

Aggiornamenti e Interrogazioni

Aggiornamento: inserimento o modifica di dati (una funzione che, data un'istanza di basi di dati ne produce un'altra)

Interrogazione: estrazione di informazioni (una funzione che a partire da una base di dati, produce una relazione su un schema specificato)

Esempio di interrogazione: dato lo schema

Persone(Nome, Età, Reddito)

Paternità(Padre, Figlio)

Maternità(Madre, Figlio)

Trovare nome e reddito delle persone con meno di 30 anni

Nome	Reddito
Andrea	21
Aldo	15

Linguaggi di interrogazione (query languages)

Algebra relazionale: procedurale, cioè in cui le operazioni complesse vengono specificate descrivendo il procedimento da seguire per ottenere la soluzione

Calcolo relazionale: dichiarativo, cioè in cui le espressioni descrivono le proprietà del risultato, piuttosto che la procedura per ottenerlo

SQL (Structured Query Language): linguaggio usato nella pratica che combina aspetti dichiarativi e procedurali

Algebra relazionale

Un insieme di operatori su relazioni che producono relazioni e che possono essere la base per espressioni complesse di interrogazione

Operatori dell'algebra relazionale:

- operatori insiemistici: unione, intersezione, differenza
- ridenominazione
- selezione
- proiezione
- join naturale, prodotto cartesiano, theta-join

Operatori insiemistici

Le relazioni sono insiemi di tuple omogenee. I risultati degli operatori insiemistici sono a loro volta insiemi di tuple omogenee con quelle di partenza

Quindi:

E' possibile applicare gli operatori insiemistici (unione, intersezione, differenza) solo a coppie di relazioni definite sugli stessi attributi (stessi nomi e stessi domini)

Unione

laureati

Matr	Cognome	Età
7345	Bianchi	37
3492	Verdi	40
3877	Neri	25

quadri

Matr	Cognome	Età
7111	Viola	50
3492	Verdi	40
3877	Neri	25

laureati \cup quadri

Matr	Cognome	Età
7111	Viola	50
3492	Verdi	40
3877	Neri	25
7345	Bianchi	37

Intersezione

laureati

Matr	Cognome	Età
7345	Bianchi	37
3492	Verdi	40
3877	Neri	25

quadri

Matr	Cognome	Età
7111	Viola	50
3492	Verdi	40
3877	Neri	25

laureati \cap quadri

Matr	Cognome	Età
3492	Verdi	40
3877	Neri	25

Differenza

laureati

Matr	Cognome	Età
7345	Bianchi	37
3492	Verdi	40
3877	Neri	25

quadri

Matr	Cognome	Età
7111	Viola	50
3492	Verdi	40
3877	Neri	25

laureati - quadri

Matr	Cognome	Età
7345	Bianchi	37

La differenza rende l'intersezione ridondante, in quanto $r \cap s = r - (r - s)$

Operazione impossibile

Alcune operazioni sono sensate ma scorrette

Paternità	Padre	Figlio
	Adamo	Caino
	Adamo	Abele
	Abramo	Isacco
	Abramo	Ismaele

Maternità	Madre	Figlio
	Eva	Caino
	Eva	Set
	Sara	Isacco
	Agar	Ismaele

Paternità \cup Maternità ??

Ridenominazione

E' un operatore *monadico* (prende un solo argomento, è “unario”)

Intuitivamente, modifica lo schema della relazione cui si applica lasciando inalterata l'istanza

Permette di superare le limitazioni imposte agli operatori insiemistici

Viene denotato con la lettera greca ρ (pronuncia rho)

Ridenominazione: esempio

Paternità	Padre	Figlio
	Adamo	Caino
	Adamo	Abele
	Abramo	Isacco
	Abramo	Ismaele

$\rho_{\text{Genitore} \leftarrow \text{Padre}}(\text{Paternità})$

Genitore	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Abramo	Ismaele

Ridenominazione: altro esempio

Impiegati

Cognome	Agenzia	Stipendio
Bianchi	Milano	45
Verdi	Torino	50

$\rho_{\text{Sede} \leftarrow \text{Agenzia}, \text{Salario} \leftarrow \text{Stipendio}}(\text{Impiegati})$

Cognome	Sede	Salario
Bianchi	Milano	45
Verdi	Torino	50

Esempio di uso

Paternità

Padre	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Abramo	Ismaele

Maternità

Madre	Figlio
Eva	Caino
Eva	Set
Sara	Isacco
Agar	Ismaele

$$\rho_{\text{Genitore} \leftarrow \text{Padre}} (\text{Paternità}) \cup \rho_{\text{Genitore} \leftarrow \text{Madre}} (\text{Maternità})$$

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

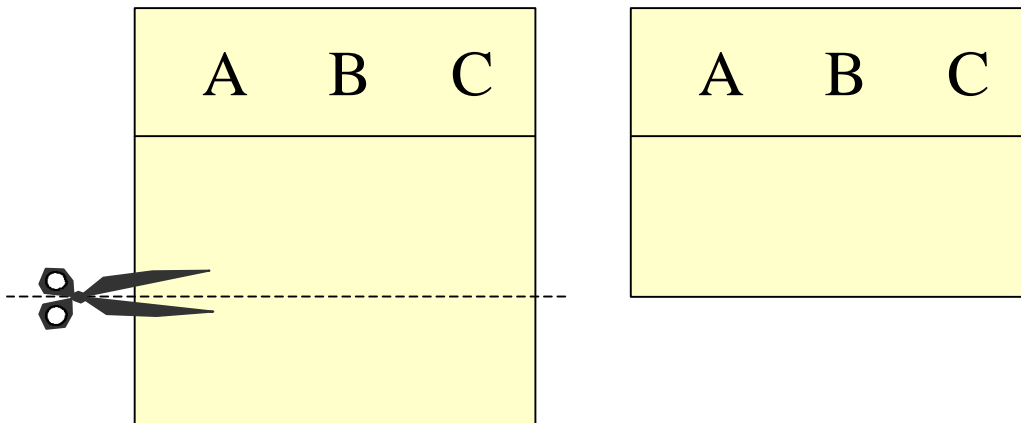
Selezione

Un altro operatore monadico

Produce una relazione che:

- ha lo stesso schema dell'operando
- contiene un sottoinsieme delle tuple dell'operando

Produce “decomposizioni orizzontali”



Si denota con la lettera greca σ (sigma)

Selezione: esempio

Impiegati

Cognome	Età	Stipendio
Bianchi	29	5.000.000
Rossi	25	3.100.000
Verdi	50	5.550.000
Neri	25	6.000.000

$\sigma_{\text{Età} < 30 \wedge \text{Stipendio} > 4.000.000}(\text{Impiegati})$

Cognome	Età	Stipendio
Bianchi	29	5.000.000
Neri	25	6.000.000

Selezione: sintassi

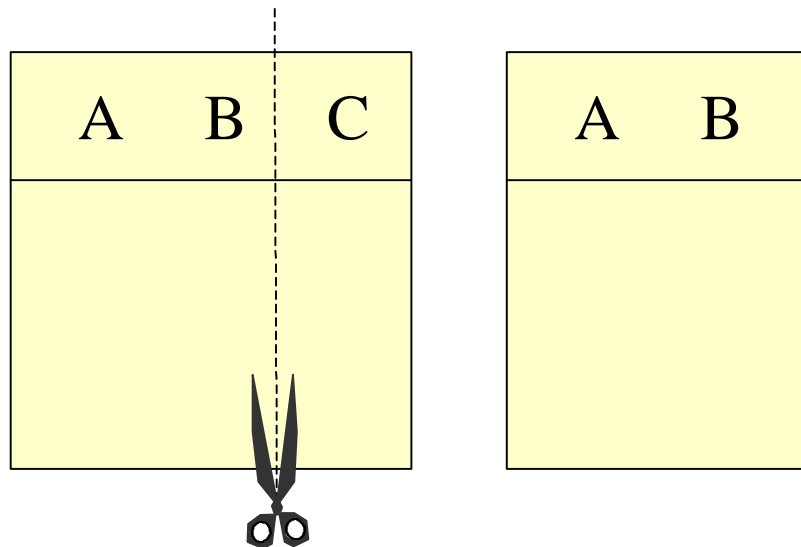
Sono ammesse espressioni booleane, cioè con connettivi \wedge , \vee , e \neg (AND, OR e NOT), i cui atomi sono confronti (con operatori di uguaglianza, disuguaglianza e ordinamento) tra attributi o valori di attributo

Proiezione

E' un operatore monadico che produce una relazione:

- su un sottoschema della relazione di partenza
- a cui contribuiscono tutte le tuple della relazione

Produce “decomposizioni verticali”



Si denota con la lettera greca π (pi)

Proiezione: esempio

Imp

Cognome	Nome	Reparto	Superiore
Rossi	Mario	Vendite	De Rossi
Neri	Luca	Vendite	De Rossi
Verdi	Nicola	Personale	Bianchi
Rossi	Elena	Personale	Bianchi

$\pi_{\text{Cognome, Nome}}(\mathbf{Imp})$

Cognome	Nome
Rossi	Mario
Neri	Luca
Verdi	Nicola
Rossi	Elena

$\pi_{\text{Reparto, Superiore}}(\mathbf{Imp})$

Reparto	Superiore
Vendite	De Rossi
Personale	Bianchi

Esempio per interrogazioni

Maternità

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

Paternità

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

Persone

Nome	Età	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

Interrogazione semplice

Con riferimento al seguente schema:

Persone(Nome, Età, Reddito)

Paternità(Padre, Figlio)

Maternità(Madre, Figlio)

**Trovare nome e reddito delle
persone con meno di 30 anni**

$\pi_{\text{Nome, Reddito}}(\sigma_{\text{Età} < 30}(\text{Persone}))$

Nome	Reddito
Andrea	21
Aldo	15
Filippo	30

Join (naturale)

Operatore binario (generalizzabile)

Correla dati di relazioni diverse

E' l'operatore più caratteristico dell'algebra relazionale

Si denota con \bowtie

Date due relazione $r_1(X_1)$ e $r_2(X_2)$

$r_1 \bowtie r_2$ è definita su $X_1 \cup X_2$

$r_1 \bowtie r_2$ ha tutte le tuple t tali che:

$$t[X_1] \in R_1$$

$$t[X_2] \in R_2$$

Join naturale “completo”

r_1

Impiegato	Sede
Rossi	Milano
Neri	Roma
Verdi	Roma

r_2

Sede	Superiore
Roma	De Rossi
Milano	Bianchi

$r_1 \bowtie r_2$

Impiegato	Sede	Superiore
Rossi	Milano	Bianchi
Neri	Roma	De Rossi
Verdi	Roma	De Rossi

*Un join è detto **completo** se ogni tupla contribuisce al risultato*

Join (e tuple *dangling*)

r_1

Impiegato	Sede
Rossi	Milano
Neri	Roma
Verdi	Perugia

r_2

Sede	Superiore
Roma	De Rossi
Torino	Gini
Milano	Bianchi

$r_1 \bowtie r_2$

Impiegato	Sede	Superiore
Rossi	Milano	Bianchi
Neri	Roma	De Rossi

*Le tuple sono dette **dangling** se non sono combinabili con le tuple dell'altra relazione*

Join naturale “vuoto”

r_1

Impiegato	Sede
Rossi	Milano
Neri	Roma
Verdi	Perugia

r_2

Sede	Superiore
Verona	De Rossi
Torino	Gini
Napoli	Bianchi

$r_1 \bowtie r_2$

Impiegato	Sede	Superiore
-----------	------	-----------

Un join vuoto (tutte le tuple sono dangling)

Join (esempi)

r_1

Impiegato	Progetto
Rossi	A
Neri	A

r_2

Progetto	Supervisore
A	De Rossi
A	Bianchi

$r_1 \bowtie r_2$

Impiegato	Progetto	Superiore
Rossi	A	De Rossi
Rossi	A	Bianchi
Neri	A	De Rossi
Neri	A	Bianchi

*Un join completo con $|r_1| \times |r_2|$ tuple:
ogni tupla di r_1 è combinata con tutte le
tuple di r_2 (e viceversa)*

Riflessioni sulla cardinalità del risultato

$r_1 \bowtie r_2$ contiene un numero di tuple
compreso tra 0 e $|r_1| \times |r_2|$

Se il join è completo \Rightarrow contiene
almeno un numero di tuple pari al
massimo tra $|r_1|$ e $|r_2|$

Se gli attributi comuni contengono una
chiave per $r_1 \Rightarrow r_1 \bowtie r_2$ contiene al più
 $|r_2|$ tuple

Se gli attributi comuni contengono una
chiave per $r_2 \Rightarrow r_1 \bowtie r_2$ contiene al più
 $|r_1|$ tuple

Uso del join

Paternità

Padre	<u>Figlio</u>
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Abramo	Ismaele

Maternità

Madre	<u>Figlio</u>
Eva	Caino
Eva	Set
Sara	Isacco
Agar	Ismaele

Paternità \bowtie Maternità

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

Figlio è chiave per entrambe le due relazioni

Join e attributi

Il Join di due relazioni sugli stessi attributi è pari all'intersezione delle due relazioni:

$$r_1(\mathbf{X}) \bowtie r_2(\mathbf{X}) = r_1(\mathbf{X}) \cap r_2(\mathbf{X})$$

Il Join di due relazioni r_1 e r_2 che non hanno attributi in comune viene chiamato **prodotto cartesiano** e contiene sempre $|r_1| \times |r_2|$ ennuple

NOTA: questo prodotto cartesiano non coincide con il tradizionale prodotto cartesiano tra insiemi

Esempio di prodotto cartesiano

Operai

Operaio	Cantiere
Adami	A
Verdi	A
Rossi	B

Cantieri

Codice	Zona
A	Stadio
B	Centro

Operai \times Cantieri

Operaio	Cantiere	Codice	Zona
Adami	A	A	Stadio
Adami	A	B	Centro
Verdi	A	A	Stadio
Verdi	A	B	Centro
Rossi	B	A	Stadio
Rossi	B	B	Centro

Join naturale: proprietà

Il join naturale è **commutativo**:

$$r_1 \bowtie r_2 = r_2 \bowtie r_1, \text{ per ogni } r_1 \text{ e } r_2$$

Il join naturale è **associativo**:

$$r_1 \bowtie (r_2 \bowtie r_3) = (r_1 \bowtie r_2) \bowtie r_3$$

E' quindi possibile definire un operatore join n-ario attraverso ripeture dell'operatore unario

Theta-join

E' un'operatore derivato: se r_1 e r_2 sono 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 un AND di atomi di ugualianza, abbiamo un **equi-join**

Esempio di equi-join

Operai

Operaio	Cantiere
Adami	A
Verdi	A
Rossi	B

Cantieri

Codice	Zona
A	Stadio
B	Centro

Operai \bowtie **Cantieri**
Cantiere=Codice

Operaio	Cantiere	Codice	Zona
Adami	A	A	Stadio
Verdi	A	A	Stadio
Rossi	B	B	Centro

Interrogazione

Con riferimento al seguente schema:

Persone(Nome, Età, Reddito)

Paternità(Padre, Figlio)

Maternità(Madre, Figlio)

Trovare i padri di persone che guadagnano più di venti milioni

$\pi_{\text{padre}}(\text{Paternità} \bowtie_{\text{Figlio=Nome}}$

$(\sigma_{\text{Reddito}>20}(\text{Persone})))$

Padre
Luigi
Franco

Interrogazione

Con riferimento al seguente schema:

Persone(Nome, Età, Reddito)

Paternità(Padre, Figlio)

Maternità(Madre, Figlio)

**Trovare i padri i cui figli
guadagnano tutti più di venti milioni**

$\pi_{\text{padre}}(\text{Paternità}) -$

$\pi_{\text{padre}}(\text{Paternità} \bowtie_{\text{Figlio=Nome}}$

$(\sigma_{\text{Reddito} \leq 20}(\text{Persone})))$

Padre
Luigi

Interrogazione

Con riferimento al seguente schema:

Persone(Nome, Età, Reddito)

Paternità(Padre, Figlio)

Maternità(Madre, Figlio)

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

$\pi_{N,R,P}(\sigma_{R>RP}((\text{Persone} \bowtie_{N=F} \text{Paternità}))$

$\bowtie_{P=NP}$

$(\rho_{NP,EP,RP \leftarrow N,E,P}(\text{Persone}))))$

Nome	Reddito	Padre
Andrea	21	Franco
Olga	41	Luigi