

Algebra Relazionale



Join

- # Combinando selezione e proiezione, è possibile estrarre informazioni da una relazione
- # Non è possibile però correlare informazioni presenti in relazioni diverse
- # Il join è l'operatore più interessante dell'algebra relazionale: permette di correlare dati in relazioni diverse

Join naturale

Operatore binario

Produce un risultato

- sull'unione degli attributi degli operandi
- con ennuple costruite ciascuna a partire da una ennupla di ognuno degli operandi

Join naturale, sintassi e semantica

$R_1(X_1), R_2(X_2)$

$R_1 \triangleright \triangleleft R_2$ è una relazione su $X_1 X_2$

$\{ t \text{ su } X_1 X_2 \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2$
con $t[X_1] = t_1$ e $t[X_2] = t_2 \}$

analogamente

$\{ t \text{ su } X_1 X_2 \mid t[X_1] \in r_1 \text{ e } t[X_2] \in r_2 \}$

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Mori
B	Bruni

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

ogni ennupla contribuisce al risultato:

- join completo

Un join non completo

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Un join vuoto

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
D	Mori
C	Bruni

Impiegato	Reparto	Capo
-----------	---------	------

Un join completo, con $n \times m$ ennuple

Impiegato	Reparto	Reparto	Capo
Rossi	B	B	Mori
Neri	B	B	Bruni

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

Cardinalità del join

- # Il join di R_1 e R_2 contiene un numero di ennuple compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$
- # Se il join coinvolge una chiave di R_2 , allora il numero di ennuple è compreso fra zero e $|R_1|$
- # Se il join coinvolge una chiave di R_2 e un vincolo di integrità referenziale, allora il numero di ennuple è pari a $|R_1|$

Join, una difficoltà

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

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

alcune ennuple non contribuiscono al risultato:
vengono "tagliate fuori"

Join esterno

- # Il join esterno estende, con valori nulli, le ennuple che verrebbero tagliate fuori da un join (interno)
- # Esiste in tre versioni:
 - sinistro, destro, completo

Join esterno

- Sinistro: mantiene tutte le ennuple del primo operando, estendendole con valori nulli, se necessario
- Destro: ... del secondo operando ...
- Completo: ... di entrambi gli operandi ...

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati ▷◁_{LEFT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\triangleright \triangleleft$ RIGHT Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\triangleright \triangleleft_{FULL}$ Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

Join e proiezioni

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

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Impiegato	Reparto	Reparto	Capo
Neri	B	B	Mori
Bianchi	B		

Proiezioni e join

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Bruni
Verdi	A	Bini

Impiegato	Reparto
Neri	B
Bianchi	B
Verdi	A

Reparto	Capo
B	Mori
B	Bruni
A	Bini

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Bruni
Neri	B	Bruni
Bianchi	B	Mori
Verdi	A	Bini

Join e proiezioni

$R_1(X_1), R_2(X_2)$

$$\pi_{X_1}(R_1 \bowtie R_2) \subseteq R_1$$

$R(X), X = X_1 \cup X_2$

$$(\pi_{X_1}(R)) \bowtie (\pi_{X_2}(R)) \supseteq R$$

Due casi estremi

- # Se $X_1 = X_2$
 - l'operazione di join coincide con l'intersezione
- # Se X_1 ed X_2 sono disgiunti
 - l'operazione di join coincide con il prodotto cartesiano

Il theta-join

- # $R_1 \triangleright \triangleleft_F R_2 = \sigma_F (R_1 \triangleright \triangleleft R_2)$
- # La condizione F è spesso una congiunzione (AND) di atomi di confronto $A_1 \vartheta A_2$ dove ϑ è uno degli operatori di confronto ($=, >, <, \dots$)
- # Se l'operatore è sempre l'uguaglianza ($=$) allora si parla di equi-join

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati $\triangleright \triangleleft_{\text{Reparto=Codice}}$ Reparti

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

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
A	Mori
B	Bruni

Impiegati ▷◁ Reparti

Equivalenza di espressioni

- # Due espressioni sono equivalenti se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati
- # L'equivalenza è importante perché i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "costose"
- # Trasformazioni di equivalenza

Algebra con valori nulli

Impiegati

Matricola	Cognome	Filiale	Età
7309	Rossi	Roma	32
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

$\sigma_{\text{Età} > 40}$ (Impiegati)

- la condizione atomica è vera solo per valori non nulli
- introduzione di un terzo valore di verità: unknown (U)

Selezione con valori nulli: soluzione

$\sigma_{Età>30}(\text{Persone}) \cup \sigma_{Età\leq 30}(\text{Persone}) \neq \text{Persone}$

$\sigma_{Età>30 \vee Età\leq 30}(\text{Persone}) \neq \text{Persone}$

- # la condizione atomica è vera solo per valori non nulli
- # per riferirsi ai valori nulli esistono forme apposite di condizioni:

IS NULL
IS NOT NULL

Quindi:

$$\sigma_{\text{Età} > 30}(\text{Persone}) \cup \sigma_{\text{Età} \leq 30}(\text{Persone}) \cup \sigma_{\text{Età IS NULL}}(\text{Persone})$$
$$=$$
$$\sigma_{\text{Età} > 30 \vee \text{Età} \leq 30 \vee \text{Età IS NULL}}(\text{Persone})$$
$$=$$

Persone

Viste (relazioni derivate)

- # Rappresentazioni diverse per gli stessi dati
- # Relazioni derivate:
 - relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- # Relazioni di base: contenuto autonomo
- # Le relazioni derivate possono essere definite su altre derivate, purché esista un ordinamento

Viste materializzate e virtuali

Due tipi di relazioni derivate:

- viste materializzate
- relazioni virtuali (o viste)

Viste materializzate

relazioni derivate memorizzate nella base di dati

- vantaggi:

- immediatamente disponibili per le interrogazioni

- svantaggi:

- ridondanti

- appesantiscono gli aggiornamenti

- sono raramente supportate dai DBMS

Viste virtuali

Relazioni virtuali:

- sono supportate dai DBMS (tutti)
- una interrogazione su una vista viene eseguita "ricalcolando" la funzione

Viste, esempio

Afferenza	Impiegato	Reparto
	Rossi	A
	Neri	B
	Bianchi	B
	Bianchi	B

Direzione	
Reparto	Capo
A	Mori
B	Bruni
B	Bruni

Una vista:

Supervisione = $\pi_{\text{Impiegato, Capo}} (\text{Afferenza} \triangleright \triangleleft \text{Direzione})$

Interrogazioni sulle viste

Sono eseguite sostituendo alla vista la sua definizione:

$\sigma_{\text{Capo}='Leoni'}$ (Supervisione)

viene eseguita come

$\sigma_{\text{Capo}='Leoni'}(\pi_{\text{Impiegato, Capo}}(\text{Afferenza} \triangleright \triangleleft \text{Direzione}))$

Viste, motivazioni

- # Schema esterno: ogni utente vede solo
 - ciò che gli interessa e nel modo in cui gli interessa, senza essere distratto dal resto
 - ciò che e' autorizzato a vedere (autorizzazioni)
- # Strumento di programmazione:
 - si può semplificare la scrittura di interrogazioni: espressioni complesse e sottoespressioni ripetute
- # Utilizzo di programmi esistenti su schemi ristrutturati

Invece:

- # L'utilizzo di viste non influisce sull'efficienza delle interrogazioni

Viste e aggiornamenti, attenzione

Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Verdi	A

Direzione

Reparto	Capo
A	Mori
B	Bruni
C	Bruni

Supervisione

Impiegato	Capo
Rossi	Mori
Neri	Bruni
Verdi	Mori

Inserire, nella vista, il fatto che Lupi ha come capo Bruni

Viste e aggiornamenti

- # "Aggiornare una vista":
 - modificare le relazioni di base in modo che la vista, "ricalcolata" rispecchi l'aggiornamento
- # L'aggiornamento sulle relazioni di base corrispondente a quello specificato sulla vista deve essere univoco
- # In generale però non è univoco!
- # Ben pochi aggiornamenti sono ammissibili sulle viste

Algebra relazionale: limiti

- ✦ Ci sono però interrogazioni interessanti non esprimibili:
 - calcolo di valori derivati: possiamo solo estrarre valori, non calcolarne di nuovi; calcoli di interesse:
 - a livello di ennupla o di singolo valore (conversioni somme, differenze, etc.)
 - su insiemi di ennuple (somme, medie, etc.)
 - interrogazioni inerentemente ricorsive, come la chiusura transitiva

Chiusura transitiva, esempio

Supervisione(Impiegato, Capo)

- ⌘ Per ogni impiegato, trovare tutti i superiori (cioè il capo, il capo del capo, e così via)

Impiegato	Capo
Rossi	Lupi
Neri	Bruni
Lupi	Falchi

Impiegato	Superiore
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Rossi	Falchi

Chiusura transitiva

Nell'esempio, basterebbe il join della relazione con se stessa, previa opportuna ridenominazione

Ma:

Impiegato	Capo
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Falchi	Leoni

Impiegato	Superiore
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Rossi	Falchi
Lupi	Leoni
Rossi	Leoni

Chiusura transitiva, impossibile!

- # Non esiste in algebra relazionale la possibilità di esprimere l'interrogazione che, per ogni relazione binaria, ne calcoli la chiusura transitiva
- # Per ciascuna relazione, è possibile calcolare la chiusura transitiva, ma con un'espressione ogni volta diversa