Basi di dati MOD 1
- Algebra relazionale -

L'algebra relazionale è un insieme di operatori su relazioni che danno in output altre relazioni.
È praticamente SQL scritto in un altro modo…
È composta da:
- operatori primitivi (ridenominazione, proiezione, unione e differenza, restrizione, prodotto)
- operatori derivati (giunzioni, divisione, ...)
- altri operatori (raggruppamento, order by, min, max)

Non si usa il valore NULL!
Come notazione useremo una relazione R (A1: T1, ..., An: Tn)
- Tipo: {(A1: T1, ..., An: Tn)}
- Grado: n
- Data una ennupla t ∈ R:
  t.Ai valore dell'attributo Ai

Ridenominazione:
Viene usata appunto per ridenominare (rinominare) un campo, si scrive come:

$$\rho_{A \leftarrow B}(R)$$

Ad esempio rinomino gli attributi Utenti in Persone.

Unione e Differenza:
Vengono usate per unire o sottrarre due relazioni (tabelle), date R e S due relazioni dello stesso tipo:
- Unione viene rappresentata con

$$R \cup S = \{t \mid t \in R \ \vee \ t \in S\}$$

- Differenza viene rappresentata con

$$R - S = \{t \mid t \in R \ \wedge \ t \notin S\}$$

Proiezione:

La proiezione seleziona gli attributi specificati di una certa relazione.

Data R(X) con {A1, ..., Am} ⊆ X, la proiezione si scrive:

$$\pi_{A_1,A_2,\ldots,A_m}(R)$$

Quindi esclude gli attributi diversi da A1,...,Am, in altri termini, seleziona gli attributi specificati!

La definizione quindi sarebbe:

$$\pi_{A_1,\ldots,A_m}(R) = \{\langle t.A_1, \ldots, t.A_m \rangle \mid t \in R\}$$

Es:

Data la tabella Studenti

Studenti

Nome	Cognome	Matricola	Anno	Prov
Paolo	Verdi	71523	2005	VE
Anna	Rossi	76366	2006	PD
Giorgio	Zeri	71347	2005	VE
Chiara	Scuri	71346	2006	VE

Trovare il nome, la matricola e la provincia degli studenti.

$$\pi_{\text{Nome, Matricola, Provincia}}(\text{Studenti})$$

Che sotto forma di tabella sarà:

Nome	Matricola	Provinci
Paolo	71523	VE
Anna	76366	PD
Giorgio	71347	VE
Chiara	71346	VE

Restrizione:
La restrizione seleziona le ennuple che soddisfano la condizione Φ (phi):

$$\sigma_{\phi}(R) = \{t \mid t \in R \ \wedge \ \phi(t)\}$$

La condizione Φ è una combinazione di (dis)uguaglianze e disequazioni tra attributi (o tra attributi e costanti):

$$\phi \ ::= \ A_i \, op \, A_j \ \mid \ A_i \, op \, c \ \mid \ \neg\phi \ \mid \ \phi \wedge \phi \ \mid \ \phi \vee \phi$$

Dove op è un operatore (==, > , != , ecc) e il controllo della condizione viene fatta per singola ennupla!
Es:
- Trovare i dati degli studenti della provincia di Venezia:

$\sigma_{\text{Provincia = 'VE'}}$ (Studenti)

Nome	Cognome	Matricola	Anno	Prov
Paolo	Verdi	71523	2005	VE
Giorgio	Zeri	71347	2005	VE
Chiara	Scuri	71346	2006	VE

- Trovare il nome, la matricola e l'anno di iscrizione degli studenti di Venezia:

$\pi_{\text{Nome,Matricola,Anno}}$ ($\sigma_{\text{Provincia = 'VE'}}$ (Studenti))

Nome	Matricola	Anno
Paolo	71523	2005
Giorgio	71347	2005
Chiara	71346	2006

Quindi la restrizione è banalmente la clausola WHERE di una query SQL!

Prodotto:
Date due relazione R ed S, il prodotto prende ogni ennupla di R e a ciascuna concatena ogni ennupla di S!

$$R \times S$$

$$R \times S = \{\langle t.A_1, \ldots, t.A_n, u.B_1, \ldots, u.B_m \rangle \mid t \in R \ \wedge \ u \in S\}$$

Es:

A	B
a1	b1
a2	b2

$\times$

C	D
c1	d1
c2	d2
c3	d3

$=$

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a1	b1	c3	d3
a2	b2	c1	d1
a2	b2	c2	d2
a2	b2	c3	d3

È molto utile per incrociare i dati tra varie tabelle!

Es:

- Trovare il nome degli studenti che hanno superato l'esame di BD con 30:

  $$\pi_{\text{Nome}}(\sigma_{\text{Materia}='BD'\wedge\text{Voto}=30}(\sigma_{\text{Matricola}=\text{Candidato}}(\text{Studenti}\times\text{Esami})))$$

Però, fare il prodotto così su tutte le tuple è uno spreco di risorse e di tempo, la facciamo in maniera un po' più "smart" introducendo il concetto di giunzione (CPU time), che è il prodotto tra relazioni basandosi su certi attributi, che di solito gli attributi sono le PK e le FK!

Es:

- Stessa interrogazione di prima:

  $$\pi_{\text{Nome}}(\sigma_{\text{Materia}\,=\,'BD'\wedge\text{Voto}\,=\,30}(\text{Studenti}\ \underset{\text{Matricola}=\text{Candidato}}{\bowtie}\ \text{Esami}))$$

  Vediamo che abbiamo fatto la giunzione sugli attributi PK ed FK delle due tabelle che le mettono in relazione!

Giunzione:

La giunzione è utile per combinare informazioni di relazioni correlate

$$R \underset{A_i=B_j}{\bowtie} S$$

La giunzione è un prodotto con una condizione

$$R \underset{A_i=B_j}{\bowtie} S \;=\; \sigma_{A_i=B_j}(R\times S)$$

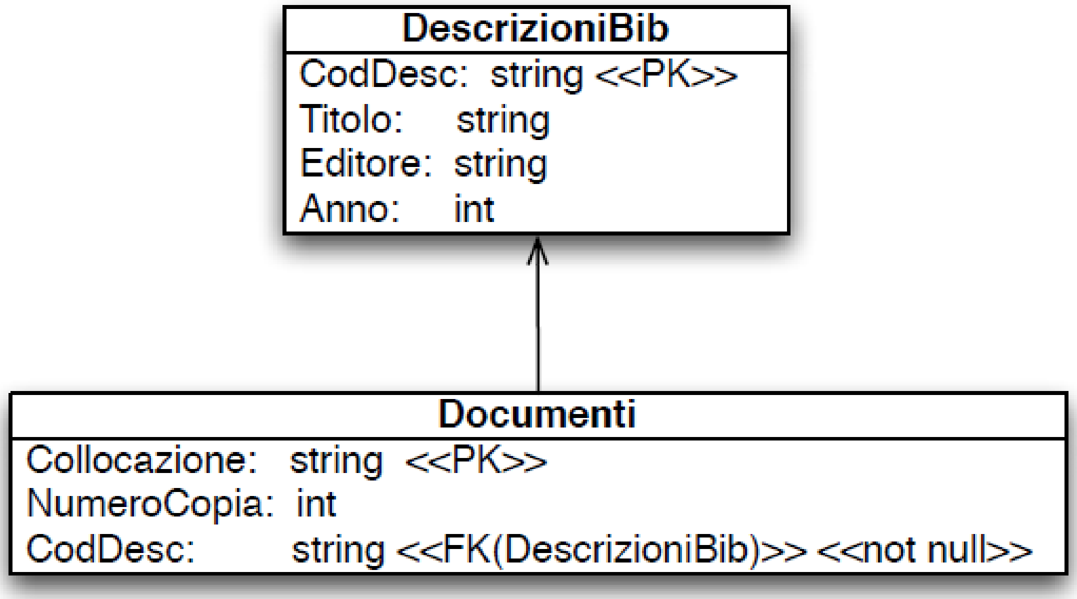
Ci sono vari tipi di giunzione:

- Giunzione naturale:

$$R \bowtie S$$

Combina le due relazioni basandosi sulla FK di R che referenzia la PK di S (o viceversa).

$$\mathsf{Documenti} \bowtie \mathsf{DescrizioniBib}$$



- Giunzione esterna:
  Non c'è scritto nulla nelle slide…

- Intersezione:

$$R \cap S$$

Praticamente gli attributi che hanno in comune…

- Divisione:
  Date due relazioni R(XY) e S(Y) la divisione produce una relazione T(X) tale che una ennupla t è in T se e solo se per ogni s in S la ennupla <t, s> appare in R.

$$R \div S$$

Non ho capito la differenza tra intersezione e divisione ma ok…

Es:
Matricola degli studenti che hanno fatto tutti gli esami che ha fatto Anna Rossi (matr. 76366).

Esami di Anna Rossi:

$$ES\_AR = \pi_{\text{Materia}}(\sigma_{\text{Candidato}='76366'}(\text{Esami}))$$

Esami studenti con matricola:

$$ES = \pi_{\text{Candidato,Materia}}(\text{Esami})$$

Il risultato finale sarà:

$$ES \div ES\_AR$$

La divisione è usata per query che coinvolgono quantificazione universale.

- Proiezione generalizzata:
  Viene usata per ridenominare espressioni.

$$\pi_{Exp_1 \ \mathbf{AS} \ A_1, Exp_2 \ \mathbf{AS} \ A_2, \ldots, Exp_n \ \mathbf{AS} \ A_n}(R)$$

Le espressioni Exp_i possono comprendere attributi, costanti, e operazioni su di essi.
Es:
Data una relazione Utente(Codice,SalarioLordo,Trattenute, ...) dare il salario netto di ogni utente

$$\pi_{\text{Codice, SalarioLordo}-\text{Trattenute} \ AS \ \text{Stipendio}}(\text{Utente})$$

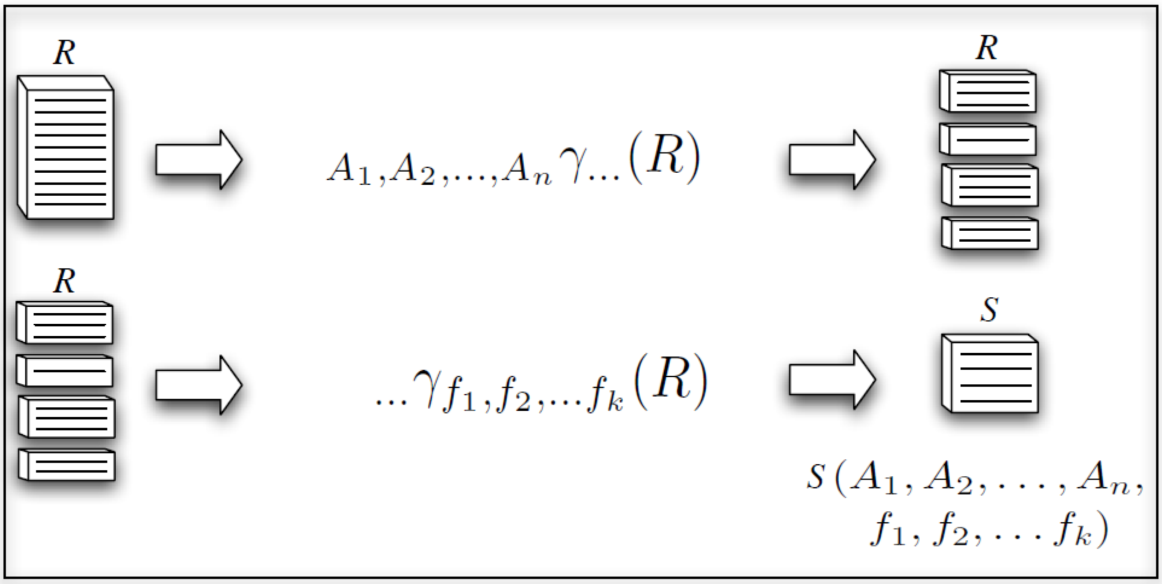
Funzioni di aggregazione:
Hanno come input multinsiemi e hanno come output un valore:

- sum: ritorna la somma degli elementi
- avg: ritorna il valore medio degli elementi
- count: ritorna il numero degli elementi
- min/max: ritorna il valore minimo/massimo dell'elemento
- distinct: ignora i duplicati, va aggiunto in coda alle altre funzioni di aggregazione!
- raggruppamento: va usato solo sulle funzioni di aggregazione!

$$_{A_1, A_2, \ldots, A_n}\gamma_{f_1, f_2, \ldots f_k}(R)$$

dove gli Ai sono attributi di R e le fi sono espressioni che usano funzioni di aggregazione (min, max, count, sum, avg, ...)

Diamo una rappresentazione di cosa accade dopo il raggruppamento:



Es:
Trovare per ogni candidato il numero degli esami, il voto minimo, massimo e medio
Tabella (relazione):

Materia	Candidato	Data	Voto	Lode
BD	71523	08.07.06	20	N
FIS	76366	08.07.07	26	N
ASD	71523	28.12.06	30	S
BD	76366	28.12.06	28	N

Query:

$$_{\text{Candidato}}\gamma_{\text{count}(*),\ \text{min}(\text{Voto}),\ \text{max}(\text{Voto}),\ \text{avg}(\text{Voto})}(\text{Esami})$$

Non ci andrebbe il pigreco davanti al candidato?

Dopo il raggruppamento:

Materia	Candidato	Data	Voto	Lode
BD	71523	08.07.06	20	N
ASD	71523	28.12.06	30	S
FIS	76366	08.07.07	26	N
BD	76366	28.12.06	28	N

E dopo il calcolo delle funzioni:

Candidato	Count(*)	min(Voto)	max(Voto)	avg(Voto)
71523	2	20	30	25
76366	2	26	28	27

Le trasformazioni algebriche e gli alberi logici non li hanno mai chiesti, quindi non li tratto…