ALGEBRA RELAZIONALE

Linguaggi di interrogazione (query language) per basi di dati relazionali

algebra relazionale	procedurale	
calcolo relazionale	dichiarativo	teorico
SQL	parzialmente dichiarativo	reale
QBE	dichiarativo	reale

Algebra relazionale

Insieme di operatori

- su relazioni
- che producono relazioni
- e quindi possono essere la base per espressioni complesse

Operatori dell'algebra relazionale:

- unione, intersezione, differenza
- ridenominazione
- selezione
- proiezione
- join (join naturale, prodotto cartesiano, theta-join)

Operatori insiemistici

- le relazioni sono insiemi
- i risultati debbono essere relazioni
  (insiemi di ennuple omogenee)

Quindi

- è possibile applicare gli operatori insiemistici
  (unione, intersezione, differenza)
  solo a coppie di relazioni definite sugli stessi attributi.

laureati

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

quadri

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

laureati $\cup$ quadri

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

laureati

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

quadri

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

laureati $\cap$ quadri

Matricola	Cognome	Età
7432	Neri	39
9824	Verdi	38

laureati

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

quadri

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

laureati − quadri

Matricola	Cognome	Età
7274	Rossi	37

paternità

Padre	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Abramo	Ismaele

maternità

Madre	Figlio
Eva	Caino
Eva	Set
Sara	Isacco
Agar	Ismaele

paternità ∪ maternità ??

Ridenominazione

- operatore monadico ("unario")
- intuitivamente, "modifica lo schema" lasciando inalterata l'istanza dell'operando
- permette di superare le limitazioni imposte agli operatori insiemistici

paternità

Padre	*Figlio*
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Isacco	Giacobbe

$\rho_{\mathsf{Genitore} \leftarrow \mathsf{Padre}}(\mathit{paternità})$

Genitore	*Figlio*
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Isacco	Giacobbe

- $r(X)$, relazione
- Y, insieme di attributi con $|X| = |Y|$
- $f : X \longrightarrow Y$, iniettiva (e suriettiva)
  (funzione di ridenominazione)

la **ridenominazione** $\rho_f(r)$ di r rispetto a f contiene le ennuple t definite su Y tali che
esista una ennupla $t' \in r$
con $t'[A] = t[f(A)]$, per ogni $A \in X$

La funzione di ridenominazione viene scritta indicando solo gli attributi su cui è diversa dall'identità, con la notazione (in cui l'ordinamento è significativo):

$$f(A_1), f(A_2), \ldots, f(A_k) \leftarrow A_1, A_2, \ldots, A_k$$

Esempio:

$X =$ *Padre,Figlio*
$Y =$ *Genitore,Figlio*
$f($*Padre*$) =$ *Genitore*
$f($*Figlio*$) =$ *Figlio*

f si indica con

Genitore $\leftarrow$ *Padre*

paternità

Padre	*Figlio*
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Abramo	Ismaele

maternità

Madre	*Figlio*
Eva	Caino
Eva	Set
Sara	Isacco
Agar	Ismaele

$\rho_{Genitore \leftarrow Padre}(paternità) \cup \rho_{Genitore \leftarrow Madre}(maternità)$

Genitore	*Figlio*
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Abramo	Ismaele
Eva	Caino
Eva	Set
Sara	Isacco
Agar	Ismaele

impiegati

Cognome	Agenzia	Stipendio
Rossi	Roma	45
Neri	Milano	53

operai

Cognome	Fabbrica	Salario
Verdi	Latina	33
Bruni	Monza	32

$\rho_{\text{Sede,Retribuzione} \leftarrow \text{Agenzia,Stipendio}}(\text{impiegati}) \cup$
$\rho_{\text{Sede,Retribuzione} \leftarrow \text{Fabbrica,Salario}}(\text{operai})$

Cognome	Sede	Retribuzione
Rossi	Roma	45
Neri	Milano	53
Verdi	Latina	33
Bruni	Monza	32

Selezione

- operatore monadico
- produce un risultato che
  - ha lo stesso schema dell'operando
  - contiene un sottoinsieme delle ennuple dell'operando
- produce decomposizioni "orizzontali"

impiegati

Cognome	*Nome*	*Età*	*Stipendio*
Rossi	Mario	25	2.000.000
Neri	Luca	40	3.000.000
Verdi	Nico	36	4.500.000
Rossi	Marco	40	3.900.000

$\sigma_{\textit{Età} < 30 \vee \textit{Stipendio} > 4.000.000}(\textit{impiegati})$

Cognome	*Nome*	*Età*	*Stipendio*
Rossi	Mario	25	2.000.000
Verdi	Nico	36	4.500.000

il risultato di una selezione contiene le ennuple dell'operando che soddisfano la condizione di selezione

- $r(X)$, relazione
- F formula proposizionale su X:
  - cioè formula ottenuta combinando con i connettivi logici $\vee$ (OR), $\wedge$ (AND), $\neg$ (NOT),
  - atomi del tipo $A\theta B$ o $A\theta c$
  - con A e B attributi in X
  - c costante “compatibile” con $dom(A)$
  - e θ operatore di confronto ($=$, $\neq$, $>$, $<$, $\geq$, $\leq$).

  È definito un valore di verità per F su ciascuna ennupla t:
  - $A\theta B$ è vera su t se $t[A]$ è in relazione θ con $t[B]$
  - $A\theta c$ è vera su t se $t[A]$ è in relazione θ con c
  - $F_1 \vee F_2$, $F_1 \wedge F_2$ e $\neg F_1$ hanno l’usuale significato

la **selezione** $\sigma_F(r)$ di r rispetto a F contiene le ennuple t di r su cui F è vera

Proiezione

- operatore monadico
- produce un risultato
  - definito su un sottoschema dell’operando
  - a cui contribuiscono tutte le ennuple dell’operando
- produce decomposizioni “verticali”

- $r(X)$, relazione
- Y sottoinsieme di X

la **proiezione** $\pi_Y(r)$ di r su Y è
un insieme di ennuple su Y:

$$\{\, t[Y] \mid t \in r \,\}$$

impiegati

Cognome	*Nome*	*Reparto*	*Capo*
Rossi	Mario	Vendite	De Rossi
Neri	Luca	Vendite	De Rossi
Verdi	Nico	Personale	Lupi
Rossi	Marco	Personale	Lupi

$\pi_{\textit{Cognome Nome}}(\textit{impiegati})$

Cognome	*Nome*
Rossi	Mario
Neri	Luca
Verdi	Nico
Rossi	Marco

$\pi_{\textit{Reparto Capo}}(\textit{impiegati})$

Reparto	*Capo*
Vendite	De Rossi
Personale	Lupi

- il risultato di una proiezione contiene al più tante ennuple quante l'operando
- può contenerne di meno
- $\pi_Y(r)$ contiene lo stesso numero di ennuple di r se e solo se Y è superchiave per r
  (N.B.: questa proprietà è definita a livello di istanza, non a livello di schema)

Join (naturale)

- operatore binario (generalizzabile)
- correla dati di relazioni diverse
- è l'operatore più caratteristico dell'algebra relazionale

r_1

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Reparto	Capo
produzione	Mori
vendite	Bruni

$r_1 \bowtie r_2$

Impiegato	Reparto	Capo
Rossi	vendite	Bruni
Neri	produzione	Mori
Bianchi	produzione	Mori

il **join naturale** $r_1 \bowtie r_2$ di $r_1(X_1)$ e $r_2(X_2)$ è una relazione definita su X_1X_2:

$$\{\, t \text{ su } X_1X_2 \;\mid\; \begin{array}{l}\text{esistono } t_1 \in r_1 \text{ e } t_2 \in r_2 \\ \text{con } t[X_1] = t_1 \text{ e } t[X_2] = t_2\}\end{array}$$

r_1

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Reparto	Capo
produzione	Mori
vendite	Bruni

$r_1 \bowtie r_2$

Impiegato	Reparto	Capo
Rossi	vendite	Bruni
Neri	produzione	Mori
Bianchi	produzione	Mori

un join completo
(ogni ennupla contribuisce al risultato)

r_1

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Reparto	Capo
produzione	Mori
acquisti	Bruni

$r_1 \bowtie r_2$

Impiegato	Reparto	Capo
Neri	produzione	Mori
Bianchi	produzione	Mori

un join con ennuple dangling
(non combinabili con ennuple dell'altra relaione)

r_1

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Reparto	Capo
concorsi	Mori
acquisti	Bruni

$r_1 \bowtie r_2$

Impiegato	Reparto	Capo

un join vuoto

r_1

Impiegato	Progetto
Rossi	A
Neri	A
Bianchi	A

r_2

Progetto	Capo
A	Mori
A	Bruni

$r_1 \bowtie r_2$

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

un altro join completo
(con $|r_1| \times |r_2|$ ennuple: ogni ennupla di r_1 è combinabile con tutte le ennuple di r_2 e viceversa)

- il join di r_1 e r_2 contiene un numero di ennuple compreso fra 0 e $|r_1| \times |r_2|$;
- se il join di r_1 e r_2 è completo, allora contiene almeno un numero di ennuple pari al massimo fra $|r_1|$ e $|r_2|$;
- se $X_1 \cap X_2$ contiene una chiave per r_1 (per r_2), allora il join di $r_1(X_1)$ e $r_2(X_2)$ contiene al più $|r_2|$ ($|r_1|$) ennuple;

paternità

Padre	*Figlio*
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Abramo	Ismaele

maternità

Madre	*Figlio*
Eva	Caino
Eva	Set
Sara	Isacco
Agar	Ismaele

paternità$\bowtie$*maternità*

Padre	*Figlio*	Madre
Adamo	Caino	Eva
Abramo	Isacco	Sara
Abramo	Ismaele	Agar

- il join di due relazioni sugli stessi attributi è pari all'intersezione delle due relazioni:

$$r_1(X) \bowtie r_2(X) = r_1(X) \cap r_2(X)$$

- il join naturale è definito anche fra relazioni che non hanno attributi in comune.
- In tal caso viene chiamato **prodotto cartesiano**.
- Il prodotto cartesiano di r_1 e r_2 contiene sempre $|r_1| \times |r_2|$ ennuple.
- Non si tratta del tradizionale prodotto cartesiano fra insiemi

impiegati

Impiegato	Progetto
Rossi	A
Neri	A
Neri	B

progetti

Codice	Nome
A	Venere
B	Marte

impiegati$\bowtie$ progetti

Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	A	Venere
Neri	B	A	Venere
Rossi	A	B	Marte
Neri	A	B	Marte
Neri	B	B	Marte

- il join naturale è commutativo:
  $r_1 \bowtie r_2 = r_2 \bowtie r_1$, per ogni r_1 e r_2
- il join naturale è associativo:
  $r_1 \bowtie (r_2 \bowtie r_3) = (r_1 \bowtie r_2) \bowtie r_3$, per ogni r_1, r_2, r_3

È quindi possibile definire un operatore di join n-ario (attraverso applicazioni ripetute dell'operatore binario).
Una definizione equivalente:

$r_1(X_1) \bowtie r_2(X_2) \bowtie \ldots \bowtie r_n(X_n)$ (in breve $\bowtie_{i=1}^{n} r_i$):

$$\{\, t \text{ su } X_1 \ldots X_n \mid \text{esistono } t_1 \in r_1, \ldots, t_n \in r_n, \text{ con } t[X_1] = t_1, \ldots, t[X_n] = t_n \}$$

Dimostrazione: per induzione.

Join e proiezione: proprietà

Join e proiezione sono (quasi) inversi.

- $r_1(X_1)$ e $r_2(X_2)$
  - $\pi_{X_1}(r_1 \bowtie r_2) \subseteq r_1$
  - $\pi_{X_1}(r_1 \bowtie r_2) = r_1$ e $\pi_{X_2}(r_1 \bowtie r_2) = r_2$ se e solo se il join di r_1 e r_2 è completo
  - il join di $\pi_{X_1}(r_1 \bowtie r_2)$ e $\pi_{X_2}(r_1 \bowtie r_2)$ è completo
- $r(X)$; $X = X_1 X_2$
  - $\pi_{X_1}(r) \bowtie \pi_{X_2}(r) \supseteq r$
  - se $\pi_{X_1}(r) \bowtie \pi_{X_2}(r) = r$ diciamo che r **si decompone senza perdita su** X_1 e X_2
  - $\pi_{X_1}(r) \bowtie \pi_{X_2}(r)$ si decompone senza perdita su X_1 e X_2

Queste proprietà sono generalizzabili al join n-ario.

Theta-join

Un'operatore derivato.
Siano r_1 e r_2 due relazioni senza attributi in comune.

$$r_1 \bowtie_F r_2 = \sigma_F(r_1 \bowtie r_2)$$

Il theta-join è un prodotto cartesiano seguito da una selezione.

Se F è una congiunzione di atomi di uguaglianza, abbiamo un **equi-join**.

impiegati

Impiegato	Progetto
Rossi	A
Neri	A
Neri	B

progetti

Codice	Nome
A	Venere
B	Marte

impiegati $\bowtie_{\text{Progetto}=\text{Codice}}$ progetti

Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	A	Venere
Neri	B	B	Marte

Espressioni ed interrogazioni

Un'**interrogazione** è una funzione che, applicata a istanze di base di dati, produce istanze di relazione.

- U_∞ insieme (infinito) numerabile di attributi (per semplicità tutti sullo stesso dominio); $\mathcal{U}$ insieme di tutte le relazioni su attributi in U_∞
- **R** schema di base di dati; $I(\mathbf{R})$ insieme delle istanze di **R**

Un'interrogazione Q è una funzione:

$$Q : I(\mathbf{R}) \rightarrow \mathcal{U}$$

Di solito gli attributi del risultato sono fissati.
Esiste $X \subset U_\infty$ tale che

$$Q : I(\mathbf{R}) \to \mathcal{X}$$

dove $\mathcal{X}$ è l'insieme di tutte le relazioni su X

$Q(\mathbf{r})$ indica il **risultato** dell'applicazione di Q ad $\mathbf{r}$
(può non essere definito — Q è in generale una funzione parziale).

Le espressioni dei vari linguaggi di interrogazione (es. l'algebra relazionale) "rappresentano" o "realizzano" interrogazioni: ogni espressione definisce una funzione.

$E(\mathbf{r})$ è il il **risultato** dell'applicazione di E ad $\mathbf{r}$

Due espressioni (dello stesso linguaggio o di due linguaggi diversi) sono **equivalenti** se rappresentano la stessa interrogazione:

- $E_1 \equiv_{\mathbf{R}} E_2$ se $E_1(\mathbf{r}) = E_2(\mathbf{r})$ per ogni $\mathbf{r} \in I(\mathbf{R})$.
- $E_1 \equiv E_2$ se $E_1 \equiv_{\mathbf{R}} E_2$ per ogni schema $\mathbf{R}$ definibile su U_∞.

$$\pi_{AB}(\sigma_{A>0}(R)) \equiv \sigma_{A>0}(\pi_{AB}(R))$$

$$\pi_{AB}(R_1) \bowtie \pi_{AC}(R_2) \equiv_{\mathbf{R}} \pi_{ABC}(R_1 \bowtie R_2)$$

(se $\mathbf{R}$ contiene $R_1(X_1)$ e $R_2(X_2)$ e $X_1 \cap X_2 = A$)

Due linguaggi sono equivalenti se permettono di formulare le stesse interrogazioni.

- L_1 è **espressivo almeno quanto** L_2 se
  per ogni espressione E_2 di L_2 e
  per ogni schema di base di dati $\mathbf{R}$
  esiste un'espressione E_1 di L_1 tale che $E_1 \equiv_{\mathbf{R}} E_2$.
- L_1 e L_2 sono **equivalenti** se
  ognuno è espressivo almeno quanto l'altro.

In algebra relazionale, le interrogazioni su uno schema di base di dati **R** vengono formlate con espressioni i cui atomi sono:

- (nomi di) relazione in **R** (le “variabili”);
- relazioni costanti (che non fanno parte delle istanze di **R**).

Abbiamo variabili e costanti come nelle espressioni di altre algebre note.

Relazioni costanti:

- Per ogni relazione su

  $$PERSONE(Nome,Sesso,\ldots)$$

  generare una relazione con associato agli uomini il titolo “Signor” e alle donne “Signora”.

- I due valori non sono contenuti nella base di dati e gli operatori non possono “creare” valori.
- Il risultato desiderato si ottiene con:

  $$PERSONE \bowtie TITOLI$$

  dove *TITOLI* è una relazione costante:

TITOLI

Sesso	*Titolo*
M	Signor
F	Signora

p_1

Nome	*Sesso*	...
Neri	F	...
Bianchi	F	...
Verdi	M	...

p_2

Nome	*Sesso*	...
Rossi	M	...
Bini	F	...

$p_1 \bowtie TITOLI$

Titolo	*Nome*	*Sesso*	...
Signora	Neri	F	...
Signora	Bianchi	F	...
Signor	Verdi	M	...

$p_2 \bowtie TITOLI$

Titolo	*Nome*	*Sesso*	...
Signor	Rossi	M	...
Signora	Bini	F	...

PERSONE(<u>Nome</u>,Eta,Reddito)

PATERNITA(Padre,<u>Figlio</u>)

MATERNITA(Madre,<u>Figlio</u>)

persone

Nome	Eta	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

maternita

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

paternita

Padre	Figlio
Sergio	Franco
Franco	Andrea
Franco	Aldo
Luigi	Olga
Luigi	Filippo

“Trovare nome e reddito delle persone con meno di 30 anni”

$$\pi_{Nome,Reddito}(\sigma_{Eta<30}(persone))$$

Nome	*Reddito*
Andrea	21
Aldo	15
Filippo	30

“Trovare i padri di persone che guadagnano più di venti milioni”

$$\pi_{Padre}(paternita \bowtie_{Figlio=Nome} (\sigma_{reddito>20}(persone)))$$

Padre
Franco
Luigi

“Trovare i padri i cui figli guadagnano tutti più di venti milioni”

$$\pi_{Padre}(paternita) -$$
$$-\pi_{Padre}(paternita \bowtie_{Figlio=Nome} (\sigma_{reddito \leq 20}(persone)))$$

Padre
Luigi

“Trovare le persone che guadagnano più dei rispettivi padri, mostrandone nomi, redditi e nomi dei rispettivi padri”

$$\pi_{N,R,P}(\sigma_{R>RP}((persone \bowtie_{N=F} paternita)$$
$$\bowtie_{P=NP}$$
$$(\rho_{NP,EP,RP \leftarrow N,E,R}(persone))))$$

Nome	*Reddito*	*Padre*
Andrea	21	Franco
Olga	41	Luigi

Schema della base di dati di una dinastia reale:

REIGNS (Sovereign, From, To)
PERSONS (Name, Sex, Birth, Death)
FATHERHOOD (Father, Child)
MOTHERHOOD (Mother, Child)

reigns

Sovereign	*From*	*To*
James I	1603	1625
Charles I	1625	1648
Charles II	1660	1685
James II	1685	1688
Mary II	1688	1694
Anne	1702	1714

persons

Name	*Sex*	*Birth*	*Death*
James I	M	1566	1625
Elizabeth	F	1590	1662
Charles I	M	1600	1649
Charles II	M	1630	1685
Mary	F	1631	1659
James II	M	1633	1701
Henrietta A.	F	1640	1670
Mary II	F	1662	1694
Anne	F	1665	1714
James F.E.	M	1686	1766

fatherhood

Father	*Child*
Lord Darnley	James I
James I	Elizabeth
James I	Charles I
Charles I	Charles II
Charles I	Mary
Charles I	James II
Charles I	Henrietta A.
James II	Mary II
James II	Anne
James II	James F.E.

motherhood

Mother	*Child*
Mary Stuart	James I
Anne of Denmark	Elizabeth
Anne of Denmark	Charles I
Henrietta Maria	Charles II
Henrietta Maria	Mary
Henrietta Maria	James II
Henrietta Maria	Henrietta A.
Anne Hyde	Mary II
Anne Hyde	Anne
Mary of Modena	James F.E.

REIGNS (Sovereign, From, To)
PERSONS (Name, Sex, Birth, Death)
FATHERHOOD (Father, Child)
MOTHERHOOD (Mother, Child)

"i sovrani che hanno regnato fino alla morte"

$$(\rho_{Name, Death \leftarrow Sovn, To}(REIGNS)) \bowtie PERSONS$$

Name	*Sex*	*Birth*	*From*	*Death*
James I	M	1566	1603	1625
Charles II	M	1633	1660	1685
Mary II	F	1662	1688	1694
Anne	F	1665	1702	1714

REIGNS (Sovereign, From, To)
PERSONS (Name, Sex, Birth, Death)
FATHERHOOD (Father, Child)
MOTHERHOOD (Mother, Child)

"i figli (noti alla base di dati) dei sovrani della famiglia"

$$\pi_{Sovn,\ Child}(REIGNS \bowtie (\rho_{Sovn \leftarrow Father}(FATHERHOOD) \cup \rho_{Sovn \leftarrow Mother}(MOTHERHOOD))$$

Sovereign	*Child*
James I	Elizabeth
James I	Charles I
Charles I	Charles II
Charles I	Mary
Charles I	James II
Charles I	Henrietta A.
James II	Mary II
James II	Anne
James II	James F.E.