Algebra e calcolo relazionale (parte 1)

Linguaggi di interrogazione per basi di dati relazionali

- Operazioni su basi di dati:
  - interrogazioni (queries): “legge” dati dal base di dati
  - aggiornamenti (updates): “modifica” il contenuto della base di dati
- Entrambe possono essere modellate come funzioni da basi di dati a basi di dati
- Fondamenti possono essere studiati nel contesto dei linguaggi di interrogazione:
  - algebra relazionale, un linguaggio “procedurale”
  - calcolo relazionale, un linguaggio “dichiarativo”
  - (brevemente) Datalog, un linguaggio di interrogazione più potente
- Poi studieremo SQL, un linguaggio pratico (con caratteristiche dichiarative e procedurali) per esprimere interrogazioni ed aggiornamenti

Algebra relazionale

- Una collezione di operatori
  - definiti su relazioni
  - produce relazioni come risultato

  e quindi possono essere combinate per formare espressioni più complesse
- Operatori
  - unione, intersezione, differenza
  - ridenominazione
  - selezione
  - proiezione
  - join (join naturale, prodotto cartesiano, theta join)

Unione, intersezione, differenza

- Relazioni sono insiemi, quindi possiamo applicare gli operatori tra insiemi
- Tuttavia, vogliamo che il risultato siano relazioni (cioè insiemi omogenei di ennuple)
- Quindi:
  - È significativo applicare l’unione, intersezione e differenza solo a coppie di relazioni definite su gli stessi attributi

Unione

Graduates

Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Managers

Number	Surname	Age
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Graduates ∪ Managers

Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38
9297	O'Malley	56

Intersezione

Graduates

Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Managers

Number	Surname	Age
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Graduates $\cap$ Managers

Number	Surname	Age
7432	O'Malley	39
9824	Darkes	38

Differenza

Graduates

Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Managers

Number	Surname	Age
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Graduates - Managers

Number	Surname	Age
7274	Robinson	37

Una utile ma impossibile unione

Paternity

Father	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael

Maternity

Mother	Child
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

Paternity $\cup$ Maternity **???**

- problema: Father e Mother sono nomi differenti, ma entrambi rappresentato un "Parent"
- soluzione: ridenominare gli attributi

Ridenominazione

- operatore unario
- “modifica i nomi di attributi" senza cambiare i valori
- elimina le limitazioni associate con gli operatori su insiemi
- notazione:
  - $\rho_{Y \leftarrow X.}(r)$
- esempio:
  - $\rho_{\text{Parent} \leftarrow \text{Father}.}(\text{Paternity})$
- se ci sono due o più attributi coinvolti, allora l’ordine è significativo:
  - $\rho_{\text{Location,Pay} \leftarrow \text{Branch,Salary}.}(\text{Employees})$

Ridenominazione, esempio

Paternity

Father	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael

$\rho_{\text{Parent} \leftarrow \text{Father}}$.(Paternity)

Parent	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael

Ridenominazione e unione

Paternity

Father	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael

Maternity

Mother	Child
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

$\rho_{\text{Parent} \leftarrow \text{Father}}$.(Paternity) $\cup$ $\rho_{\text{Parent} \leftarrow \text{Mother}}$.(Maternity)

Parent	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

Ridenominanzione e unione con più attributi

Employees

Surname	Branch	Salary
Patterson	Rome	45
Trumble	London	53

Staff

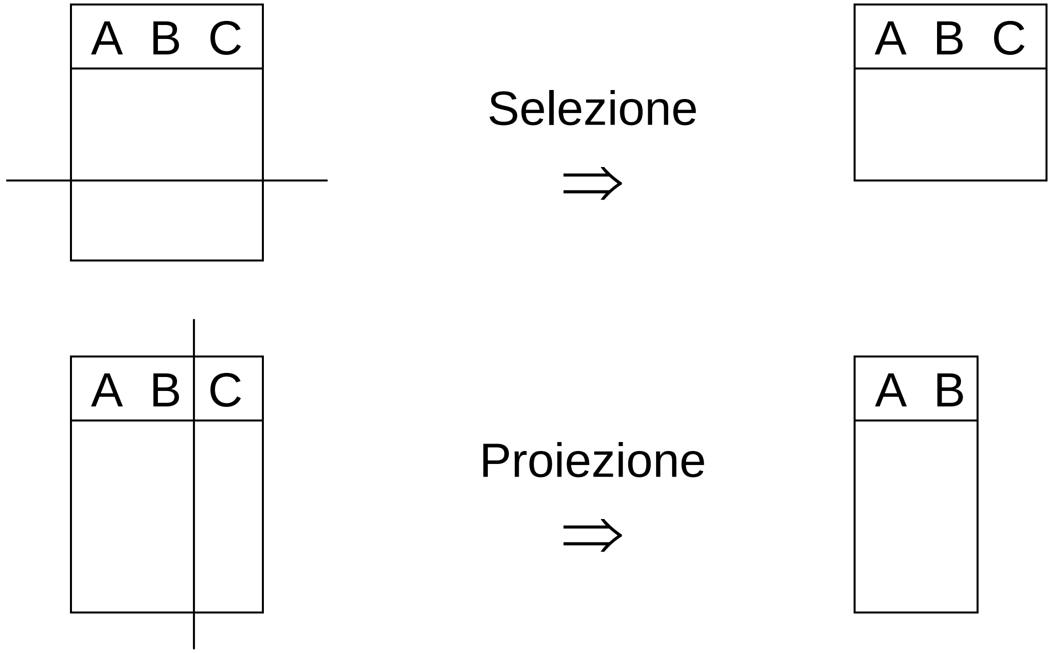
Surname	Factory	Wages
Cooke	Chicago	33
Bush	monza	32

$$\rho_{\text{Location,Pay} \leftarrow \text{Branch,Salary}}(\text{Employees}) \cup \rho_{\text{Location,Pay} \leftarrow \text{Factory, Wages}}(\text{Staff})$$

Surname	Location	Pay
Patterson	Rome	45
Trumble	London	53
Cooke	Chicago	33
Bush	Monza	32

Selezione e proiezione

- Due operatori unari, che sono ortogonali tra di loro:
  - selezione realizza una decomposizione “orizzontale"
  - Proiezione realizza una decomposizione "verticale”



Selezione

- Produce risultati
  - con lo stesso schema dell'operando
  - con un sottoinsieme delle ennuple (che soddisfano una condizione)
- Notazione:
  - $\sigma_F(r)$
- Semantica:
  - $\sigma_F(r) = \{ t \mid t \in r$ e t soddisfa $F\}$

Selezione, esempio

Employees

Surname	FirstName	Age	Salary
Smith	Mary	25	2000
Black	Lucy	40	3000
Verdi	Nico	36	4500
Smith	Mark	40	3900

$\sigma_{\text{Age}<30 \lor \text{Salary}>4000}$ (Employees)

Surname	FirstName	Age	Salary
Smith	Mary	25	2000
Verdi	Nico	36	4500

Selezione, un altro esempio

Citizens

Surname	FirstName	PlaceOfBirth	Residence
Smith	Mary	Rome	Milan
Black	Lucy	Rome	Rome
Verdi	Nico	Florence	Florence
Smith	Mark	Naples	Florence

$\sigma_{\text{PlaceOfBirth=Residence}}$ (Citizens)

Surname	FirstName	PlaceOfBirth	Residence
Black	Lucy	Rome	Rome
Verdi	Nico	Florence	Florence

Proiezione

- Produce risultati
  - su un sottoinsieme degli attributi dell'operando
  - con valori da tutte le sue ennuple
- Notazione (data una relazione r(X) e un sottoinsieme Y di X):
  - $\pi_Y(r)$
- Semantica:
  - $\pi_Y(r) = \{ t[Y] \mid t \in r \}$

Proiezione, esempio

Employees

Surname	FirstName	Department	Head
Smith	Mary	Sales	De Rossi
Black	Lucy	Sales	De Rossi
Verdi	Mary	Personnel	Fox
Smith	Mark	Personnel	Fox

$\pi_{\text{Surname, FirstName}}(\text{Employees})$

Surname	FirstName
Smith	Mary
Black	Lucy
Verdi	Mary
Smith	Mark

Proiezione, un altro esempio

Employees

Surname	FirstName	Department	Head
Smith	Mary	Sales	De Rossi
Black	Lucy	Sales	De Rossi
Verdi	Mary	Personnel	Fox
Smith	Mark	Personnel	Fox

$\pi_{\text{Department, Head}}$ (Employees)

Department	Head
Sales	De Rossi
Personnel	Fox

Cardinalità della proiezione

- Il risultato di una proiezione deve contenere al più tante ennuple quante sono le ennuple dell'operando
- Può contenerne di meno, se diverse ennuple “collassano"
- $\pi_Y(r)$ contiene tante ennuple quante r se e solo se Y è una superchiave per r;
  - Questo vale anche se Y è superchiave “per caso“ (cioè non è stata definita come superchiave nello schema, ma è superchiave per una specifica istanza), vediamo un esempio

Ennuple che collassano

Students

RegNum	Surname	FirstName	BirthDate	DegreeProg
284328	Smith	Luigi	29/04/59	Computing
296328	Smith	John	29/04/59	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	01/05/61	Fine Art
965536	Black	Lucy	05/03/58	Fine Art

$\pi_{\text{Surname, DegreeProg}}$ (Students)

Surname	DegreeProg
Smith	Computing
Smith	Engineering
Black	Fine Art

Ennuple che non collassano “per caso”

Students

RegNum	Surname	FirstName	BirthDate	DegreeProg
296328	Smith	John	29/04/59	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	01/05/61	Fine Art
965536	Black	Lucy	05/03/58	Engineering

$\pi_{\text{Surname, DegreeProg}}$ (Students)

Surname	DegreeProg
Smith	Computing
Smith	Engineering
Black	Fine Art
Black	Engineering

Join

- È il più caratteristico degli operatori dell'algebra relazionale
- Permette di correlare dati contenuti in diverse relazioni, confrontando i valori contenuti in esse ed utilizzando la natura "value-based" del modello relazionale
- Due sono i tipi principali di join:
  - join "naturale" : prende in considerazione i nomi degli attributi
  - "theta" join
- Tutti e due sono denotati con il simbolo $\triangleright\triangleleft$

Il join naturale

r_1

Employee	Department
Smith	sales
Black	production
White	production

r_2

Department	Head
production	Mori
sales	Brown

$r_1 \triangleright\triangleleft r_2$

Employee	Department	Head
Smith	sales	Brown
Black	production	Mori
White	production	Mori

Join naturale: definizione

- $r_1(X_1)$, $r_2(X_2)$
- $r_1 \rhd\lhd r_2$ (join naturale di r_1 ed r_2) è una relazione su X_1X_2 (l'unione dei due insiemi):

$$\{ t \text{ su } X_1X_2 \mid t[X_1] \in r_1 \text{ e } t[X_2] \in r_2 \}$$

o, equivalentemente

$$\{ t \text{ su } X_1X_2 \mid \text{esiste } t_1 \in r_1 \text{ e } t_2 \in r_2 \text{ con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$$

Join naturale: commenti

- Le ennuple nel risultato sono ottenute combinando le ennuple negli operandi con uguali valori sugli attributi comuni
- Gli attributi comuni spesso formano una chiave di uno dei due operandi (ricordiamo: i riferimenti sono realizzati attraverso chiavi, e noi facciamo il join per poter seguire i riferimenti)

Un altro join naturale

Offences

Code	Date	Officer	Dept	Registartion
143256	25/10/1992	567	75	5694 FR
987554	26/10/1992	456	75	5694 FR
987557	26/10/1992	456	75	6544 XY
630876	15/10/1992	456	47	6544 XY
539856	12/10/1992	567	47	6544 XY

Cars

Registration	Dept	Owner	…
6544 XY	75	Cordon Edouard	…
7122 HT	75	Cordon Edouard	…
5694 FR	75	Latour Hortense	…
6544 XY	47	Mimault Bernard	…

Offences ▷◁ Cars

Code	Date	Officer	Dept	Registration	Owner	…
143256	25/10/1992	567	75	5694 FR	Latour Hortense	…
987554	26/10/1992	456	75	5694 FR	Latour Hortense	…
987557	26/10/1992	456	75	6544 XY	Cordon Edouard	…
630876	15/10/1992	456	47	6544 XY	Cordon Edouard	…
539856	12/10/1992	567	47	6544 XY	Cordon Edouard	…

Ancora un altro join

- Confronto con l'unione:
  - Gli stessi dati possono essere combinati in vari modi

Paternity

Father	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael

Maternity

Mother	Child
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

Paternity $\triangleright\triangleleft$ Maternity

Father	Child	Mother
Adam	Cain	Eve
Abraham	Isaac	Sarah
Abraham	Ishmael	Hagar

I join possono essere "incompleti"

- Se una ennupla non ha una "controparte" nell'altra relazione, allora non contribuisce al join ("dangling" ennupla)

r_1

Employee	Department
Smith	sales
Black	production
White	production

r_2

Department	Head
production	Mori
purchasing	Brown

$r_1 \rhd\lhd r_2$

Employee	Department	Head
Black	production	Mori
White	production	Mori

I join possono essere vuoti

- Come caso estremo, possiamo avere che nessuna ennupla ha una controparte, e tutte le ennuple sono dangling

r_1

Employee	Department
Smith	sales
Black	production
White	production

r_2

Department	Head
marketing	Mori
purchasing	Brown

$r_1 \bowtie r_2$

Employee	Department	Head

L'altro estremo

- Se ogni ennupla di un operando può essere combinata con tutte le ennuple dell'altro, allora il join ha una cardinalità che è il prodotto delle cardinalità degli operandi

r_1

Employee	Project
Smith	A
Black	A
White	A

r_2

Project	Head
A	Mori
A	Brown

$r_1 \bowtie r_2$

Employee	Project	Head
Smith	A	Mori
Black	A	Brown
White	A	Mori
Smith	A	Brown
Black	A	Mori
White	A	Brown

Quante tuple ci sono in un join?

Consideriamo $r_1 (X_1)$, $r_2 (X_2)$

- il join ha una cardinalità tra zero ed il prodotto delle cardinalità degli operandi:

  $$0 \leq | r_1 \bowtie r_2 | \leq | r_1 | \times | r_2 |$$

  ($| r |$ è cardinalità della relazione r)
- inoltre:
  - se il join è completo, allora la sua cardinalità è almeno il massimo tra $| r_1 |$ e $| r_2 |$
  - se $X_1 \cap X_2$ contiene una chiave per r_2, allora $| r_1 \bowtie r_2 | \leq | r_1 |$
  - se $X_1 \cap X_2$ è la chiave primaria per r_2 e c'è un vincolo di integrità referenziale tra $X_1 \cap X_2$ in r_1 e tale chiave, allora $| r_1 \bowtie r_2 | = | r_1 |$

Join esterno

- una variante del join, per mantenere tutti le informazioni contenute negli operandi
- “riempe con nulls" le ennuple che non hanno una controparte
- Ci sono tre varianti:
  - “sinistra": solo ennuple del primo operando sono riempite con nulls
  - “destra": solo ennuple del secondo operando sono riempite con nulls
  - "full": ennuple di tutti e due gli operando sono riempite con nulls

Join esterni

r_1

Employee	Department
Smith	sales
Black	production
White	production

r_2

Department	Head
production	Mori
purchasing	Brown

$r_1 \triangleright\triangleleft_{\text{LEFT}} r_2$

Employee	Department	Head
Smith	Sales	NULL
Black	production	Mori
White	production	Mori

$r_1 \triangleright\triangleleft_{\text{RIGHT}} r_2$

Employee	Department	Head
Black	production	Mori
White	production	Mori
NULL	purchasing	Brown

$r_1 \triangleright\triangleleft_{\text{FULL}} r_2$

Employee	Department	Head
Smith	Sales	NULL
Black	production	Mori
White	production	Mori
NULL	purchasing	Brown

Join ennario

- Il join naturale è
  - commutativo: $r_1 \bowtie r_2 = r_2 \bowtie r_1$
  - associativo: $(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$
- Quindi possiamo scrivere join ennari senza ambiguità:

$$r_1 \bowtie r_2 \bowtie \ldots \bowtie r_n$$

Join ennario, esempio

r_1

Employee	Department
Smith	sales
Black	production
Brown	marketing
White	production

r_2

Department	Division
production	A
marketing	B
purchasing	B

r_3

Division	Head
A	Mori
B	Brown

$r_1 \rhd\lhd r_2 \rhd\lhd r_3$

Employee	Department	Division	Head
Black	production	A	Mori
Brown	marketing	B	Brown
White	production	A	Mori

Prodotto cartesiano

- Il join naturale è definito anche quando gli operandi non hanno attributi in comune
- In questo caso nessuna condizione è imposta sulle ennuple, e quindi il risultato contiene ennuple ottenute combinando le ennuple degli operandi in tutti I modi possibili

Prodotto cartesiano: esempio

Employees

Employee	Project
Smith	A
Black	A
Black	B

Projects

Code	Name
A	Venus
B	Mars

Employes $\rhd\lhd$ Projects

Employee	Project	Code	Name
Smith	A	A	Venus
Black	A	A	Venus
Black	B	A	Venus
Smith	A	B	Mars
Black	A	B	Mars
Black	B	B	Mars

Theta-join

- In molti casi, il prodotto cartesiano è significativo solo se seguito da una selezione:
  - a tale scopo è stato introdotto il **theta-join** che è un operatore *derivato*

$$r_1 \triangleright\triangleleft_F r_2 = \sigma_F(r_1 \triangleright\triangleleft r_2)$$

  - se F è una congiunzione di uguaglianze, allora abbiamo un **equi-join**

Equi-join: esempio

Employees

Employee	Project
Smith	A
Black	A
Black	B

Projects

Code	Name
A	Venus
B	Mars

Employes $\triangleright\triangleleft_{\text{Project=Code}}$ Projects

Employee	Project	Code	Name
Smith	A	A	Venus
Black	A	A	Venus
Black	B	B	Mars

Interrogazioni

- Una interrogazione è una funzione da instanze di basi di dati a relazioni
- Le interrogazioni sono definite usando l'algebra relazionale mediante espressioni su relazioni

Una base di dati per gli esempi

Employees

Number	Name	Age	Salary
101	Mary Smith	34	40
103	Mary Bianchi	23	35
104	Luigi Neri	38	61
105	Nico Bini	44	38
210	Marco Celli	49	60
231	Siro Bisi	50	60
252	Nico Bini	44	70
301	Steve Smith	34	70
375	Mary Smith	50	65

Supervision

Head	Employee
210	101
210	103
210	104
231	105
301	210
301	231
375	252

Esempio 1

- Trova il numero, nome ed eta' degli impiegati che guadagnano più di 40 mila.

$$\pi_{Number,Name,Age}(\sigma_{Salary>40}(\text{EMPLOYEES})) \tag{3.1}$$

Number	Name	Age
104	Luigi Neri	38
210	Marco Celli	49
231	Siro Bisi	50
252	Nico Bini	44
301	Steve Smith	34
375	Mary Smith	50

Esempio 2

- Trova le matricole dei capi degli impiegati che guadagnano più di 40 mila

$$\pi_{\text{Head}}(\text{SUPERVISION} \bowtie_{\text{Employee=Number}}(\sigma_{\text{Salary}>40}(\text{EMPLOYEES}))) \qquad (3.2)$$

Head
210
301
375

Esempio 3

- Trova nome e stipendio dei capi degli impiegati che guadagnano più di 40 mila

$$\pi_{\text{NameH,SalaryH}}\,(\rho_{\text{NumberH,NameH,SalaryH,AgeH}\leftarrow\text{Number,Name,Salary,Age}}\,(\text{EMPLOYEES}) \bowtie_{\text{NumberH=Head}} (\text{SUPERVISION} \bowtie_{\text{Employee=Number}} (\sigma_{\text{Salary}>40}(\text{EMPLOYEES})))) \qquad (3.3)$$

NameH	SalaryH
Marco Celli	60
Steve Smith	70
Mary Smith	65

Esempio 4

- Trova gli impiegati che guadagnano più del rispettivo capo, mostrando matricola, nome e stipendio di ciascuno di essi e del capo

$$\begin{gathered}\pi_{\text{Number,Name,Salary,NumberH,NameH,SalaryH}}\\
(\sigma_{\text{Salary}>\text{SalaryH}}(\rho_{\text{NumberH,NameH,SalaryH,AgeH}\leftarrow\text{Number,Name,Salary,Age}}(\text{EMPLOYEES})\\
\bowtie_{\text{NumberH=Head}}(\text{SUPERVISION} \bowtie_{\text{Employee=Number}}(\text{EMPLOYEES})))) \qquad (3.4)\end{gathered}$$

Number	Name	Salary	NumberH	NameH	SalaryH
104	Luigi Neri	61	210	Marco Celli	60
252	Nico Bini	70	375	Mary Smith	65

Esempio 5

- Trova matricola e nome dei capi i cui impiegati guadagnano tutti più di 40 mila
- Applica la doppia negazione: trova i capi per i quali non vi sia alcun impiegato con stipendio non superiore a 40 mila
- Prendiamo tutti i capi meno quelli che hanno un impiegato che guadagna non più di 40 mila

$$\pi_{\text{Number,Name}}(\text{EMPLOYEES} \bowtie_{\text{Number=Head}} (\pi_{\text{Head}}(\text{SUPERVISION}) - \pi_{\text{Head}}(\text{SUPERVISION} \bowtie_{\text{Employee=Number}} (\sigma_{\text{Salary} \leq 40}(\text{EMPLOYEES}))))) \quad (3.5)$$

Number	Name
301	Steve Smith
375	Mary Smith

Equivalenza

- L'algebra relazionale, in modo analogo ad altri strumenti formali consente di formulare espressioni tra loro *equivalenti*
- Nel contesto dei numeri reali e agli operatori di addizione e moltiplicazione, vale l'equivalenza

$$x \times (y + z) \equiv x \times y + x \times z$$

- Nell'algebra relazionale possiamo dare una definizione analoga facendo attenzione al fatto che l'equivalenza può essere assoluta, oppure dipendere dallo schema
  - $E_1 \equiv_{\mathbf{R}} E_2$ se $E_1(\mathbf{r}) = E_2(\mathbf{r})$**,** per ogni istanza **r** di **R**
  - $E_1 \equiv E_2$ se $E_1 \equiv_{\mathbf{R}} E_2$ per ogni schema **R**

Tipi di equivalenze

- L'equivalenza *relativa allo schema* è: $E_1 \equiv_R E_2$
- L'equivalenza *assoluta* è: $E_1 \equiv E_2$

La distinzione è importante poiché gli schemi degli operandi non vengono esplicitati nelle espressioni ed il comportamento può variare a secondo degli attributi negli schemi

Es. eq assoluta (sempre valida)

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

Es. eq relativa (valida se e solo se in **R** l'intersezione fra gli attributi di R_1 e R_2 è pari ad A)

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

Motivazioni

- L’equivalenze è importante dal punto di vista applicativo, nella fase di esecuzione delle interrogazioni per motivi di ottimizzazione.
- Data una certa interrogazione, questa viene trasformata in una equivalente, ma che richiede meno tempo di esecuzione
- In particolare sono interessanti le trasformazioni che riducono le dimensioni dei risultati intermedi e quelle che preparano all’applicazione di una trasformazione che riduce le dimensioni del risultato intermedio

Trasformazioni

1. Atomizzazione delle selezioni

$$\sigma_{F1 \wedge F2}(E) \equiv \sigma_{F1}(\sigma_{F2}(E))$$

2. Idempotenza della proiezione

$$\pi_{X}(E) \equiv \pi_{X}(\pi_{XY}(E))$$

3. Anticipazione della selezione rispetto al join

$$\sigma_{F}(E_1 \bowtie E_2) \equiv E_1 \bowtie \sigma_{F}(E_2)$$

se la condizione F è riferita solo ad attributi della sottoespressione E_2

Trasformazioni (2)

4. Anticipazione della proiezione rispetto al join

$$\pi_{X1Y2}(E_1 \bowtie E_2) \equiv E_1 \bowtie \pi_{Y2}(E_2)$$

siano E_1 ed E_2 definite su X1 e X2; se $Y2 \subseteq X2$ e $Y2 \supseteq X1 \cap X2$ (cioè gli attributi in X2 – Y2 non sono coinvolti nel join) allora vale l'equivalenza

5. Inglobamento di una selezione in un prodotto cartesiano a formare un join

$$\sigma_F(E_1 \bowtie E_2) \equiv E_1 \bowtie_F E_2$$

Trasformazioni (3)

6. Distributività della selezione rispetto all'unione

$$\sigma_F (E_1 \cup E_2) \equiv \sigma_F (E_1) \cup \sigma_F (E_2)$$

7. Distributività della selezione rispetto differenza

$$\sigma_F (E_1 - E_2) \equiv \sigma_F (E_1) - \sigma_F (E_2)$$

8. Distributività della proiezione rispetto all'unione

$$\pi_X (E_1 \cup E_2) \equiv \pi_X (E_1) \cup \pi_X (E_2)$$

Si noti che la proiezione non è distributiva rispetto alla differenza

Trasformazioni (4)

Altre trasformazioni si basano sulla corrispondenza fra gli operatori insiemistici e le selezioni complesse

9. $\sigma_{F1 \vee F2}(R) \equiv \sigma_{F1}(R) \cup \sigma_{F2}(R)$

10. $\sigma_{F1 \wedge F2}(R) \equiv \sigma_{F1}(R) \cap \sigma_{F2}(R) \equiv \sigma_{F1}(R) \triangleright\triangleleft \sigma_{F2}(R)$

11. $\sigma_{F1 \wedge \neg(F2)}(R) \equiv \sigma_{F1}(R) - \sigma_{F2}(R)$

C'è anche la proprietà commutativa e associativa di tutti gli operatori binari esclusa la differenza e la proprietà distributiva del join rispetto all'unione

12. $E \triangleright\triangleleft (E_1 \cup E_2) \equiv E \triangleright\triangleleft E_1 \cup (E \triangleright\triangleleft E_2)$

Esempio

- Trova i numeri di matricola dei capi di impiegati con meno di 30 anni.

$\pi_{Capo} (\sigma_{Matr=Imp \wedge Età<30} (Impiegati \bowtie Supervisione))$

Regola 1, spezziamo la selezione

$\pi_{Capo} (\sigma_{Matr=Imp} (\sigma_{Età<30} (Impiegati \bowtie Supervisione)))$

Regola 5 per selezione esterna e Regola 3 per altra selezione

$\pi_{Capo} (\sigma_{Età<30} (Impiegati) \bowtie_{Matr=Imp} Supervisione)$

Regola 4 elimina attributi inutili al primo membro

$\pi_{Capo} (\pi_{Matr} (\sigma_{Età<30} (Impiegati)) \bowtie_{Matr=Imp} Supervisione)$

Algebra con i valori nulli

People

Name	Age	Salary
Aldo	35	15
Andrea	27	21
Maria	NULL	42

- $\sigma_{\text{Age}>30}$ (People)
- quali ennuple appartengono al risultato?
- la prima si, la seconda no e la terza?