

Ripasso

- Il concetto di tabella viene mappato formalmente nella teoria degli insiemi con una struttura non posizionale.
- Ogni n-upla (senza duplicati) è composta da più attributi e ogni attributo ha un dominio.
- Fra tutti gli attributi presenti in una relazione, un sottoinsieme che permette di identificare univocamente i record si dice superchiave e la superchiave minimale (minor numero di attributi) si chiama chiave.
- Ne viene scelta una fra le chiavi possibili e quella sarà chiave primaria. E' quello strumento che permette di collegare record fra una relazione e un'altra (fra tabelle)
- La chiave esterna è un insieme di attributi che fanno riferimento alla chiave primaria di un'altra tabella
- Il concetto di chiavi esterne si può paragonare ai puntatori in C++ per certi versi.
- Per definire un modello di dati: serve strutture dati e anche un **set di operazioni** che permettono di rispondere a delle domande ben precise, cioè per **fare query sui dati**.

Vedremo le operazioni di **base** tra tutte

ALGEBRA RELAZIONALE

L'algebra relazionale comprende un set di operatori che:

- sono **definiti sulle relazioni**;
- producono come **risultato una relazione**.
- ovvero sa, in un certo senso: $f : Relazione \rightarrow Relazione$

Gli operatori possono essere combinati per formare espressioni complesse.

Per esempio: l'operatore "join" è derivato e si ottiene da operazioni più semplici

Operatori primitivi

Le relazioni sono degli insiemi, quindi possiamo applicare gli operatori sugli insiemi.

Il risultato deve essere un set omogeneo di n-uple, quindi, applichiamo gli **operatori sui set solo fra relazioni con gli stessi attributi**. (Vuol dire che R ed S devono avere la stessa struttura, cioè devono avere le stesse colonne (non in numero)).

Devo poter *cercare* fra le righe (selezione) e *prendere una colonna* (proiezione).
Dovrei anche poter cambiare nome della colonna (ridenominazione).

Gli operatori primitivi sono:

- **ridenominazione** δ : cambia il nome di una colonna;
- **unione** \cup : che corrisponde all'unione insiemistica;
- **selezione** σ , prende in considerazione una riga;
- **differenza** $R \setminus S$ insiemistica;
- **proiezione** π , prende in considerazione una colonna da una relazione;
- **restrizione**, o selezione: corrisponde alla ricerca fra le righe;
- **prodotto cartesiano**: che corrisponde a quello insiemistico.

In simboli si ha:

- $R, S, T \dots$ sono **relazioni** (quasi le ultime lettere maiuscole);
- A, B, C sono i **singoli attributi** (prime lettere dell'alfabeto);
 - X, Y, Z sono **insiemi di attributi** (ultime lettere dell'alfabeto).

Ridenominazione δ

Questa operazione primitiva si indica con
 $\delta_{Nome_originale} \rightarrow Nuovo_nome(Tabella)$

Esempio:

STUDENTE

Corso	Matricola	Voto
Programmazione	123456	27
EINN	23456	28

$\delta_{Matricola \rightarrow Codice\ Studente}(STUDENTE)$

..e si ottiene:

Corso	Codice Studente	Voto
Programmazione	132456	27
EINN	23456	28

Unione

Essa si rappresenta $R \cup S$ = prendo le righe di R, prendo le righe di S e le unisco in una nuova tabella. Se ho 2 righe uguali, nell'unione sarà presente una volta.

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 \cup Managers

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

Esempio non fattibile perchè hanno colonne diverse ↓

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 ???

- “Father” e “Mother” sono attributi con nomi diversi ma entrambi sono “Genitori”
- Soluzione: ridenominare gli attributi

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

$$\delta_{Father \rightarrow Parent}(Paternity) \cup \delta_{Mother \rightarrow Parent}(Maternity)$$

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

Proiezione π

La proiezione prende in considerazione delle colonne.

• Definizione

- Sia R una relazione e siano A_1, A_2, \dots, A_n alcuni suoi attributi allora:

$$\pi_{A_1, A_2, \dots, A_n}(R) = \{t[A_1, A_2, \dots, A_n] \mid t \in R\}$$

La **cardinalità** sarà: numero di righe \leq numero di righe di partenza perchè **i duplicati non ci sono**.

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

Quindi si vanno a guardare le colonne

Esempio senza duplicati:

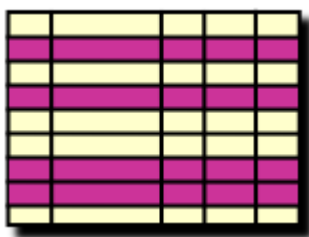
Employees

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

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

Department	Head
Sales	De Rossi
Personnel	Fox

Selezione σ



• Definizione

- Sia R una relazione allora

$$\sigma_{\lambda}(R) = \{t | t \in R \wedge \lambda(t) = \text{TRUE}\}$$

Questa operazione prende tutte le righe della relazione R per cui il predicato λ è soddisfatto

λ deve essere una **formula proposizionale**, una congiunzione di condizioni valutate vere o false collegate da AND OR NOT logici.

Esempio:

Employees

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

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

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

Citizens

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

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

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

Prodotto cartesiano

$$R \times S = \{tu | t \in R \wedge u \in S\}$$

La nuova relazione ha tutti gli attributi della relazione R e tutti gli attributi della relazione S.

Le due relazioni non devono avere attributi dello stesso nome quindi:

- Siano $R(A_1: T_1, \dots, A_n: T_n)$ e $S(A_{n+1}: T_{n+1}, \dots, A_{n+m}: T_{n+m})$ due relazioni con $\{A_1, \dots, A_n\} \cap \{A_{n+1}, \dots, A_{n+m}\} = \emptyset$.

Employees	Employee	Project
	Smith	A
	Black	A
	Black	B

Projects	Code	Name
	A	Venus
	B	Mars

Employees X 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

Il risultato del prodotto cartesiano non necessariamente producono qualcosa di "in comune" ma se abbinata ad un'altra operazione, la si potrebbe ottenere.

Operatori derivati

Gli operatori derivati sono utili e si possono esprimere in funzione di quelli primitivi.

Intersezione

- Siano R ed S dello stesso tipo

$$R \cap S = \{t | t \in R \wedge t \in S\}$$

- Essa si può esprimere in funzione degli operatori primitivi:

$$R \cap S = R - (R - S)$$

IL "-"(MENO) E' ERRATO SOPRA. SI USA, INVECE, QUELLO INSIEMISTICO "\"

JOIN ⋈

Questa operazione combina t-uple **basandosi sui valori degli attributi da relazioni diverse**.

- Fondamentalmente due tipi (più qualche variante):
 - Natural JOIN
 - Theta JOIN

- **Natural join** \bowtie --> prese 2 tabelle, le unisce sulla base degli attributi che presentano lo stesso nome.
- **Theta join** \bowtie_F --> richiede una condizione che specifichiamo noi, quindi è una natural join con restrizione.

NATURAL JOIN

- Siano R con attributi XY ed S con attributi YZ
- $R \bowtie S$ è una relazione di attributi XYZ costituita da tutte le n -uple t tali che: $t[XY] \in R$, $t[YZ] \in S$.

$$R \bowtie S = \{t | t[XY] \in R \text{ e } t[YZ] \in S\}$$

- La giunzione è derivata perché
 - Si rinominano gli attributi Y in S come Y' e si ottiene S' . Si opera la giunzione (equijoin) rispetto ad Y ed Y' . Si proietta rispetto a XYZ

$$R \bowtie S = \pi_{XYZ}(R \bowtie_{Y=Y'} S')$$

EQUI JOIN

- Siano $R(A_1: T_1, \dots, A_n: T_n)$ ed $S(A_{n+1}: T_{n+1}, \dots, A_{n+m}: T_{n+m})$ con $\{A_1, \dots, A_n\} \cap \{A_{n+1}, \dots, A_{n+m}\} = \emptyset$.
- Allora si pone
 - $R \bowtie_{A_i=A_k} S = \{tu | t \in R, u \in S, t.A_i = u.A_k\}$
 - Con $1 \leq i \leq n$ e $n+1 \leq k \leq n+m$.
- La giunzione è derivata perché

$$R \bowtie_{A_i=A_k} S = \sigma_{A_i=A_k}(R \times S)$$

THETA JOIN

- Estensione del NATURAL JOIN
- Viene specificato un predicato per la selezione delle n-uple
- E' un operatore derivato:

$$R \bowtie_F S = \sigma_F (R \times S)$$

- Quando F è una congiunzione di uguaglianze si parla di **equi-JOIN**

Esempio:

r ₁	Employee	Department
	Smith	sales
	Black	production
	White	production

r ₂	Department	Head
	production	Mori
	sales	Brown

le colonne in comune vengono riportate una volta

SI OTTERRA'

r₁ ⋈ r₂

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

Formalmente si ha:

- Sia R con attributi XY ed S con attributi YZ
- $R \bowtie S$ e' una relazione di attributi XYZ costituita da tutte le n-uple t tali che $t[XY] \in R$ e $t[YZ] \in S$
- Quindi:

$$R \bowtie S = \{t | t[XY] \in R \text{ e } t[YZ] \in S\}$$

- Cioè: le n-uple del risultato sono ottenute combinando le n-uple di R e S che hanno gli stessi valori negli attributi con lo stesso nome

Prendo una riga in R e vado a cercare il corrispettivo in S. Se non ci sono attributi in comune allora non sarà nel risultato finale.

Un altro esempio:

Offences	Code	Date	Officer	Dept	Registration
	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	Mimault Bernard	...
539856	12/10/1992	567	47	6544 XY	Mimault Bernard	...

Altro esempio di Natural join:

Paternity	Father	Child
	Adam	Cain
	Adam	Abel
	Abraham	Isaac
	Abraham	Ishmael

Maternity	Mother	Child
	Eve	Cain
	Eve	Seth
	Sarah	Isaac
	Hagar	Ishmael

Paternity ⋈ Maternity

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

L'operazione collega una tabella con un'altra sulla base degli attributi in comune.

Theta join : estensione della natural join dove viene specificato un predicato per l'unione, quindi siamo noi a dire come unire le due tabelle.

Esempio di equi-JOIN

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

Employees $\bowtie_{\text{Project=Code}}$ Projects

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

Query

- La Query è una funzione che prende un database e restituisce una tabella, quindi relazione.
- Serve per **interrogare un database**.

Database di esercitazione

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

- Trovare numero, nome ed età di tutti gli impiegati che guadagnano piu' di 40 mila euro

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

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

- Trovare il numero dei responsabili degli impiegati che guadagnano piu' di 40 mila euro

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

Head
210
301
375

In una join, quando si collegano le tabelle, gli attributi non possono avere gli stessi nomi, pertanto si devono ridenominare perchè il prodotto cartesiano è alla base della join e 2 attributi con lo stesso nome non possono dare risultato valido dopo aver applicato il prodotto cartesiano

- **Trovare nome e salario dei responsabili degli impiegati che guadagnano piu' di 40 mila euro.**

$$\pi_{\text{NameH, SalaryH}} (\rho_{\text{NumberH, NameH, SalaryH, AgeH} \leftarrow \text{Number, Name, Salary, Age}} (\text{EMPLOYEES})$$

$$\bowtie_{\text{NumberH=Head}} (\text{SUPERVISION} \bowtie_{\text{Employee=Number}} (\sigma_{\text{Salary}>40} (\text{EMPLOYEES})))) \quad (3.3)$$

NameH	SalaryH
Marco Celli	60
Steve Smith	70
Mary Smith	65

Potremmo andare avanti per un bel po' ma così facendo le espressioni diventerebbero davvero lunghe.

- Per "riconoscere" attributi con lo stesso nome gli premettiamo il nome della relazione
- Usiamo "assegnazioni" (**viste**) per ridenominare le relazioni (e gli attributi solo quando serve per l'unione).

Allora si **creano relazioni virtuali**, per esempio:

Esempio: Trovare gli impiegati che guadagnano più del proprio capo, mostrando matricola, nome e stipendio dell'impiegato e del capo

Capi := Imp

$$\pi_{\text{Imp.Matr, Imp.Nome, Imp.Stip, Capi.Matr, Capi.Nome, Capi.Stip}} ($$

$$\sigma_{\text{Imp.Stip} > \text{Capi.Stip}} ($$

$$\text{Capi} \bowtie_{\text{Capi.Matr} = \text{Capo}} (\text{Sup} \bowtie_{\text{Imp} = \text{Imp.Matr}} \text{Imp})$$

$$)$$

$$)$$

per **accedere agli attributi di una tabella** posso usare
`nomeTabella.nomeAttributo`