Se  $f$  è una congiunzione di uguaglianze si parla di equi-JOIN.

Theta-Join

Imp		Compleanni		Impiegati $\bowtie_{Imp.Id < Compleanni.Id}$ Progetti			
Id	Progett o	Id	Giorno	Id	Progetto	Id	Giorno
2	A	2	10/10/7	22	A	58	11/12/80
2		2	5				
3	A	5	11/12/8	31	A	58	11/12/80
1		8	0				
5	C						
8							

Equi-Join

Impiegati		Progetti		Impiegati $\bowtie_{Progetti=Codice}$ Progetti			
Impiegat o	Progett o	Codic e	Nome	Impiegato	Progetto	Codice	Nome
Smith	A	A	Venu s	Smith	A	A	Venus
Black	A	B	Mars	Black	A	A	Venus
Black	B			Black	B	B	Venus

### Casi di JOIN

#### JOIN incompleto

Nel caso in cui alcuni valori tra gli attributi comuni non coincidono; alcune n-uple non partecipano al JOIN.

#### JOIN vuoto

Quando nessuna nupla trova il corrispettivo

### Natural Outer Join

Mantiene nel risultato le n-uple che non partecipano al JOIN. Gli attributi delle dangling n-uple (attributi penzolanti) vengono riempiti con NULL. Tre varianti: – Left: solo dangling n-uple del primo operando – Right: solo dangling n-uple del secondo operando – Full: n-uple da entrambi gli operandi.

$r_1$	
Impiegat o	Dipartiment o
Smith	Vendite
Black	Produzione
White	Produzione

$r_2$	
Dipartiment o	Capo
Produzione	Mori
Acquisizione	Brow n

$r_1 \bowtie_{LEFT} r_2$		
Imp.	Dip.	Capo
Smith	Vendite	NUL L
Black	Produzion e	Mori

$r_1 \bowtie_{RIGHT} r_2$		
Imp.	Dip.	Capo
Black	Produzione	Mori
Whit e	Produzione	Mori

$r_1 \bowtie_{FULL} r_2$			
Imp.	Dip.	Capo	
Smith	Vendite		NULL
Black	Produzione	Mori	

Join multipli

White	Produzione	Mori	NULL	Acquisizione	Brown	White	Produzione	Mori
NULL	Acquisizione	Brown	White	Acquisizione	Brown	White	Acquisizione	Brown

Il JOIN è commutativo e associativo, quindi possiamo avere sequenze di JOIN senza rischio di ambiguità.

$r_1$		$r_2$		$r_3$	
Impiegato	Dipartimento	Dipartimento	Divisione	Divisione	Capo
Smith	Vendite	Produzione	A	A	Mori
Black	Produzione	Marketing	B	B	Brown
Brown	Marketing	Acquisizione	B		
White	Produzione				

$r_1 \bowtie r_2 \bowtie r_3$			
Impiegato	Dipartimento	Divisione	Capo
Black	Produzione	A	Mori
Brown	Marketing	B	Brown
White	Produzione	A	Mori

Prodotto cartesiano generato dal join

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

Semi-Giunzione

Relazione di attributi costituita da tutte le n-uple di Studenti che partecipano a Studenti  $\bowtie$  Esami.

Studenti		Esami	
Nome	Matricola	Indirizzo	Telefono
Mario Rossi	12345	Via Etnea 1	22222
Ugo Bianchi	12346	Via Roma 2	33333
Teo Verdi	12347	Via Torino 3	44444

Corso	Matricola	Voto
Architetture	12345	30
Programmazione	12346	18
Architetture	12346	27

Studenti ⋈ Esami			
Nome	Matricola	Indirizzo	Telefono
Mario Rossi	12345	Via Etnea 1	22222
Ugo Bianchi	12346	Via Roma 2	33333

Unione esterna

Estensione delle due tabelle con le colonne dell'altro con valori nulli e si fa l'unione

R				S				$R \cup^{\leftrightarrow} S$					
A	B	C	D	B	C	D	E	A		B	C	D	E
X	Y	Z	X	Y	Z	X	Y	X	Y	Z	X	X	NULL
X	Y	Z	X	Y	Z	X	M	X	Y	Z	X	X	NULL
X	Y	W	X	Y	W	X	Y	X	Y	W	X	X	NULL
X	Y	W	X	Y	W	X	M	X	Y	W	X	X	NULL
								NULL	Y	Z	X	Y	
								NULL	Y	Z	X	Y	
								NULL	Y	W	X	Y	
								NULL	Y	W	X	M	

Selezione con valori nulli

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

$SEL_{Età > 40}(Impiegati)$

La condizione atomica è vera solo per valori non nulli.

$SEL_{Età > 30}(Impiegati) \cup SEL_{Età \leq 30}(Impiegati)$

Darà come risultato:

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

Perché le condizioni atomiche vengono valutate separatamente

## Chiavi

**Chiave:** insieme di uno o più attributi di una relazione che identificano univocamente le tuple della relazione stessa.

Una chiave è una **superchiave minimale**, ovvero una superchiave che abbia le dimensioni minime possibili.

**Superchiave:** un sottoinsieme di attributi K di una relazione r che non contiene due tuple che hanno valori uguali su K, garantendo quindi l'univocità della tupla

**Chiave primaria:** chiave costituita da un attributo aggiunto appositamente per l'identificazione univoca delle tuple. Non ammette valori nulli per i suoi attributi.

**Chiave esterna:** uno o un insieme di attributi che riferisce una chiave primaria di un altro schema

Tabella 1

Matricola	Cognome	Nome	Nascita	Corso
4328	Rossi	Luigi	29/04/79	Informatica
6328	Rossi	Dario	29/04/79	Informatica
4766	Rossi	Luca	01/05/81	Fisica
4856	Neri	Luca	01/05/81	Economia
5536	Neri	Luca	05/03/78	Economia

- {Matricola} è una chiave primaria poiché ha la funzione di identificare univocamente le tuple;
- {Cognome, Nome, Nascita} è una chiave poiché l'insieme identifica univocamente una tupla;
- {Matricola, Corso} è una superchiave poiché non hanno valori uguali su K;
- {Nome, Corso} non è una superchiave poiché può contenere valori uguali.
- {Matricola} per la Tabella1 è chiave esterna che si riferisce a Matricola della tabella Studenti, per la quale è chiave primaria.

## Query Optimization

### 1. Raggruppamento di restrizioni

$$\sigma_{C(X)}(\sigma_{C(Y)}(E)) = \sigma_{C(X) \& C(Y)}(E)$$

### 2. Commutatività di $\sigma$ e $\pi$

$$\sigma_{C(X)}(\pi_Y(E)) = \pi_Y(\sigma_{C(X)}(E)), \text{ se } X \subseteq Y;$$

$$\pi_Y(\sigma_{C(X)}(\pi_{XY}(E))) = \pi_Y(\sigma_{C(X)}(E)) \text{ se } X \not\subseteq Y$$

### 3. Anticipazione di $\sigma$ rispetto a $\times$

$$\sigma_{C(X)}(E \times F) = \sigma_{C(X)}(E) \times F, \text{ se } X \subseteq \text{attr}(E)$$

$$\sigma_{C(X) \& C(Y)}(E \times F) = \sigma_{C(X)}(E) \times \sigma_{C(Y)}(F), \text{ se } X \subseteq \text{attr}(E), Y \subseteq \text{attr}(F)$$

$$\sigma_{C(X) \& C(Y) \& C(Z)}(E \times F) = \sigma_{C(Z)}(\sigma_{C(X)}(E) \times \sigma_{C(Y)}(F))$$

$$\text{se } X \subseteq \text{attr}(E), Y \subseteq \text{attr}(F), Z \cap \text{attr}(E) \neq \emptyset, Z \cap \text{attr}(F) \neq \emptyset$$

### 4. Raggruppamento di proiezioni

$$\pi_X(\pi_Y(E)) = \pi_X(E), \text{ se } X \subseteq Y$$

### 5. Eliminazione di proiezioni superflue

$$\pi_X(E) = E, \text{ se } X = \text{attr}(E)$$

### 6. Anticipazione della $\pi$ rispetto a $\times$

$$\pi_{XY}(E \times F) = \pi_X(E) \times \pi_Y(F), \text{ se } X \subseteq \text{attr}(E), Y \subseteq \text{attr}(F).$$

### Algoritmo per anticipare la selezione

Si anticipa  $\sigma$  rispetto a  $\pi$  usando la 3  $\pi_Y(\sigma_{C(X)}(E)) = \sigma_{C(X)}(\pi_Y(E))$ ;

Si raggruppano le restrizioni usando la 1  $\sigma_{C(X)}(\sigma_{C(Y)}(E)) = \sigma_{C(X) \& C(Y)}(E)$

Si anticipa l'esecuzione di  $\sigma$  su  $\times$  usando la 4

### Algoritmo per anticipare le proiezioni

Si eliminano le proiezioni superflue usando la 6  $\pi_X(E) = E, \text{ se } X = \text{attr}(E)$

Si raggruppano le proiezioni mediante la regola 5  $\pi_X(\pi_Y(E)) = \pi_X(E), \text{ se } X \subseteq Y$

Si anticipa l'esecuzione delle proiezioni rispetto al prodotto usando ripetutamente la 3  $\pi_Y(\pi_X(\pi_Y(E))) = \pi_Y(\pi_X(E))$  se  $X \not\subseteq Y$

$$[\pi_Y(\sigma_{C(X)}(\pi_{XY}(E)))] = \pi_Y(\sigma_{C(X)}(E)) \text{ se } X \not\subseteq Y$$

quando E è un prodotto, applicata da destra verso sinistra] e la 6 [Anticipazione della  $\pi$  rispetto a  $\times$ ].

### Distributività

- $\sigma_C(R_1 \cup R_2) = \sigma_C(R_1) \cup \sigma_C(R_2)$
- $\sigma_C(R_1 - R_2) = \sigma_C(R_1) - \sigma_C(R_2)$
- $\pi_X(R_1 \cup R_2) = \pi_X(R_1) \cup \pi_X(R_2)$

- NON VALE  $\pi_X(R_1 - R_2) = \pi_X(R_1) - \pi_X(R_2)$
- $\sigma_{C \vee D}(R) = \sigma_C(R) \cup \sigma_D(R)$
- $\sigma_{C \wedge D}(R) = \sigma_C(R) \cap \sigma_D(R)$
- $\sigma_{C \wedge D}(R) = \sigma_C(R) - \sigma_D(R)$

### **3. Algoritmi utili allo svolgimento delle Query**

## Algebra relazionale

Ricerca del valore massimo

R1:  $\pi_{id, valori}(Schema)$   
R2:  $\delta_{valori \rightarrow valoriTmp}(R1)$   
R3:  $R1 \bowtie_{valori < valoriTmp} (R2)$  //prendo i valori minori  
R4:  $R1 - R3$  //sottraggo i valori minori allo schema iniziale trovando il maggiore  
R5:  $\pi_{id}(R4)$

Ricerca del valore minimo

R1:  $\pi_{id, valori}(Schema)$   
R2:  $\delta_{valori \rightarrow valoriTmp}(R1)$   
R3:  $R1 \bowtie_{valori > valoriTmp} (R2)$   
R4:  $R1 - R3$   
R5:  $\pi_{id}(R4)$

Ricerca del valore massimo con tempo da calcolare

R1:  $\pi_{id, tempoInizio, tempoFine}(Schema)$   
R2:  $\delta_{tempoInizio, tempoFine \rightarrow tempoInizio1, tempoFine1}(R1)$   
R3:  $\pi_{id, tempoInizio, tempoFine}(R1 \bowtie_{(tempoFine - tempoInizio) < (tempoFine1 - tempoInizio1)} R2)$   
R4:  $R2 - R3$   
R5:  $\pi_{id}(R4)$

Ricerca di un valore ripetuto

R1:  $\pi_{id, valori}(Schema)$   
R2:  $\delta_{valori \rightarrow valoriTmp}(R1)$   
R3:  $R2 \bowtie R1$

Ricerca di un valore accaduto lo stesso giorno

R1:  $\delta_{Data \rightarrow Data1}(Schema)$   
R2:  $\delta_{Data \rightarrow Data2}(Schema)$   
R3:  $\sigma_{Data1 = Data2}(R1 \bowtie R2)$

Ricerca degli id in relazione con tutti i valori

R1:  $\pi_{valore}(Schema2)$   
R2:  $\pi_{id, valore}(Schema1)$   
R3:  $R2 / R1$  //prendo gli id in comune con i valori  
R4:  $\pi_{id}(R3 \bowtie Schema1)$

Ricerca di una coppia di id in relazione con tutti i valori

R1:  $\pi_{id, valore}(Schema)$   
R2:  $\delta_{id \rightarrow id1}(R1)$   
R3:  $R2 \bowtie R1$  //trova la coppia id in relazione con valore  
R4:  $\sigma_{id < id1}(R3)$  //eliminazione duplicati

