Algebra e Calcolo Relazionale (I)

PROF. DIOMAIUTA CRESCENZO

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA «*LUIGI VANVITELLI*»

Linguaggi per basi di dati

- Operazioni sullo schema
  - DDL: data definition language
- Operazioni sui dati
  - DML: data manipulation language
    - interrogazione ("query")
    - aggiornamento

Linguaggi di interrogazione per basi di dati relazionali

- **Dichiarativi**
  - specificano le proprietà del risultato ("**che cosa**")
- **Procedurali**
  - specificano le modalità di generazione del risultato ("**come**")

Linguaggi di interrogazione

- Algebra relazionale: procedurale
- Calcolo relazionale: dichiarativo (teorico)
- SQL (Structured Query Language): parzialmente dichiarativo (reale)
- QBE (Query by Example): dichiarativo (reale)

Algebra relazionale

- L'algebra relazionale è un ***linguaggio procedurale*** per il modello relazionale:
  - le operazioni vengono specificate descrivendo i passi necessari per ottenere il risultato.
- E' costituito da un insieme di operatori definiti su relazioni che producono altre relazioni
  - Insiemistici
  - Specifici
  - Join

Da aspettarselo perché la relazione è un insieme

Operatori Insiemistici

- Le relazioni sono insiemi quindi ha senso considerare i classici operatori insiemistici:
  - **Unione ∪**
  - **Intersezione ∩**
  - **Differenza –**
- Una relazione è un insieme di tuple omogenee, definite sugli stessi attributi
  - **E' possibile definire queste operazioni su relazioni con attributi diversi ma la cosa non ha molto senso.**

Unione

- **Def**: L' unione di due relazioni R_1 e R_2 definite **sugli stessi attributi** (X), indicata con $R_1 \cup R_2$, è la relazione (ancora definita su X) composta dalle tuple che appartengono ad R_1 oppure a R_2, oppure ad entrambe.

LAUREATI

Matricola	Cognome	Età
7274	Rossi	37
7432	Verdi	39
9824	Bianchi	38

DIRIGENTI

Matricola	Cognome	Età
9297	Neri	56
7432	Verdi	39
9824	Bianchi	38

LAUREATI ∪ DIRIGENTI

Matricola	Cognome	Età
7274	Rossi	37
7432	Verdi	39
9824	Bianchi	38
9297	Neri	56

Intersezione

- **Def**: L' **intersezione** di R_1 e R_2 definite sugli stessi attributi (X), indicata con $R_1 \cap R_2$, è la relazione (ancora definita su X) composta dalle tuple che appartengono sia ad R_1 sia a R_2.

LAUREATI

Matricola	Cognome	Età
7274	Rossi	37
7432	Verdi	39
9824	Bianchi	38

DIRIGENTI

Matricola	Cognome	Età
9297	Neri	56
7432	Verdi	39
9824	Bianchi	38

LAUREATI ∩ DIRIGENTI

Matricola	Cognome	Età
7432	Verdi	39
9824	Bianchi	38

Differenza

- **Def**: La **differenza** di R_1 e R_2, indicata con $\mathbf{R_1 - R_2}$, (**definite sugli stessi attributi** X) è la relazione (ancora definita su X) composta dalle tuple che appartengono ad R_1 e non ad R_2.

LAUREATI

Matricola	Cognome	Età
7274	Rossi	37
7432	Verdi	39
9824	Bianchi	38

DIRIGENTI

Matricola	Cognome	Età
9297	Neri	56
7432	Verdi	39
9824	Bianchi	38

LAUREATI – DIRIGENTI

Matricola	Cognome	Età
7274	Rossi	37

Ridenominazione

- La limitazione posta relativamente al fatto di considerare relazioni sugli stessi attributi è particolarmente pesante in casi come:

PATERNITA'

Padre	**Figlio**
Adamo	Caino
Adamo	Abele
Abramo	Isacco

MATERNITA'

Madre	**Figlio**
Eva	Caino
Eva	Set
Sara	Isacco

- Se consideriamo ad esempio,

Paternità ∪ Maternità ???

ha senso per ottenere tutte le coppie «**genitore-figlio**».

Ridenominazione

- Per poter operare in queste situazioni è stato introdotto l' operatore ridenominazione: $\rho_{A1 \leftarrow A2}$
- **Def**: Sia R una relazione definita sull'insieme di attributi X e sia Y un altro insieme di attributi con la stessa cardinalità, siano $A_1,A_2,\ldots,A_K$ e $B_1,B_2,\ldots,B_K$ un ordinamento per gli attributi in X e uno per quelli in Y, allora la ridenominazione

$$\rho_{B_1,B_2,\ldots,B_K \leftarrow A_1,A_2,\ldots,A_K}$$

In questo caso è importante l'ordine

Contiene una tupla t' per ciascuna tupla t in R, definita come segue: t' è una tupla su Y e $t'[B_i] = t[A_i]$, per i =1,…,K

- <u>**La definizione in pratica conferma che cambiano solo i nomi degli attributi**</u>
- E' possibile, una volta stabilito un ordinamento, la ridenominazione su più campi.

Ridenominazione - Esempio

- La ridenominazione agisce solo sullo schema, il corpo rimane invariato

PATERNITA’

Padre	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco

$\rho_{\text{genitore} \leftarrow \text{padre}}$ (Paternità)

Genitore	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco

- ρ -> **Operatore di ridenominazione (monadico – un solo argomento)**

Ridenominazione - Esempio

$\rho_{\text{genitore} \leftarrow \text{padre}}(\text{Paternità}) \cup \rho_{\text{genitore} \leftarrow \text{madre}}(\text{Maternità})$

PATERNITA'

Genitore	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco

MATERNITA'

Genitore	Figlio
Eva	Caino
Eva	Set
Sara	Isacco

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

Ridenominazione – Esempio 2

$\rho_{\text{Sede,Retribuzione} \leftarrow \text{Agenzia,Stipendio}}(\text{Impiegati}) \cup \rho_{\text{Sede,Retribuzione} \leftarrow \text{Fabbrica,Salario}}(\text{Operai})$

IMPIEGATI

Cognome	Agenzia	Stipendio
Rossi	Roma	45
Neri	Napoli	53

OPERAI

Cognome	Fabbrica	Salario
Verdi	Latina	33
Bruni	Milano	32

Cognome	Sede	Retribuzione
Rossi	Roma	45
Neri	Napoli	53
Verdi	Latina	33
Bruni	Milano	32

Formula proposizionale

- Data la relazione r(X), una FP è una formula ottenuta combinando con gli operatori and ($\wedge$), or ($\vee$) e not ($\neg$) condizioni logiche semplici del tipo A θ B o A θ c dove:
  - **θ** è un operatore di confronto ($=, \neq, >, <, \geq, \leq$)
  - **A** e **B** sono attributi in X tali che il confronto θ abbia senso
  - **c** è una costante «compatibile» con il dominio di A (c $\in$ dom(A)).

Posso creare un PREDICATO usando questi valori. Esso può assumere un valore vero o falso

Formula proposizionale

- Data una formula F e una tupla t, è definito un valore di verità per F su t:
  - **A θ B** è vero se t[A] è in relazione θ con t[B] (es: A=B è vera se t[A]=t[B]).
  - **A θ c** è vero se t[A] è in relazione θ con c.
  - **F1 $\wedge$ F2, F1 $\vee$ F2, $\neg$ F1** hanno il significato indicato dagli operatori.

Selezione

- **Selezione** è un operatore monadico (opera su una sola relazione e fornisce come risultato una porzione dell’operando).
- Produce un risultato che:
  - ha lo stesso schema dell'operando
  - contiene un sottoinsieme delle ennuple dell'operando,
  - quelle che soddisfano una condizione

Selezione

- La selezione $\sigma_F(r)$ produce una relazione sugli stessi attributi di r che contiene le tuple di r su cui F è vera
- Notazione:
  - $\sigma_F(r)$
- Semantica:
- $\sigma_F(r) = \{ t \mid t \in r \text{ AND } F(t) = \text{vero}\}$

Grado della relaz. di partenza (n. colonne) è uguale a quello di arrivo. n. di righe ≤ a quello di partenza

* *F è una condizione*: espressione booleana (come quelle dei vincoli di ennupla)

Selezione - Esempi

Impiegati

Cognome	Nome	Età	Stipendio
Rossi	Mario	25	2000
Neri	Luca	40	3000
Verdi	Nicola	36	4500
Rossi	Marco	40	3900

$$\sigma_{\text{Età} > 30 \wedge \text{Stipendio} > 4000}(\text{Impiegati})$$

Cognome	Nome	Età	Stipendio
Verdi	Nicola	36	4500

Selezione - Esempi

Cittadini

Cognome	Nome	CittàDiNascita	Residenza
Rossi	Maria	Roma	Milano
Neri	Luca	Bergamo	Bergamo
Verdi	Nicola	Siena	Siena
Rossi	Marco	Napoli	Firenze

$\sigma_{\text{CittàDiNascita=Residenza}}$ **(Cittadini)**

Cognome	Nome	CittàDiNascita	Residenza
Neri	Luca	Bergamo	Bergamo
Verdi	Nicola	Siena	Siena

Selezione e Proiezione

- **Selezione** e **Proiezione** svolgono due funzioni che possono essere definite **ortogonali** tra loro.
- **Selezione**:
  - Produce un sottoinsieme delle tuple su tutti gli attributi
- **Proiezione**:
  - Produce un risultato cui contribuiscono tutte le tuple su un sottoinsieme di attributi

Decomposizioni orizzontali

Decomposizioni verticali

Proiezione

- **Proiezione** è un operatore monadico (opera su una sola relazione e fornisce come risultato una porzione dell'operando).
- Produce un risultato che:
  - ha parte degli attributi dell'operando
  - contiene ennuple cui contribuiscono tutte le ennuple dell'operando

Grado della relaz. di partenza (n. colonne) è ≤ a quella di arrivo. n. di righe ≤ a quello di partenza

↑ cardinalità

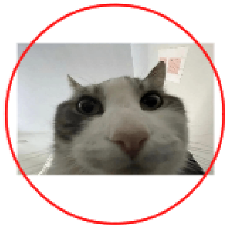
Proiezione

- Proiezione:
  - data una relazione $r(X)$ ed un sottoinsieme Y di X, $\pi_Y(r)$ produce una relazione costituita dalle tuple di r considerando solo i valori su Y.
- Notazione:
  - $\pi_Y(r)$
- Semantica:
  - $\pi_Y(r) = \{ t[Y] \mid t \in r \}$

Proiezione - Esempi

Impiegati

Cognome	Nome	Reparto	Capo
Rossi	Mario	Vendite	Gatti
Neri	Lucio	Vendite	Gatti
Verdi	Mario	Personale	De Micheli
Rossi	Marco	Personale	De Micheli



$\pi_{\text{Cognome, Nome}}(\textbf{Impiegati})$

Cognome, Nome sono superchiave

Cognome	Nome
Rossi	Mario
Neri	Lucio
Verdi	Mario
Rossi	Marco

Proiezione - Esempi

Impiegati

Cognome	Nome	Reparto	Capo
Rossi	Mario	Vendite	Gatti
Neri	Lucio	Vendite	Gatti
Verdi	Mario	Personale	De Micheli
Rossi	Marco	Personale	De Micheli

Reparto, Capo non sono superchiave

quindi la cardinalità è diminuita

$\pi_{\text{Reparto, Capo}}$**(Impiegati)**

Reparto	Capo
Vendite	Gatti
Personale	De Micheli

Ho avuto doppioni, ma non li prendo in considerazione perché sto lavorando con insiemi.*

* Qui è diminuita anche la cardinalità!

Cardinalità delle proiezioni

- La cardinalità della relazione risultante è minore o uguale della cardinalità della relazione di partenza.
- Esiste un legame tra le **proiezioni** e i **vincoli di chiave**
- $\pi_Y(r)$ contiene lo stesso numero di tuple di r se e solo se Y è superchiave per r
  - Se Y è superchiave, r non contiene tuple uguali su Y e quindi nella proiezione entrano tutte le tuple

Proiezione - Esempi

Studenti

Matricola	Cognome	Nome	DataDiNascita	Corso
284328	Rossi	Luigi	29/04/01	Ing. Informatica
296328	Rossi	Giovanni	29/04/02	Ing. Informatica
587614	Rossi	Lucio	01/05/02	Ing. Meccanica
934856	Neri	Paolo	01/05/03	Ing. Civile
965536	Neri	Paolo	05/03/02	Ing. Civile

$\pi_{\text{Cognome, Corso}}(\text{Studenti})$

Cognome	Corso
Rossi	Ing. Informatica
Rossi	Ing. Meccanica
Neri	Ing. Civile

Proiezione - Esempi

Studenti

Matricola	Cognome	Nome	DataDiNascita	Corso
284328	Rossi	Luigi	29/04/01	Ing. Informatica
296328	Rossi	Giovanni	29/04/02	Ing. Aerospaziale
587614	Neri	Lucio	01/05/02	Ing. Meccanica
965536	Neri	Paolo	05/03/02	Ing. Civile

$\pi_{\text{Cognome, Corso}}(\text{Studenti})$

Cognome	Corso
Rossi	Ing. Informatica
Rossi	Ing. Aerospaziale
Neri	Ing. Meccanica
Neri	Ing. Civile

Selezione e Proiezione

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

Impiegati

Cognome	Nome	Età	Stipendio
Rossi	Mario	25	2000
Neri	Luca	40	3000
Verdi	Nicola	36	4500
Rossi	Marco	40	3900

Cognome e Nome degli impiegati che hanno più di 30 anni e che guadagnano più di 4000

Ovvio che faccio prima selezione e poi proiezione: se proietto prima perdo lo stipendio. Prima filtro sulle righe e poi sugli attributi.

$$\pi_{\text{Cognome, Nome}}(\sigma_{\text{Età} > 30 \wedge \text{Stipendio} > 4000}(\text{Impiegati}))$$

Cognome	Nome
Verdi	Nicola

Selezione e Proiezione

- Combinando selezione e proiezione, possiamo estrarre informazioni da una relazione
- Non possiamo però correlare informazioni presenti in relazioni diverse, né informazioni in ennuple diverse di una stessa relazione

Join Naturale

- È un operatore che combina le tuple di due relazioni sulla base dell’**uguaglianza dei valori degli attributi comuni alle due relazioni**

r_1

Impiegato	Reparto
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Reparto	Capo
produzione	Mori
vendite	Bruni

$r_1 \bowtie r_2$

Impiegato	Reparto	Capo
Rossi	vendite	Bruni
Neri	Produzione	Mori
Bianchi	produzione	Mori

Si ha una relazione sull’unione degli insiemi di attributi degli operandi e le sue tuple sono la combinazione delle tuple degli operandi con valori uguali sugli attributi comuni (Reparto)

Join Naturale: definizione

- Operatore binario (generalizzabile)
- Produce un risultato
  - sull'unione degli attributi degli operandi
  - con ennuple costruite ciascuna a partire da una ennupla di ognuno degli operandi

Join Naturale: definizione

- Il join naturale di $R_1(X_1)$ e $R_2(X_2)$ è una relazione definita su X_1 e X_2:
  - $R_1 \bowtie R_2 = \{t \text{ su } (X_1 \cup X_2) \mid t[X_1] \in R_1 \text{ e } t[X_2] \in R_2\}$
- La definizione implica che le tuple del risultato sono ottenute combinando tuple degli operandi con valori uguali sugli attributi comuni («match»).
  - Sia $X_{1,2}=X_1 \cap X_2$ (valori comuni), dalla definizione $\forall\, t \in (R_1 \bowtie R_2)$
    - $\exists\, t_1 \in R_1 \mid t_1=t[X_{1,2}]$ e $\exists\, t_2 \in R_2 \mid t_2=t[X_{1,2}] \Rightarrow t_1[X_{1,2}] = t_2[X_{1,2}]$

Join Naturale: Esempi

Il nome del campo è lo stesso per gli attributi che ci fanno fare il match per join

Esami

Matricola	CodCorso	Voto	Lode
2932	484	28	NO
3965	739	30	SÌ
2932	953	26	NO
3546	953	30	NO

Corsi

CodCorso	Titolo	Docente	Anno
484	Analisi	Biondi	1
739	Analisi	Neri	1
953	Basi di Dati	Rossi	2

Esami $\bowtie$ Corsi

Matricola	CodCorso	Voto	Lode	Titolo	Docente	Anno
2932	484	28	NO	Analisi	Biondi	1
3965	739	30	SÌ	Analisi	Neri	1
2932	953	26	NO	Basi di Dati	Rossi	2
3546	953	30	NO	Basi di Dati	Rossi	2

Join Naturale: Esempi

Voli

Codice	Data	Comandante
AZ427	21/07/2001	Bianchi
AZ427	23/07/2001	Rossi
TW056	21/07/2001	Neri

Linee

Codice	Partenza	Arrivo
AZ427	FCO	JFK
TW056	LAX	FCO

Voli $\bowtie$ Linee

Codice	Data	Comandante	Partenza	Arrivo
AZ427	21/07/2001	Bianchi	FCO	JFK
AZ427	23/07/2001	Rossi	FCO	JFK
TW056	21/07/2001	Neri	LAX	FCO

Join Naturale: Esempi

Prenotazioni

Codice	Data	Classe	Cliente
AZ427	21/07/2001	Economy	A. Bini
AZ427	21/07/2001	Business	F. Dini
AZ427	23/07/2001	Economy	A. Cini

Voli ▷◁ Prenotazioni

Codice	Data	Comandante	Classe	Cliente
AZ427	21/07/2001	Bianchi	Economy	A. Bini
AZ427	21/07/2001	Bianchi	Business	F. Dini
AZ427	23/07/2001	Rossi	Economy	A. Cini

LInee ▷◁ Prenotazioni

Codice	Partenza	Arrivo	Data	Classe	Cliente
AZ427	FCO	JFK	21/07/2001	Economy	A. Bini
AZ427	FCO	JFK	21/07/2001	Business	F. Dini
AZ427	FCO	JFK	23/07/2001	Economy	A. Cini

Per il Join Naturale il grado è la somma dei due gradi. Nella teoria. Nella pratica è minore

grado è dato dalla somma

Join Naturale - Osservazioni

- Il grado del risultato è minore o uguale della somma dei gradi delle relazioni partecipanti (**gli attributi omonimi compaiono una sola volta**).
  - Se il grado della relazione R è m e il grado di S è n, il grado della relazione è m+n
- Il join si dice **completo** se ogni tupla di ogni operando contribuisce ad almeno una tupla del risultato.
- In un join **non completo**, le tuple che non partecipano al risultato si dicono dondolanti (***dangling***).
- Il caso limite è quello in cui il risultato del join sia vuoto.

Join Naturale - Incompleto

Impiegato lavora in un dipartimento in cui non c'è capo

r_1

Impiegato	Dipartimento
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Dipartimento	Capo
produzione	Mori
acquisto	Bruni

$r_1 \bowtie r_2$

Impiegato	Dipartimento	Capo
Neri	produzione	Mori
Bianchi	produzione	Mori

Join Naturale - Vuoto

r_1

Impiegato	Dipartimento
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Dipartimento	Capo
marketing	Mori
acquisto	Bruni

$r_1 \bowtie r_2$

Impiegato	Dipartimento	Capo

Non c'é nessuna tupla che rende vero quel matching.
NOTA: il nome dell'attributo é ancora lo stesso

Join Naturale – Completo con n x m ennuple

r_1

Impiegato	Dipartimento
Rossi	B
Neri	B

r_2

Dipartimento	Capo
B	Mori
B	Bruni

É una sorta di prodotto cartesiano

$r_1 \bowtie r_2$

Impiegato	Dipartimento	Capo
Rossi	B	Mori
Rossi	B	Bruni
Neri	B	Mori
Neri	B	Bruni

Join Naturale - Cardinalità

- Il join di R1 e R2 contiene un numero di ennuple compreso fra zero e il prodotto di |R1| e |R2|
- Se il join coinvolge una chiave di R2, allora il numero di ennuple è compreso fra zero e |R1|
- Se il join coinvolge una chiave di R2 e un vincolo di integrità referenziale, allora il numero di ennuple è pari a |R1| (es. slide 31)

Join Naturale - Cardinalità

Formule di quanto visto prima

- R1(A,B) , R2 (B,C)
- in generale

$$0 \leq |R1\ \text{JOIN}\ R2| \leq |R1| \times |R2|$$

- se B è chiave in R2

$$0 \leq |R1\ \text{JOIN}\ R2| \leq |R1|$$

- se B è chiave in R2 ed esiste vincolo di integrità referenziale fra B (in R1) e R2:

$$|R1\ \text{JOIN}\ R2| = |R1|$$

Join Naturale – Tuple lasciate fuori

r_1

Impiegato	Dipartimento
Rossi	vendite
Neri	produzione
Bianchi	produzione

r_2

Dipartimento	Capo
produzione	Mori
acquisto	Bruni

$r_1 \bowtie r_2$

Impiegato	Dipartimento	Capo
Neri	produzione	Mori
Bianchi	produzione	Mori

Posso comunque estrapolare info per le tuple tagliate fuori: il dipartimento vendite non ha capo.

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

Join Esterno

- Serve ad evitare di “tralasciare” tuple
- Le tuple che non hanno una controparte vengono completate con valori **nulli**.
- Esistono tre tipi di join esterno
  - **Left**: estende solo le tuple del primo operando
  - **Right**: estende solo le tuple del secondo operando
  - **Full**: estende tutte le tuple sia quelle sinistre che quelle destre

Semi Join: Join tra R1 ed R2 e proiez. su attrib. di R1.

Join Esterno - Esempi

r_1

Impiegato	Dipartimento
Rossi	vendita
Neri	produzione
Bianchi	produzione

r_2

Dipartimento	Capo
produzione	Mori
acquisto	Gatti

$r_1 \bowtie_{\text{LEFT}} r_2$

Impiegato	Dipartimento	Capo
Rossi	**Vendita**	**NULL**
Neri	Produzione	Mori
Bianchi	Produzione	Mori

$r_1 \bowtie_{\text{RIGHT}} r_2$

Impiegato	Dipartimento	Capo
Neri	Produzione	Mori
Bianchi	Produzione	Mori
NULL	**Acquisto**	**Gatti**

$r_1 \bowtie_{\text{FULL}} r_2$

Impiegato	Dipartimento	Capo
Rossi	**Vendita**	**NULL**
Neri	Produzione	Mori
Bianchi	Produzione	Mori
NULL	**acquisto**	**Gatti**

Join e Proiezioni

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

SEMI JOIN

Impiegato	Reparto
Neri	B
Bianchi	B

Reparto	Capo
B	Mori

Join e proiezioni

- $R_1(X_1), R_2(X_2)$

$$\pi_{X1} (R_1 \bowtie R_2) \subseteq R_1$$

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

$$(\pi_{X1}(R)) \bowtie (\pi_{X2}(R)) \supseteq R$$

Fare proiezioni ci aiuta ad estrapolare informazioni:

Join + Proiez. di attrib. su 1ª relazione ottengo sottoinsieme di R_1

R = Join tra R_1 e R_2.

Proioni e join es. 2 di prima

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

Bianchi lo trovo con bruni e con mori. Tuple non corrette.

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

Join Naturale N-Ario

- Il join naturale è un operatore **commutativo** $R_1 \bowtie R_2 = R_2 \bowtie R_1$.
- Il join naturale è un operatore **associativo** $R_1 \bowtie (R_2 \bowtie R_3) = (R_1 \bowtie R_2) \bowtie R_3)$.
- Esistono poi due casi estremi:
  - X1 e X2 coincidenti => in questo caso si ottiene **l'intersezione**
  - X1 e X2 disgiunti => si ottiene il **prodotto cartesiano**.
- Un prodotto cartesiano è tipicamente inutile perché concatena tuple non correlate. Esso è allora in genere seguito da una selezione che centra l'interesse su tuple significative.

Join Naturale N-Ario

r_1

Impiegato	Dipartimento
Rossi	vendita
Neri	Produzione
Pinco	marketing
Bianchi	produzione

r_2

Dipartimento	Divisione
produzione	A
marketing	B
acquisto	B

r3

Divisione	Capo
A	Mori
B	Gatti

$r_1 \bowtie r_2 \bowtie r_3$

Impiegato	Dipartimento	Divisione	Capo
Neri	produzione	A	Mori
Pinco	marketing	B	Gatti
Bianchi	produzione	A	Mori

Join Naturale

- Il join naturale è definito anche quando gli operandi non hanno attributi in comune, in questo caso non viene imposta alcuna condizione sulle tuple e quindi come risultato ci sono tutte le tuple della prima combinata con tutte le tuple della seconda, si ha ovvero il **prodotto cartesiano**.
- Definizione impropria poiché il prodotto cartesiano è definito su coppie non su tuple

Impiegati

Impiegato	Progetto
Rossi	A
Neri	A
Neri	B

Progetti

Codice	Nome
1	Venere
2	Marte

Impiegati ⋈ Progetti

Impiegato	Progetto	Codice	Nome
Rossi	A	1	Venere
Neri	A	1	Venere
Neri	B	1	Venere
Rossi	A	2	Marte
Neri	A	2	Marte
Neri	B	2	Marte

Join Naturale

- Il join naturale è definito anche quando gli operandi non hanno attributi in comune, in questo caso non viene imposta alcuna condizione sulle tuple e quindi come risultato ci sono tutte le tuple della prima combinata con tutte le tuple della seconda, si ha ovvero il **prodotto cartesiano**.
- Definizione impropria poiché il prodotto cartesiano è definito su coppie non su tuple

Impiegati

Impiegato	Progetto
Rossi	A
Neri	A
Neri	B

Progetti

Codice	Nome
A	Venere
B	Marte

Impiegati ⋈ Progetti

Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	A	Venere
Neri	B	A	Venere
Rossi	A	B	Marte
Neri	A	B	Marte
Neri	B	B	Marte

Qui ho nomi diversi per gli attributi!

NOTA: Join Naturale impone avere nomi uguali

Theta-join ed Equi-join

- Tale considerazione porta a definire un operatore aggregato (theta-join) su due relazioni R1 e R2 (che non hanno attributi in comune) come **prodotto cartesiano seguito da una selezione** su F:

$$\mathbf{R1 \bowtie_{F} R2 = \sigma_F(R1 \bowtie R2)}$$

- Un theta-join in cui la F sia una congiunzione di atomi di uguaglianza (Operatore =), con un attributo della prima relazione ed un attributo della seconda relazione si dice **equi-join**.

↳ Attributi con nomi diversi!

NOTA: Join è abbastanza laboriosa come computazione. Per una query qualsiasi lui fa il prodotto cartesiano

Theta-join

Impiegati

Impiegato	Progetto
Rossi	A
Neri	A
Neri	B

Progetti

Codice	Nome
A	Venere
B	Marte

$$\sigma_{\text{Progetto=Codice}} (\text{Impiegati} \bowtie \text{Progetti})$$

Impiegato	Progetto	Codice	Nome
Rossi	**A**	**A**	**Venere**
Neri	**A**	**A**	**Venere**
Neri	**B**	**B**	**Marte**

Theta-join - Esempio

Qui si nota che CodProgetto e Sigla hanno attributo differente

Ricercatori

Nome	CodProgetto
Rossi	PK27
Verdi	AL2000
Bianchi	PK27
Verdi	FK28
Neri	AL2000

Progetti

Sigla	Responsabile
PK27	Bianchi
AL2000	Neri
FK28	Verdi

Ricercatori $\bowtie_{\text{CodProgetto=Sigla}}$ Progetti

Nome	CodProgetto	Sigla	Responsabile
Rossi	PK27	PK27	Bianchi
Verdi	AL2000	AL2000	Neri
Bianchi	PK27	PK27	Bianchi
Verdi	FK28	FK28	Verdi
Neri	AL2000	AL2000	Neri

Ricercatori $\bowtie_{\text{(CodProgetto=Sigla) AND (Nome} \neq \text{Responsabile)}}$ Progetti

Nome	CodProgetto	Sigla	Responsabile
Rossi	PK27	PK27	Bianchi
Verdi	AL2000	AL2000	Neri

Interrogazioni (Query)

schema di base di dati

- Una **interrogazione** su un DB con schema R è una funzione che ad ogni istanza r di R associa una relazione con un dato schema

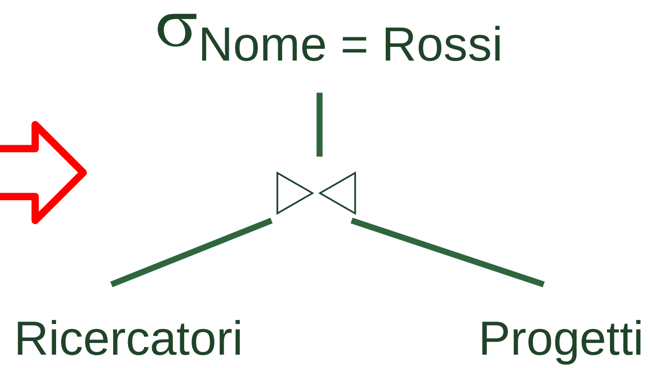
Combinando operatori posso interrogare

Espressioni e viste

- Gli operatori dell'Algebra Relazionale si possono liberamente combinare tra loro, avendo cura di rispettare le regole stabilite per la loro applicabilità
- Oltre alla rappresentazione "**lineare**" è anche possibile adottare una rappresentazione grafica in cui l'espressione è rappresentata ad **albero**

$\sigma_{\text{Nome = Rossi}}(\text{Ricercatori} \bowtie \text{Progetti})$

Prima Join, poi selezione



- Per "semplificare" espressioni complesse è possibile fare uso di **viste**, cioè espressioni a cui viene assegnato un nome e che è possibile riutilizzare in di altre espressioni

ProgettiRossi = $\sigma_{\text{Nome = Rossi}}(\text{Ricercatori} \bowtie \text{Progetti})$

Interrogazioni: Esempi

VINCOLO DI INTEGRITÀ REFERENZIALE
si basa sempre su chiave PRIMARIA!
Quindi legame tra Sedi e Prog tramite
città non è legato da quel vincolo

Imp

→ Vincolo di integrità referenziale

CodImp	Nome	*Sede*	Ruolo	Stipendio
E001	Rossi	S01	Analista	2000
E002	Verdi	S02	Sistemista	1500
E003	Bianchi	S01	Programmatore	1000
E004	Gialli	S03	Programmatore	1000
E005	Neri	S02	Analista	2500
E006	Grigi	S01	Sistemista	1100
E007	Violetti	S01	Programmatore	1000
E008	Aranci	S02	Programmatore	1200

Sedi

Sede	Responsabile	Citta
S01	Biondi	Milano
S02	Mori	Bologna
S03	Fulvi	Milano

Prog

CodProg	Citta
P01	Milano
P01	Bologna
P02	Bologna

↓ legame fatto su attributo che non è chiave

1ª Cosa da fare: Capire quali tabelle c'entrano per i dati che cerco.

Interrogazioni: Esempi

1) Nome, sede e stipendio degli impiegati che guadagnano più di 1300 Euro, definendo la vista ImpFac

$$ImpFac = \pi_{Nome,Sede,Stipendio}(\sigma_{Stipendio > 1300}(Imp))$$

oppure:

$$ImpFac = \sigma_{Stipendio > 1300}(\pi_{Nome,Sede,Stipendio}(Imp))$$

ImpFac

Nome	Sede	Stipendio
Rossi	S01	2000
Verdi	S02	1500
Neri	S02	2500

Nella tabella impiegati non ho info su responsabili e città

↳ quindi dovrò usare il Join

2) Sedi, responsabili e città degli impiegati che guadagnano più di 1300 Euro

$$\pi_{Sede,Responsabile,Citta}(Sedi \bowtie (\sigma_{Stipendio > 1300}(Imp)))$$

oppure: $\pi_{Sede,Responsabile,Citta}(Sedi \bowtie ImpFac)$

Sede	Responsabile	Citta
S01	Biondi	Milano
S02	Mori	Bologna

Interrogazioni: Esempi

3) Progetti nelle città delle sedi degli impiegati che guadagnano più di 1300 Euro

$$\pi_{CodProg}(Prog \bowtie (Sedi \bowtie ImpFac))$$

CodProg
P01
P02

Triplo Join per prendere tutte le relazioni

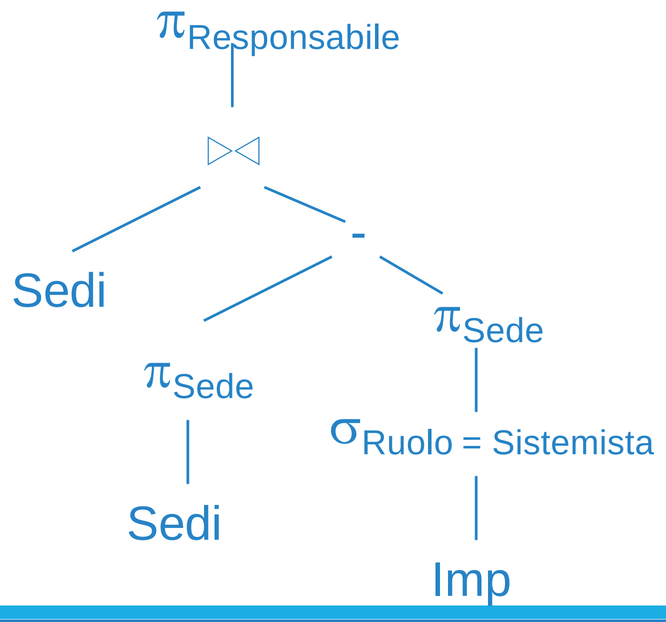
Interrogazioni: Esempi

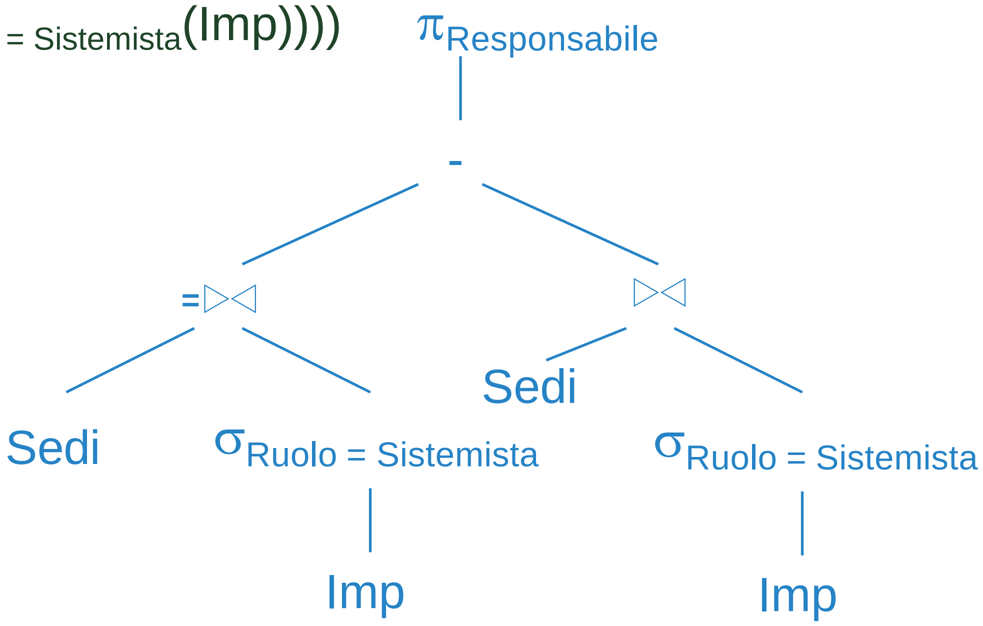
4) Responsabili delle sedi senza sistemisti

Responsabile
Fulvi

$$\pi_{Responsabile}(\text{Sedi} \bowtie (\pi_{Sede}(\text{Sedi}) - \pi_{Sede}(\sigma_{Ruolo = Sistemista}(\text{Imp}))))$$

oppure: $\pi_{Responsabile}((\text{Sedi} =\!\bowtie (\sigma_{Ruolo = Sistemista}(\text{Imp}))) - (\text{Sedi} \bowtie (\sigma_{Ruolo = Sistemista}(\text{Imp}))))$





Interrogazioni: Esempi

Impiegati(Matr, Nome, Età, Stip)

Supervisione(Capo, Imp)

- trovare Matricola, Nome ed Età degli impiegati che guadagnano più di 40.

Impiegati

Matr	Nome	Età	Stip
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

Supervisione

Campo	Imp
210	101
210	103
210	104
231	105
301	210
301	231
375	252

- Soluzione
  - R1 = $\sigma_{Stip>40}$(Impiegati)
  - R2 = $\pi_{matr,nome,età}$(R1)

Interrogazioni: Esempi

- Trovare la matricola dei capi degli impiegati che guadagnano più di 40.
- Soluzione 1
  - $R1 = \sigma_{Stip>40}(Imp)$
  - $R2 = Sup \bowtie_{Imp=matr} R1$
  - $R3 = \pi_{capo}(R2)$
- Soluzione 2
  - $R1 = \pi_{matr}(\sigma_{Stip>40}(Imp))$
  - $R2 = Sup \bowtie_{Imp=matr} R1$
  - $R3 = \pi_{capo}(R2)$

Interrogazioni: Esempi

- Trovare nomi e stipendi dei capi degli impiegati che guadagnano più di 40.
- Soluzione
  - $R1 = \pi_{matr}(\sigma_{Stip>40}(Imp))$
  - $R2 = Sup \bowtie_{Imp=matr} R1$
  - $R3 = \pi_{capo}(R2)$
  - $R4 = IMP \bowtie_{Matr=Capo} R3$
  - $R5 = \pi_{Nome,Stip}(R4)$

Interrogazioni: Esempi

- Trovare gli impiegati che guadagnano più del proprio capo, mostrando matricola, nome e stipendio di ciascuno di essi e del relativo capo.
- Soluzione
  - $R1 = (Sup \bowtie_{Imp=matr} Imp)$
  - $R2 = \rho_{MatrC,NomeC,StipC,EtaC \leftarrow Matr,Nome,Stip,Eta}(Imp)$
  - $R3 = R1 \bowtie_{MatrC=Capo} (R2)$
  - $R4 = \sigma_{Stip>StipC}(R3)$
  - $R5 = \pi_{Matr,\ Nome,\ Stip,\ MatrC,\ NomeC,\ StipC}(R4)$

Interrogazioni: Esempi

- Trovare matricola e nome dei capi i cui impiegati guadagnano **<u>tutti</u>** più di 40.
- Soluzione
  - R1 = $\sigma_{Stip \leq 40}(Imp)$
  - R2 = $\pi_{Capo}(Sup \bowtie_{Imp=Matr} R1)$
  - R3 = $\pi_{Capo}(Sup) - R2$
  - R4 = $IMP \bowtie_{Matr=Capo} R3$
  - R5 = $\pi_{Matr,Nome}(R4)$