

# Basi di dati

L' Algebra Relazionale

# Introduzione all' Algebra Relazionale

---

Una volta definito lo schema logico di un database, partendo da un Diagramma E-R, e dopo aver inserito le tabelle nel database, eventualmente riempite con i dati opportuni

- ▶ ho un insieme di tabelle
- ▶ delle relazioni fra di esse

che nel complesso schematizzano una realtà

**Ricorro ai linguaggi di interrogazione e manipolazione per poter accedere, manipolare, ricavare informazioni da tali dati**

- ▶ in particolare utilizzerò i linguaggi di interrogazione
  - ▶ direttamente dal “terminale”
  - ▶ tramite istruzioni “embedded” (nascoste) negli applicativi

# Linguaggi interrogazione per basi di dati relazionali

---

Un linguaggio di manipolazione, o DML, permette di interrogare e modificare istanze di Basi di Dati

Tipologia:

- **Procedurali:**
  - si dice al sistema ciò che si vuole
- **Non Procedurali:**
  - si enunciano le proprietà del risultato

Rappresentanti più significativi:

- **Algebra relazionale:** linguaggio procedurale di tipo algebrico i cui operandi sono relazioni
- **SQL** (Structured Query Language): a carattere misto, procedurale e non



# L' Algebra Relazionale

---

L' **Algebra Relazionale** è utile perché, essendo svincolata dai “dettagli implementativi” dell'SQL, permette di comprendere rapidamente la tecnica d'uso dei linguaggi di interrogazione per basi di dati relazionali



# L'Algebra Relazionale

---

L'Algebra Relazionale è costituita da un **insieme di operazioni** che accettano una o due tabelle (relazioni) come **input** e producono una tabella (relazione) in **output**.

Costituita da un insieme di operatori:

- definiti su relazioni
- che producono relazioni
- e possono essere composti
  
- ▶ in sostanza, ciò che si fa con un linguaggio di interrogazione è:
  - ▶ prendere delle tabelle di input
  - ▶ generare delle tabelle di output secondo opportuni criteri

# Elementi di Algebra Relazionale

---

Le Operazioni di base comprendono:

operazioni di arità 1:

- ▶ **selezione** (notazione  $\sigma$ )
- ▶ **proiezione** (notazione  $\Pi$ )

operazioni di arità 2:

- ▶ **unione** (notazione  $\cup$ )
- ▶ **differenza insiemistica** (notazione  $-$ )
- ▶ **prodotto cartesiano** (notazione  $\times$ )

Operazioni derivate dalle precedenti:

- ▶ **intersezione** (notazione  $\cap$ )
- ▶ **Join** (notazione  $\triangleright \triangleleft$ )
- ▶ Gli operatori si possono comporre, dando luogo a espressioni algebriche di complessità arbitraria
- ▶ Gli operandi sono o (nomi di) relazioni del BD o espressioni (ben formate)

# Elementi di Algebra Relazionale

---

## Selezione ( $\sigma$ )

- ▶ la selezione è una operazione unaria, poiché accetta come argomento una sola tabella
- ▶ “seleziona” le tuple di una relazione che soddisfano un predicato
- ▶ il predicato è costituito dal nome di un attributo, da un operatore, da un valore costante
- ▶ più predicati possono essere uniti con or ( $\vee$ ) o and ( $\wedge$ )
- ▶ operatori sono
  - ▶ =, >, <, >=, <=, # (diverso)

$\sigma$  Predicato (relazione)

# Elementi di Algebra Relazionale

---

Data la Tabella con schema:

Impiegati (Matricola, Cognome, Filiale, Stipendio)

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Napoli	44
5698	Bianchi	Torino	64
2568	Rossi	Roma	55

Istanza della Tabella  
Impiegati

## Selezione ( $\sigma$ )

Vogliamo trovare gli impiegati che:

- guadagnano più di 62
- guadagnano più di 62 e lavorano a Milano



# Elementi di Algebra Relazionale

---

## Esempio operazione di Selezione:

- guadagnano più di 62

$\sigma$  Stipendio > 62 (Impiegati)

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64
5698	Bianchi	Torino	64

- guadagnano più di 62 e lavorano a Milano

$\sigma$  Stipendio > 62 AND Filiale = 'Milano' (Impiegati)

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

# Elementi di Algebra Relazionale

---

## Proiezione ( $\Pi$ )

- ▶ anche la proiezione è una operazione unaria
- ▶ data una tabella, la sua proiezione su un dato insieme di attributi è costituita dalla tabella generata dagli attributi specificati
- ▶ ignorando eventuali tuple (righe) duplicate

$\Pi$  Lista Attributi (**relazione**)

# Elementi di Algebra Relazionale

---

Data la Tabella con schema:

Impiegati (Matricola, Cognome, Filiale, Stipendio)

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Napoli	44
5698	Bianchi	Torino	64
2568	Rossi	Roma	55

Istanza della Tabella  
Impiegati

## Proiezione ( $\Pi$ )

per tutti gli impiegati:

- matricola e cognome
- cognome e filiale

# Elementi di Algebra Relazionale

---

## Esempio operazione di Proiezione:

- matricola e cognome di tutti gli impiegati

$\Pi$  matricola, cognome (Impiegati)

Matricola	Cognome
7309	Rossi
5998	Neri
9553	Milano
5698	Bianchi
2568	Rossi

- cognome e filiale di tutti gli impiegati

$\Pi$  cognome, filiale (Impiegati)

Cognome	Filiale
Rossi	Roma
Neri	Milano
Milano	Napoli
Bianchi	Torino
Rossi	Roma

# Elementi di Algebra Relazionale

---

## Combinazioni di operazioni di selezione e di proiezione

- dal momento che selezione e proiezione richiedono come argomenti una tabella (una relazione) e che producono come risultato una tabella, posso combinare le due operazioni per ottenere query complesse (e più espressive)
- Se voglio sapere matricola e cognome degli impiegati che guadagnano più di 62, sottopongo la query

$\Pi$  matricola, cognome  $(\sigma$  Stipendio>62 (Impiegati))

# Elementi di Algebra Relazionale

---

## Esempio: selezione e proiezione

$\Pi$  matricola, cognome ( $\sigma$  stipendio>62 (Impiegati))

Otengo il risultato:

Matricola	Cognome
5998	Neri
5698	Bianchi

e se scambiassi l'ordine di selezione e proiezione?

$\sigma$  stipendio>62 ( $\Pi$  matricola, cognome (Impiegati))

È un **errore**, poiché la tabella risultante dall'operazione di proiezione ha come schema gli attributi matricola e cognome, e su di esso non posso effettuare la selezione con predicato sull'attributo stipendio

# Elementi di Algebra Relazionale

---

## Prodotto cartesiano (X)

- ▶ è una operazione binaria, ossia accetta in input due tabelle
- ▶ a differenza delle operazioni di selezione e proiezione, che consentono di estrarre informazione da una sola tabella, permette di combinare tabelle e, dunque, di combinare informazione
- ▶ è estremamente costoso dal punto di vista computazionale e di risorse di memorizzazione

# Elementi di Algebra Relazionale

---

## Prodotto cartesiano (X)

- ▶ Il prodotto cartesiano fra due tabelle (relazioni) è una tabella con schema la somma (non l'unione matematica) degli schemi
  - ▶ attributi di nome uguale sono ripetuti
- ▶ il contenuto della tabella è dato dal prodotto cartesiano dei suoi elementi, ossia da tutte le coppie degli elementi appartenenti alle due relazioni

$$R1 \times R2$$



# Elementi di Algebra Relazionale

---

## Prodotto cartesiano:

Date le Tabelle con schemi:

Banchieri(NomeCliente, NomeBanchiere)

Clienti(NomeCliente, Via, Città)

### Istanza della Tabella Clienti

NomeCliente	Via	Città
Rossi	Pardi	Pisa
Bianchi	Roma	Pisa
Blu	Italia	Napoli
Verdi	Foscolo	Torino

### Istanza della Tabella Banchieri

NomeCliente	NomeBanchiere
Rossi	Cuccia
Bianchi	Visco

# Elementi di Algebra Relazionale

## Prodotto cartesiano:

Clienti(NomeCliente, Via, Città)

NomeCliente	Via	Città
Rossi	Pardi	Pisa
Bianchi	Roma	Pisa
Blu	Italia	Napoli
Verdi	Foscolo	Torino

Banchieri(NomeCliente, NomeBanchiere)

NomeCliente	NomeBanchiere
Rossi	Cuccia
Bianchi	Visco

$R_1 \times R_2$

Clienti x Banchieri

Clienti x Banchieri(C.NomeCliente, Via, Città, B.NomeCliente, NomeBanchiere)

C.NomeCliente	Via	Città	B.NomeCliente	NomeBanchiere
Rossi	Pardi	Pisa	Rossi	Cuccia
Rossi	Pardi	Pisa	Bianchi	Visco
Bianchi	Roma	Pisa	Rossi	Cuccia
Bianchi	Roma	Pisa	Bianchi	Visco
Blu	Italia	Napoli	Rossi	Cuccia
Blu	Italia	Napoli	Bianchi	Visco
Verdi	Foscolo	Torino	Rossi	Cuccia
Verdi	Foscolo	Torino	Bianchi	Visco

**OSS:** l'attributo NomeCliente è ripetuto due volte. Per questo l'operazione produce una "unione non matematica" degli schemi.

# Elementi di Algebra Relazionale

---

## Osservazioni:

- ▶ se  $N1$  è il numero di tuple di Clienti ed  $N2$  è il numero di tuple di Banchieri, il risultato del Prodotto Cartesiano ha numero di tuple  $N1 \times N2$ 
  - ▶ nel caso precedente  $4 \times 2 = 8$
- ▶ **importante:** combinare semplicemente l'informazione con il prodotto cartesiano non serve a niente, se non ad ottenere “tabelloni”
- ▶ si possono generare delle informazioni prive di significato, ed anche perdere contenuto informativo (ossia incapacità di estrarre dal prodotto cartesiano la stessa informazione delle tabelle di partenza)

# Elementi di Algebra Relazionale

---

## Esempi di utilizzo del prodotto cartesiano

- supponiamo, dati gli schemi clienti e banchieri precedenti, di voler saper la via e la città in cui abitano tutti i clienti di Cuccia
- per ottenere tale informazione non basta solo la tabella banchieri, perché associa ad un cliente il relativo banchiere, né la tabella clienti da sola, perché non ho il banchiere
- devo allora combinare le due tabelle tramite il prodotto cartesiano, perché nell'insieme delle tabelle è contenuta l'informazione che mi serve

***Clienti x Banchieri***

# Elementi di Algebra Relazionale

---

- A questo punto, nella tabella risultante, ho tutta l'informazione che mi serve (o meglio, di più)
- devo estrarre solo le tuple significative
- prendo le tuple che hanno NomeBanchiere=Cuccia ed infine proietto

$$\sigma_{C.NomeCliente=B.NomeCliente} (Clienti \times Banchieri)$$

$$\Pi_{via, città} (\sigma_{NomeBanchiere=Cuccia} (\sigma_{C.NomeCliente=B.NomeCliente} (Clienti \times Banchieri)))$$

# Elementi di Algebra Relazionale

---

## Ricapitolando: (funziona così anche in SQL)

Una volta formulata la query:

- cerco le tabelle (relazioni) in cui è contenuta l'informazione che mi serve
- effettuo l'operazione di prodotto cartesiano se l'informazione è presente in più tabelle che vanno concatenate
- scarto le tuple non significative con una selezione sugli attributi a comune
- inizio a fare selezioni e proiezioni per rispondere alla query

# Elementi di Algebra Relazionale

---

**Il join naturale** (simbolo  $\bowtie$  )

- l'operazione di prodotto cartesiano accompagnata dalla selezione sugli attributi a comune fra le due relazioni prende il nome di join naturale
- l'operatore di join naturale, combina quindi le tuple di due relazioni sulla base dell'uguaglianza dei valori degli attributi comuni alle due relazioni

così, la query:

$\sigma_{C.NomeCliente=B.NomeCliente} (Clienti \times Banchieri)$

si rappresenta come

*clienti*  $\bowtie$  *banchieri*

# Elementi di Algebra Relazionale

---

## Join naturale:

Voli(Codice, Data, Comandante)

Codice	Data	Comandante
AZ427	21/07/2001	Bianchi
AZ427	23/07/2001	Rossi
TW056	21/07/2001	Smith

Linee(Codice, Partenza, Arrivo)

Codice	Partenza	Arrivo
AZ427	FCO	JFK
TW056	LAX	FCO

Voli  $\bowtie$  Linee

Codice	Data	Comandante	Partenza	Arrivo
AZ427	21/07/2001	Bianchi	FCO	JFK
AZ427	23/07/2001	Rossi	FCO	JFK
TW056	21/07/2001	Smith	LAX	FCO

$\sigma_{V.Codice=L.Codice}$  (Voli X Linee)



# Elementi di Algebra Relazionale

---

## Osservazioni

- ▶ il join (natural) non è sempre supportato (DB2 per esempio)
- ▶ tipicamente, tutte le operazioni che richiedono un prodotto cartesiano richiedono in realtà un join
  - ▶ se le tabelle non hanno attributi a comune, non sono in relazione e dunque non serve a niente fare il prodotto cartesiano
- ▶ è bene allora “avere in mente” il join, anche se non è supportato dal linguaggio
  - ▶ dovremo vedere come fare il join in SQL, in termini di prodotto cartesiano e selezione

# Elementi di Algebra Relazionale

---

## Unione, Intersezione e Differenza

- Le Relazioni sono insiemi, quindi possiamo applicare gli operatori tra insiemi
- Tuttavia, vogliamo che il risultato siano relazioni (cioè insiemi omogenei di ennuple)
  - È quindi significativo applicare l'unione, intersezione e differenza solo a coppie di relazioni definite su gli stessi attributi

# Elementi di Algebra Relazionale

---

## Unione

- l'unione fra due tabelle è rappresentata da una tabella costituita dall'unione “matematica” delle due tabelle
  - ▶ L' unione fra due tabelle A e B è una tabella che contiene le tuple che sono presenti in A, in B o in entrambe
  - ▶ tuple replicate sono considerate un'unica tupla
- affinché l'unione abbia senso, è necessario che
  - ▶ le due tabelle abbiano lo **stesso numero di attributi**
  - ▶ i tipi degli attributi corrispondenti siano uguali
- Se le relazioni hanno **nomi di attributo diversi**, nella relazione risultato per convenzione si usano i nomi della prima relazione (in questo caso A), a meno di opportune ridenominazioni
- Il **grado** della relazione risultato è uguale al grado delle relazioni operandi

# Elementi di Algebra Relazionale

---

## Esempio dell' operazione di Unione:

si hanno le seguenti tabelle:

- ▶ depositi(NomeFiliale, NumeroDeposito, NomeCliente, Saldo)
  - ▶ prestiti(NomeFiliale, NumeroPrestito, NomeCliente, Saldo)
- ▶ si vogliono trovare tutti i clienti della filiale di Roma che hanno un deposito o un prestito o entrambi

$$\Pi \text{ NomeCliente}(\sigma \text{ NomeFiliale}=\text{Roma}(\text{Depositi}))$$
$$\cup$$
$$\Pi \text{ NomeCliente}(\sigma \text{ NomeFiliale}=\text{Roma}(\text{Prestiti}))$$

# Elementi di Algebra Relazionale

Depositi

Filiale	NumeroDeposito	Ciente	Saldo
Milano	010025	Rossi	1000
Roma	012025	Bianchi	15000
Roma	010525	Neri	200
Pisa	013025	Torino	100

Prestiti

Filiale	NumeroPrestito	Ciente	Saldo
Roma	010625	Rossi	1500
Milano	017025	Bianchi	3000

$\Pi$  Cliente ( $\sigma_{\text{Filiale}=\text{Roma}}$ (Depositi))

Cliente
Bianchi
Neri

$\Pi$  Cliente ( $\sigma_{\text{Filiale}=\text{Roma}}$ (Prestiti))

Cliente
Rossi

$\Pi$  Cliente ( $\sigma_{\text{Filiale}=\text{Roma}}$ (Depositi))  $\cup$   $\Pi$  Cliente ( $\sigma_{\text{Filiale}=\text{Roma}}$ (Prestiti))

Cliente
Rossi
Bianchi
Neri

# Elementi di Algebra Relazionale

---

## Differenza - :

- La differenza fra due tabelle A e B è una tabella che contiene le tuple che sono presenti in A ma non in B
- La differenza (come l'unione) può essere eseguita solo se le relazioni hanno lo stesso grado e gli attributi sono compatibili
- Il grado della relazione risultato è uguale al grado delle relazioni operandi

## Esempio dell' operazione di Differenza:

Trovare i clienti che hanno un deposito ma non un prestito nella filiale di Roma

$$\Pi \text{ Cliente } (\sigma_{\text{Filiale}=\text{Roma}}(\text{Depositi})) - \Pi \text{ Cliente } (\sigma_{\text{Filiale}=\text{Roma}}(\text{Prestiti}))$$

# Elementi di Algebra Relazionale

Depositi

Filiale	NumeroDeposito	Cliente	Saldo
Milano	010025	Rossi	1000
Roma	012025	Bianchi	15000
Roma	010525	Neri	200
Pisa	013025	Torino	100

Prestiti

Filiale	NumeroPrestito	Cliente	Saldo
Roma	010625	Rossi	1500
Milano	017025	Bianchi	3000
Roma	018221	Neri	10000

$\Pi$  Cliente ( $\sigma$  Filiale=Roma(Depositi))

Cliente
Bianchi
Neri

$\Pi$  Cliente ( $\sigma$  Filiale=Roma(Prestiti))

Cliente
Rossi
Neri

$\Pi$  Cliente ( $\sigma$  Filiale=Roma(Depositi)) -  $\Pi$  Cliente ( $\sigma$  Filiale=Roma(Prestiti))

Cliente
Bianchi

# Elementi di Algebra Relazionale

---

## Intersezione $\cap$ :

- L' intersezione fra due tabelle  $A \cap B$  è una tabella che contiene le tuple che sono presenti in A e in B

## Esempio dell' operazione di Intersezione :

- ▶ trovare i clienti che hanno sia un conto che un prestito

$\Pi$  Cliente ( $\sigma_{\text{Filiale}=\text{Roma}}(\text{Depositi})$ )  $\cap$   $\Pi$  Cliente ( $\sigma_{\text{Filiale}=\text{Roma}}(\text{Prestiti})$ )



# Elementi di Algebra Relazionale

Depositi

Filiale	NumeroDeposito	Cliente	Saldo
Milano	010025	Rossi	1000
Roma	012025	Bianchi	15000
Roma	010525	Neri	200
Pisa	013025	Torino	100

Prestiti

Filiale	NumeroPrestito	Cliente	Saldo
Roma	010625	Neri	1500
Milano	017025	Bianchi	3000

$\Pi$  Cliente ( $\sigma$  Filiale=Roma(Depositi))

Cliente
Bianchi
Neri

$\Pi$  Cliente ( $\sigma$  Filiale=Roma(Prestiti))

Cliente
Neri

$\Pi$  Cliente ( $\sigma$  Filiale=Roma(Depositi))  $\cap$   $\Pi$  Cliente ( $\sigma$  Filiale=Roma(Prestiti))

Cliente
Neri

# Elementi di Algebra Relazionale

---

## Ridenominazione:

- operatore unario
- “modifica i nomi di attributi” senza cambiare i valori
- elimina le limitazioni associate con gli operatori su insiemi

## Notazione:

$$\rho_{(Y \leftarrow X)}(R)$$

## Esempio dell' operazione di Ridenominazione:

Persone

CF	Cognome	Nome
BNCGRG78F2IA	Bianchi	Mario
RSSGRG78F23X	Rossi	Paolo

$$\rho_{(\text{CodiceFiscale} \leftarrow \text{CF})}(\text{Persone})$$

CodiceFiscale	Cognome	Nome
BNCGRG78F2IA	Bianchi	Mario
RSSGRG78F23X	Rossi	Paolo

# Conclusioni

---

L' **algebra relazionale** è un linguaggio per BD costituito da un insieme di operatori che si applicano a una o più relazioni e che producono una relazione

- Gli **operatori di base** sono: selezione, proiezione, ridenominazione, unione, prodotto cartesiano e differenza
  - Sulla base di questi si possono poi definire altre operazioni (derivate) quali divisione, intersezione, join, ...

## In generale:

- un'interrogazione sulla BD può essere rappresentata in Algebra Relazionale mediante diverse espressioni, tutte tra loro equivalenti dal punto di vista del risultato, ma non necessariamente dal punto di vista dell'efficienza.