ALGEBRA RELAZIONALE
(CAPITOLO 3)

R. Basili

a.a. 2013-2014

Algebra Relazionale

- **Linguaggio di interrogazione o query language è un linguaggio tramite il quale è possibile richiedere le informazioni contenute in una base di dati.**

- **Algebra relazionale: procedurale, solo teorico**
- **Calcolo relazionale: dichiarativo, solo teorico**
- **Datalog: dichiarativo, poco diffuso**

Operatori su relazioni

- Il risultato dell'applicazione di un operatore ad una relazione è una relazione.
- Posso quindi costruire espressioni e query complesse

Operatori

- Unione (U), Differenza (-) e Intersezione (∩)
- Ridenominazione (ρ), Selezione (σ) e Proiezione (π)
- Join (equijoin, naturale o prodotto cartesiano)

Operatori insiemistici

- E' possibile applicare gli operatori insiemistici solamente a coppie di relazioni OMOGENEE, cioè definite sugli stessi attributi.
- Date le relazioni A e B definite sullo stesso schema:
- UNIONE (AUB) = relazione contenente le tuple presenti in A ed in B
- DIFFERENZA (A–B) = relazione contenente le tuple presenti in A ma non in B
- INTERSEZIONE ($A \cap B$) = relazione contenente le tuple presenti sia in A che in B

(Nota: $A \cap B = A-(A-B)$)

Union Compatibility

- Due relazioni sono *Union compatible* sse:
  - Esse hanno lo stesso numero di campi
  - Campi corrispondenti secondo l’ordine sx->dx hanno lo stesso dominio

- Solo in tal caso quini sono applicabili gli operatori insiemistici di unione, differenza ed intersezione

Operatore di Ridenominazione (ρ)

- Sia r una relazione definita sull'insieme di attributi X e sia Y un (altro) insieme di attributi con la stessa cardinalità. Siano A1 A2... Ak e B1 B2... Bk rispettivamente un ordinamento per gli attributi in X ed Y allora la RIDENOMINAZIONE
- ρB1B2..Bk $\rightarrow$ A1A2..Ak(r)
- contiene una tupla t' per ciascuna tupla t$\in$r tale che:

  t'[Bi]=t[Ai]   i=1,...,k

- La RIDENOMINAZIONE modifica solamente il nome degli attributi agendo sullo schema della relazione e lasciando inalterata la sua istanza

Operatore di Selezione (σ)

- $\sigma_{eta'=18}(S)$

$\sigma_F(r)$

Formula proposizionale

Formula proposizionale F su X è una formula booleana (¬,∧,∨) di *termini* T.

Dove T e' definito come:

attr op attr o ***attr op c***

op è un operatore di confronto (=,≠,≤,≥,<,>);

S

sid	nome	login	età	gpa
0012	Rossi	ro@ec	18	3.4
0072	Verdi	bi@ec	19	3.2
0033	Bianchi	bi@tt	18	3.8

sid	nome	login	età	gpa
0012	Rossi	ro@ec	18	3.4
0033	Bianchi	bi@tt	18	3.8

Produce una relazione avente lo stesso schema di r e contenente le tuple di r per le quali F e' **vera**

Operatore di Proiezione (π)

- π età (S)

$\pi_{C}(r)$

Condizione

S

sid	nome	login	età	gpa
0012	Rossi	ro@ec	18	3.4
0072	Verdi	bi@ec	19	3.2
0033	Bianchi	bi@tt	18	3.8

Condizione di proiezione *C* su r è l'elenco degli attributi dello schema X di r che debbono venire 'proiettati':

$attr_1$, $attr_2$, … ,$attr_n$

$attr_i \in X$ per $1 \geq i \leq n$

età
18
19

Distinte

Tutte le tuple della relazione r contribuiscono al risultato di una proiezione, ma soltanto con i valori corrispondenti agli attributi specificati dalla condizione C.

Esempi di Istanze:

Sailors (1)

S1

sid	sname	rating	age
22	Davide	7	45.0
31	Luca	8	55.5
58	Remo	10	35.0

Sailors (2)

S2

sid	sname	rating	age
28	Eolo	9	35.0
31	Luca	8	55.5
44	Giulio	5	35.0
58	Remo	10	35.0

Reserves

R1

sid	bid	day
22	101	10/10/03
58	103	11/12/02

Boats

B1

bid	bname	color
101	Futura	verde
58	Onda	rossa

Il Prodotto Cartesiano

R X S

Ogni tupla di S viene associata ad ogni tupla di R. Lo schema risultante ha tutti gli attributi di R seguiti da quelli di S.

S1 X R1

(sid)	sname	rating	age	(sid)	bid	day
22	Davide	7	45.0	22	101	10/10/03
22	Davide	7	45.0	58	103	11/12/02
31	Luca	8	55.5	22	101	10/10/03
31	Luca	8	55.5	58	103	11/12/02
58	Remo	10	35.0	22	101	10/10/03
58	Remo	10	35.0	58	103	11/12/02

In caso di *conflitto* sui nomi degli attributi di R ed S applico l'operatore di ***ridenominazione*** :

$\rho(C(1 \rightarrow sid1, 5 \rightarrow sid2), S1 X R1)$

Operazioni di join

$$R \bowtie_C S = \sigma_C(R \times S)$$

Condizione di Join

La relazione risultante avrà lo **schema** del prodotto cartesiano di R ed S e **l'istanza** composta dalle tuple del prodotto cartesiano che soddisfano la condizione C

$$S \bowtie_{S1.sid<R1.sid} R$$

(sid)	sname	rating	age	(sid)	bid	day
22	Davide	7	45.0	58	103	11/12/02
31	Luca	8	55.5	58	103	11/12/02

Operazioni di Equi-Join e Natural-Join

$$R \bowtie_{C} S$$

Condizione di Join

- ***Equi-Join*** è un join in cui la condizione ***C*** e' composta solamente di operazioni di uguaglianza.
- ***Natural Join*** e'un Equi-Join su ***tutti*** gli attributi in comune

$$S \bowtie_{sid} R$$

(sid)	sname	rating	age	(sid)	bid	day
22	Davide	7	45.0	~~22~~	101	11/10/03
58	Remo	10	35.0	~~58~~	103	11/12/02

Operazione di Divisione

$$\boxed{A/B} = \pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$$

Non e' supportato come primitiva ma e' utile per esprimere query del tipo:

*Trova i marinai che hanno prenotato **tutte** le barche*

$A/B = \{ <x> \mid \exists <x,y> \in A \;\; \forall <y> \in B \}$

L'IDEA è cogliere la natura distributiva del prodotto cartesiano. La *divisione* è l'inversa del prodotto cartesiano nel senso seguente:

A/B = Q (quoziente) **sse** esiste R tc

$$Q \times B \cup R = A$$

Operazione di Divisione

$$A/B = \pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$$

Non e' supportato come primitiva ma e' utile per esprimere query del tipo:

*Trova i marinai che hanno prenotato **tutte** le barche*

$Pren/B = \{ \langle x \rangle \mid \exists \langle x,y \rangle \in A \ \forall \langle y \rangle \in B \}$

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

Pren

pno
p2

B1

pno
p2
p4

B2

Pren/B1

sno
s1
s3

Pren/B2

sno
s1

Q1

Trova il nome dei marinai (tabella S) che hanno riservato la barca numero 103.

$$\pi_{sname}(\sigma_{bid=103}(Reserves \bowtie Sailors))$$

Opt

$$\boxed{\pi_{sname}((\sigma_{bid=103}Reserves) \bowtie Sailors)}$$

In maniera più elegante:

$$\rho(Temp1, \sigma_{bid=103}Reserves)$$

$$\rho(Temp2, Temp1 \bowtie Sailors)$$

$$\pi_{sname}(Temp2)$$

Q2

Trova il nome dei marinai (tabella S) che hanno riservato la barca rossa.

$$\pi_{sname}(\sigma_{color='rossa'} \text{Boats} \bowtie \text{Reserves} \bowtie \text{Sailors})$$

Opt

$$\boxed{\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='rossa'}\text{Boats})\bowtie\text{Reserves})\bowtie\text{Sailors})}$$

In maniera più elegante:

$$\rho(\text{Temp1}, \pi_{bid}\sigma_{color='rossa'}\text{Boats})$$

$$\rho(\text{Temp2}, \pi_{sid}(\text{Temp1}\bowtie\text{Reserves}))$$

$$\rho(\text{Temp3},\text{Temp2} \bowtie \text{Sailors})$$

$$\boxed{\pi_{sname}(\text{Temp3})}$$

Q3 e Q4

Trova il nome dei marinai (tabella S) che hanno riservato una barca rossa **o** *una barca verde.*

$$\rho(TempBoats, (\sigma_{color='rossa'} Boats) \cup (\sigma_{color='verde'} Boats))$$

oppure:

$$\rho(TempBoats, (\sigma_{color='rossa' \vee color='verde'} Boats))$$

$$\boxed{\pi_{sname}(TempBoats \bowtie Reserves \bowtie Sailors)}$$

Trova il nome dei marinai (tabella S) che hanno riservato una barca rossa **ed** *una barca verde.*

$$\rho(TempRossa, \pi_{sid}(\sigma_{color='rossa'} Boats) \bowtie Reserves)$$

$$\rho(TempVerde, \pi_{sid}(\sigma_{color='verde'} Boats) \bowtie Reserves)$$

$$\boxed{\pi_{sname}((TempRossa \cap TempVerde) \bowtie Sailors)}$$

Perché non abbiamo utilizzato la congiunzione?

Q5 e Q6

Trova il nome dei marinai (tabella S) che hanno riservato almeno due barche

$$\rho(Reservations, \pi_{sid,\ sname,bid}(\text{Sailors} \bowtie \text{Reserves}))$$

$$\rho(ReservationPairs(1 \rightarrow \text{sid1}, 2 \rightarrow \text{sname1}, 3 \rightarrow \text{bid1}, 4 \rightarrow \text{sid2}, 5 \rightarrow \text{sname2}, 6 \rightarrow \text{bid2}), \text{Reservations} \times \text{Reservations}$$

$$\boxed{\pi_{\text{sname2}}\ \sigma_{(sid1=sid2) \wedge (bid1 \neq bid2)}\ \text{ReservationPairs}}$$

Trova il sid dei marinai (tabella S) che hanno più di 20 anni e che non hanno riservato una barca rossa.

$$\boxed{\begin{array}{l}\pi_{\text{sid}}(\sigma_{age>20}\text{Sailors}) - \\ \qquad \pi_{\text{sid}}((\sigma_{color='rossa'}\text{Boats}) \bowtie \text{Reserves})\end{array}}$$

Allegato A –
Alcune regole di equivalenza

- Il **join naturale** è **commutativo ed associativo**:
$E_1 \rhd\lhd E_2 \equiv E_2 \rhd\lhd E_1$
$(E_1 \rhd\lhd E_2) \rhd\lhd E_3 = E_1 \rhd\lhd (E_2 \rhd\lhd E_3) = E_1 \rhd\lhd E_2 \rhd\lhd E_3$

- **Selezione e proiezione** si possono **raggruppare**:

$$\sigma_{F1}(\sigma_{F2}(E)) = \sigma_{F1 \wedge F2}(E) \quad \pi_Y(\pi_{xy}(E)) = \pi_y(E)$$

- **Selezione** e **proiezione commutano** (F si riferisce solo ad attributi in Y):

$$\pi_Y(\sigma_F(E)) = \sigma_F(\pi_Y(E))$$

- "*Push-down*" della selezione rispetto al join (F è nello schema di E_1):

$$\sigma_F(E_1 \rhd\lhd E_2) = \sigma_F(E_1) \rhd\lhd E_2$$

Allegato A – Osservazione: *equivalenza* rispetto a *identità*

- La equivalenza $\equiv$ usata nella regola:

$$E_1 \rhd\lhd E_2 \equiv E_2 \rhd\lhd E_1$$

È diversa dalla identità = che vale invece in:

$$(E_1 \rhd\lhd E_2) \rhd\lhd E_3 = E_1 \rhd\lhd (E_2 \rhd\lhd E_3) = E_1 \rhd\lhd E_2 \rhd\lhd E_3$$

- La equivalenza $\equiv$ non richiede la identità anche rispetto allo schema logico. Osservate che infatti $E_1 \rhd\lhd E_2$ *ed* $E_2 \rhd\lhd E_1$ hanno schemi logici diversi, nel caso generale. Quindi affinchè $A \equiv B$ basta che esistano regole di scambio di righe tali che i due schemi siano riconducibili l'uno all'altro e che le tuple così ottenute siano le stesse.

- L'identità algebrica espressa da $A=B$ richiede che i due insiemi siano identici rispetto <u>sia</u> allo schema <u>che</u> ai valori.

Domande

- Che significa che gli operatori di algebra relazionale possono essere composti?
- Qual'e' la cardinalità minima e massima di
  - $R \times \sigma_F R$
  - $\sigma_{F2} R \cup \sigma_{F1} R$
  - $(\sigma_F R1 \times R2) \cap (R1 \times \sigma_F R2)$

  in funzione di |R|, |R1|, |R2|?
- Dimostrare sotto quali condizioni:
  - $\sigma_F R \times A \subseteq R \times A$
  - $(A \cup B) \cap C = (A \cup B) \setminus \overline{C}$
  - $(\sigma_F R1 \times R2) \neq (R1 \times \sigma_F R2)$