
Parte III

L'algebra relazionale

Linguaggi di interrogazione

- **Dichiarativi**: specificano le proprietà del risultato ("*che cosa*")
- **Procedurali**: specificano le modalità di generazione del risultato ("*come*")
- Rappresentanti più significativi:
 - **Algebra relazionale**: procedurale (usata dai DBMS)
 - **Calcolo relazionale**: dichiarativo (teorico)
 - **SQL** (Structured Query Language): parzialmente dichiarativo (reale)
 - **QBE** (Query by Example): dichiarativo (reale)

Algebra relazionale

- Costituita da un insieme di operatori
 - definiti su relazioni
 - producono relazioni
 - possono essere composti
- Operatori dell'algebra relazionale:
 - **Unione, Intersezione, Differenza** (*operatori insiemistici*)
 - **Ridenominazione**
 - **Selezione**
 - **Proiezione**
 - **Join** (join naturale, prodotto cartesiano, theta-join)

Operatori insiemistici

- Le relazioni sono insiemi (di tuple), quindi ad esse possono essere applicati, in linea di principio, gli operatori insiemistici
- Esistono comunque delle limitazioni:
 - Il risultato deve essere anch'esso una relazione
 - Una relazione non è un insieme qualsiasi
- Possibile applicare **Unione, Intersezione, Differenza** solo a relazioni definite sugli stessi attributi

Unione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cup Quadri

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45
9297	Neri	33

Intersezione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cap Quadri

Matricola	Nome	Età
7432	Neri	54
9824	Verdi	45

Differenza

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati - Quadri

Matricola	Nome	Età
7274	Rossi	42

Un'unione sensata ma impossibile

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Paternità \cup Maternità

??

Ridenominazione

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Ascendenza-P

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

$$\text{Ascendenza-P} = \text{REN}_{\text{Genitore} \leftarrow \text{Padre}} (\text{Paternità})$$

L'operatore **REN** consente la ridenominazione degli attributi, per rendere possibili successive operazioni

Passo 1: ridenominazione

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco



Ascendenza-P

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco



Ascendenza-M

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

$$\text{Ascendenza-P} = \text{REN}_{\text{Genitore} \leftarrow \text{Padre}} (\text{Paternità})$$

$$\text{Ascendenza-M} = \text{REN}_{\text{Genitore} \leftarrow \text{Madre}} (\text{Maternità})$$

Passo 2: unione

Ascendenza-P

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Ascendenza-M

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Ascendenza

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

$$\text{Ascendenza} = \text{Ascendenza-P} \cup \text{Ascendenza-M}$$

Selezione

- Operatore monadico
- Produce come risultato una relazione che
 - Ha lo stesso schema dell'operando
 - Contiene un sottoinsieme delle n-uple dell'operando
 - Sono conservate solo le tuple che soddisfano una data condizione
- Permette di valutare tutte le interrogazioni che implicano condizioni fra attributi della stessa tupla

Selezione: esempi

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Neri	Napoli	64

- Trovare:
 1. Gli impiegati che guadagnano più di 50
 2. Gli impiegati che guadagnano più di 50 e lavorano a Milano
 3. Gli impiegati che hanno lo stesso nome della filiale presso cui lavorano

Selezione: sintassi e semantica

• Sintassi

SEL *Condizione* (*Operando*)

- *Operando*: una qualsiasi relazione
- *Condizione*: espressione booleana definita sugli attributi dell'operando (come quelle dei vincoli di tupla)

• Semantica

- il risultato contiene le n-uple dell'operando che soddisfano la condizione

Selezione: esempio 1

Gli impiegati che guadagnano più di 50

Imp-50 = SEL_{Stipendio > 50} (**Impiegati**)

Imp-50

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
5698	Neri	Napoli	64

Selezione: esempio 2

Gli impiegati che guadagnano più di 50 e lavorano a Milano

Imp-50-M = SEL_{(Stipendio > 50) AND (Filiale='Milano')} (**Impiegati**)

Imp-50-M

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

Selezione: esempio 3

Gli impiegati che hanno lo stesso nome della filiale presso cui lavorano

Imp-NF = SEL_{Cognome=Filiale} (Impiegati)

Imp-NF

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

Selezione e proiezione

Sono due operatori "ortogonali"

- **Selezione:** decomposizione orizzontale
- **Proiezione:** decomposizione verticale



Proiezione: sintassi e semantica

• Sintassi

PROJ_{ListaAttributi} (Operando)

- *Operando*: una qualsiasi relazione
- *ListaAttributi*: un sottoinsieme degli attributi della relazione operando

• Semantica

Il risultato contiene le ennuple ottenute da tutte le ennuple dell'operando *ristrette agli attributi nella lista*

Proiezione: esempio

Matricola e Cognome di tutti gli impiegati

Impiegati-MC

Matricola	Cognome
7309	Neri
5998	Neri
9553	Rossi
5698	Rossi

Impiegati-MC = PROJ_{Matricola, Cognome} (Impiegati)

Proiezione: esempio 2

Cognome e Filiale di tutti gli impiegati

Impiegati-CF

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma
Rossi	Roma

Impiegati-CF = PROJ_{Cognome, Filiale} (Impiegati)

Cardinalità delle proiezioni

- Una proiezione
 - contiene al più tante n-uple quante l'operando
 - può contenerne di meno (vengono eliminati i duplicati)
- Se X è una superchiave di R , allora $PROJ_X(R)$ contiene esattamente tante ennuple quante R
- Se X è una superchiave di R , allora tutte le n-ple assumono valori diversi su X

Selezione e proiezione

Combinando selezione e proiezione, possiamo estrarre interessanti informazioni da una relazione

Matricola e cognome degli impiegati che guadagnano più di 50

Matricola	Cognome
7309	Rossi
5998	Neri
5698	Neri

PROJ_{Matricola, Cognome} (SEL_{Stipendio > 50} (Impiegati))

Join

- Combinando selezione e proiezione, possiamo estrarre informazioni da **un'unica** relazione
- Non riusciamo però correlare informazioni presenti in relazioni diverse
- Il **JOIN** è l'operatore più interessante dell'algebra relazionale
- Permette di correlare dati in relazioni diverse
- Utilizza i collegamenti sui valori che sono alla base del modello relazionale

Join: esempio

- In un concorso pubblico i compiti sono anonimi e ad ognuno è associata una busta chiusa con il nome del candidato
- Ciascun compito e la relativa busta vengono contrassegnati con uno stesso numero

1	Mario Rossi
2	Nicola Russo
3	Mario Bianchi
4	Remo Neri

1	25
2	13
3	27
4	28

Mario Rossi	25
Nicola Russo	13
Mario Bianchi	27
Remo Neri	28

Join: esempio (continua)

Votazione

Numero	Voto
1	25
2	13
3	27
4	28

Candidati

Numero	Candidato
1	Mario Rossi
2	Nicola Russo
3	Mario Bianchi
4	Remo Neri

Risultati

Numero	Candidato	Voto
1	Mario Rossi	25
2	Nicola Russo	13
3	Mario Bianchi	27
4	Remo Neri	28

Join naturale

- Operatore binario (generalizzabile)
- Produce un risultato
 - Definito sull'unione degli attributi degli operandi
 - Con n-uple costruite ciascuna a partire da una n-upla di ognuno degli operandi
- Formalmente:
 - Date due relazioni $R_1(X_1)$, $R_2(X_2)$
 - $R_1 \text{ JOIN } R_2$ è una relazione su $X_1 \cup X_2$
 - $\{ t \text{ su } X_1 \cup X_2 \mid t_1 \in R_1 \text{ e } t_2 \in R_2, t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$

Join completo

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Reparto	Capo
A	Mori
B	Bruni

Impiegati-Capi

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

In un *join completo* ogni tupla contribuisce al risultato

Join non completo

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Reparto	Capo
B	Mori
C	Bruni

Impiegati-Capi

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Alcune tuple non contribuiscono al risultato

Join vuoto

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Reparto	Capo
D	Mori
C	Bruni

Impiegati-Capi

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

Se i valori assunti dagli attributi di join non coincidono in nessuna coppia di tuple, il risultato è vuoto

Join completo con n x m n-ple

Impiegati

Impiegato	Reparto
Rossi	B
Neri	B

Capi

Reparto	Capo
B	Mori
B	Bruni

Impiegati-Capi

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

Ciascuna coppia di tuple contribuisce: la cardinalità è massima

Cardinalità del join

- $R_1(A,B)$, $R_2(B,C)$
- Il join di R_1 e R_2 contiene un numero di tuple compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$:

$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1| \times |R_2|$$

- Se il join coinvolge una chiave di R_2 , allora il numero di tuple è compreso fra zero e $|R_1|$:

$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1|$$

- Se il join coinvolge una chiave di R_2 ed esiste un vincolo di integrità referenziale fra B (in R_1) e R_2 , allora il numero di ennuple è pari a $|R_1|$:

$$|R_1 \text{ JOIN } R_2| = |R_1|$$

Join: tuple che non partecipano

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Reparto	Capo
B	Mori
C	Bruni

Impiegati-Capi

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Alcune tuple non partecipano al risultato

Join esterno (*outer join*)

- Il *join esterno* estende, con valori nulli, le ennuple che verrebbero tagliate fuori da un join (interno)
- Esiste in tre versioni:
 - *sinistro*: mantiene tutte le ennuple del primo operando, estendendole con valori nulli, se necessario
 - *destra*: ... del secondo operando ...
 - *completo*: ... di entrambi gli operandi ...

Join esterno sinistro

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{LEFT} Capi

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

Join esterno destro

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{RIGHT} Capi

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

Join esterno completo

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{FULL} Capi

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

Prodotto cartesiano e theta-join

- È un join naturale su relazioni *senza attributi in comune*
- Contiene sempre un numero di ennuple pari al prodotto delle cardinalità degli operandi (*tutte le ennuple sono combinabili*)
- Ha senso (quasi) solo se seguito da selezione:

$SEL_{\text{Condizione}} (R_1 \text{ JOIN } R_2)$

- L'operazione complessiva viene chiamata **theta-join** ed è indicata con:

$R_1 \text{ JOIN}_{\text{Condizione}} R_2$

Theta-join

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Capi

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN_{Reparto=Codice} Capi

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

Esempio: database degli impiegati

Impiegati

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Supervisione

Impiegato	Capo
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

Esempio 1

Query 1: *Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni.*

SEL_{Stipendio>40}(Impiegati)

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Esempio 2

Query 2: *Trovare matricola, nome ed età degli impiegati che guadagnano più di 40 milioni*

PROJ_{Matricola, Nome, Età} (SEL_{Stipendio>40}(Impiegati))

Matricola	Nome	Età
7309	Rossi	34
5698	Bruni	43
4076	Mori	45
8123	Lupi	46

Esempio 3

Query 2: *Trovare le matricole dei capi degli impiegati che guadagnano più di 40 milioni*

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

PASSO 1: $S_1 = \text{SEL}_{\text{Stipendio}>40}(\text{Impiegati})$

PASSO 2: $S_2 = S_1 \text{ JOIN}_{\text{Matricola=Impiegato}} \text{Supervisione}$

PASSO 3: $S_3 = \text{PROJ}_{\text{Capo}}(S_2)$

Esempio 4

Query 2: *Trovare nomi e stipendi dei capi degli impiegati che guadagnano più di 40 milioni*

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

PASSO 1: $S_1 = \text{SEL}_{\text{Stipendio}>40}(\text{Impiegati})$

PASSO 2: $S_2 = S_1 \text{ JOIN}_{\text{Matricola=Impiegato}} \text{Supervisione}$

PASSO 3: $S_3 = \text{PROJ}_{\text{Capo}}(S_2)$

PASSO 4: $S_4 = S_3 \text{ JOIN}_{\text{Capo=Matricola}} \text{Impiegati}$

PASSO 5: $S_5 = \text{PROJ}_{\text{Nome, Stipendio}}(S_4)$

Esempio 5

Query 2: Trovare le matricole dei capi i cui impiegati guadagnano **tutti** più di 40 milioni

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

PASSO 1: $S_1 = \text{SEL}_{\text{Stipendio} \leq 40}(\text{Impiegati})$

PASSO 2: $S_2 = S_1 \text{ JOIN}_{\text{Matricola}=\text{Impiegato}} \text{Supervisione}$

PASSO 3: $S_3 = \text{PROJ}_{\text{Capo}}(S_2)$

PASSO 4: $S_4 = \text{PROJ}_{\text{Capo}}(\text{Supervisione})$

PASSO 5: $S_5 = S_4 - S_3$

Equivalenza di espressioni

- Due espressioni dell'algebra sono *equivalenti* se producono lo stesso risultato, qualunque sia l'istanza attuale della base di dati
- Gli utenti formulano le interrogazioni usando linguaggi dichiarativi (SQL)
- I DBMS sono dotati di un *motore algebrico* e valutano le interrogazioni calcolando un'espressione algebrica
- L'equivalenza è importante perché i DBMS possono scegliere tra tutte le espressioni equivalenti quella cui corrisponde il minore costo esecutivo

Un caso importante

- Conviene anticipare le selezioni (*push selections*)
- Le selezioni tipicamente riducono in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione).

Esempio: se A è un attributo di R_1

1. $S_1 = R_1 \text{ JOIN } R_2$

2. $S_2 = \text{SEL}_{A=10}(S_1)$

1. $S_1 = \text{SEL}_{A=10}(R_1)$

2. $S_2 = S_1 \text{ JOIN } R_2$

- Il *piano esecutivo* di destra è sicuramente meno costoso di quello di sinistra, perché il join viene effettuato tra R_2 e una relazione sicuramente più piccola di R_1

Condizioni e valori nulli

Impiegati

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

$\text{SEL}_{\text{Età} > 40}(\text{Impiegati})$

$\text{SEL}_{\text{Età} \leq 40}(\text{Impiegati})$

- Le condizioni atomiche sono vere *solo per i valori non nulli*
- L'ultima tupla non viene presa in nessuna delle due selezioni

Gestione dei valori nulli

- Per riferirsi ai valori nulli esistono apposite condizioni atomiche: **IS NULL** e **IS NOT NULL**

Impiegati

Matricola	Cognome	Filiale	Età
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

SEL (Età > 40) OR (Età IS NULL) **(Impiegati)**