

Join, una difficoltà

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

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

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

- Vedere anche su Relax (nella base di dati mostrata per gli esempi, usare R1 e R2)

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\text{JOIN}_{\text{LEFT}}$ Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	<i>NULL</i>

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\text{JOIN}_{\text{RIGHT}}$ Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
<i>NULL</i>	C	Bruni

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\text{JOIN}_{\text{FULL}}$ Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	<i>NULL</i>
<i>NULL</i>	C	Bruni

- Su Relax il risultato ha una forma leggermente diversa, ma il concetto è lo stesso

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

Impiegato	Reparto
Neri	B
Bianchi	B

Reparto	Capo
B	Mori

Join e proiezioni

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

$$\text{PROJ}_{X_1}(R_1 \text{ JOIN } R_2) \subseteq R_1$$

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
Neri	B	Bruni
Bianchi	B	Mori
Bianchi	B	Bruni
Verdi	A	Bini

Proiezioni e join

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

Impiegato	Reparto
Neri	B
Bianchi	B
Verdi	A

Reparto	Capo
B	Mori
A	Bini

Join e proiezioni

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

$$\text{PROJ}_{X_1}(R_1 \text{ JOIN } R_2) \subseteq R_1$$

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

$$(\text{PROJ}_{X_1}(R)) \text{ JOIN } (\text{PROJ}_{X_2}(R)) \supseteq R$$

Join e proiezioni

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

$$\text{PROJ}_{X_1}(R_1 \text{ JOIN } R_2) \subseteq R_1$$

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

$$(\text{PROJ}_{X_1}(R)) \text{ JOIN } (\text{PROJ}_{X_2}(R)) \supseteq R$$

- Quando c'è uguaglianza?

Proiezioni e join

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

Impiegato	Reparto
Neri	B
Bianchi	B
Verdi	A

Reparto	Capo
B	Mori
A	Bini

Join e proiezioni

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

$$(\text{PROJ}_{X_1}(R)) \text{ JOIN } (\text{PROJ}_{X_2}(R)) \supseteq R$$

- Quando c'è uguaglianza?
 - quando gli attributi comuni alle due proiezioni $X_1 \cap X_2$ sono chiave (in effetti, basta superchiave) per una delle due proiezioni)

Prodotto cartesiano

- un join naturale su relazioni senza attributi in comune
- contiene sempre un numero di ennuple pari al prodotto delle cardinalità degli operandi (le ennuple sono tutte combinabili)

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN Reparti

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

- Il prodotto cartesiano, in pratica, ha senso (quasi) solo se seguito da selezione:

$SEL_{Condizione} (R_1 JOIN R_2)$

- L'operazione viene chiamata **theta-join** e indicata con

$R_1 JOIN_{Condizione} R_2$

Equi-join

- Se l'operatore di confronto nel theta-join è sempre l'uguaglianza (=) allora si parla di **equi-join**

Nota: ci interessa davvero l'equi-join, non il theta-join più generale

- **Equi-join: prodotto cartesiano seguito da selezione di uguaglianza**

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN_{Reparto=Codice} Reparti

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

Join naturale ed equi-join

- In pratica, ciò che ci interessa è l'equi-join
- Il join naturale lo abbiamo usato solo a fini didattici, perché i concetti sono più semplici
- Nelle interrogazioni "pratiche" useremo l'equi-join

Equivalenza di espressioni

- Due espressioni sono **equivalenti** se producono risultati uguali fra loro su ogni istanza della base di dati
- L'equivalenza è importante in pratica perché i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "costose"

Un'equivalenza importante

- Push selections (se A è attributo di R_1)

$$\text{SEL}_{A=10}(R_1 \text{ JOIN } R_2) = \text{SEL}_{A=10}(R_1) \text{ JOIN } R_2$$

Nota

- In questo corso, ci preoccupiamo poco dell'efficienza:
 - L'obiettivo è di scrivere interrogazioni corrette e leggibili
- Motivazione:
 - I DBMS si preoccupano di scegliere le strategie realizzative efficienti

Aggregazione

- Il contenuto delle basi di dati viene spesso aggregato:
 - Il voto massimo nell'esame di basi di dati
 - Il voto medio per ciascun esame
 - Il numero di CFU conseguiti da ciascuno studente

Aggregazione, sintassi e semantica

- Sintassi

AGG *Attributi Raggruppamento; Funzione (Attributo), ... (Operando)*

- *Funzione*: count, sum, avg, max, min

- Semantica

- Il risultato contiene la proiezione sugli attributi di raggruppamento e il valore delle funzioni in corrispondenza al sottoinsieme raggruppato
- simbolo più compatto ... γ (da GROUP BY)

Versione semplice

- Senza attributo di raggruppamento
- Calcolo sull'intera relazione

AGG *Funzione (Attributo) (Operando)*

o meglio

AGG *Funzione (Attributo) → Nome (Operando)*

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Il numero totale di esami

AGG count(*) → NumeroEsami (esami)

NumeroEsami
5

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Il numero totale di esami

AGG count(*) → NumeroEsami (esami)

NumeroEsami
5

- In Relax (con o senza parentesi)

Y ; COUNT(*) → NumeroEsami **Esami**

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Il voto medio complessivo

AGG $\text{avg}(\text{Voto}) \rightarrow \text{MediaGenerale}(\text{esami})$

MediaGenerale
27

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Il voto medio complessivo

AGG $_{avg(Voto)} \rightarrow$ MediaGenerale (esami)

MediaGenerale
27

$\Upsilon ; AVG(Voto) \rightarrow$ MediaGenerale (Esami)

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Il numero di esami e il voto medio complessivo

NumeroEsami	MediaGenerale
5	27

AGG $\text{count}(\ast) \rightarrow \text{NumeroEsami}, \text{avg}(\text{Voto}) \rightarrow \text{MediaGenerale} (\text{esami})$

$\gamma \text{ count}(\ast) \rightarrow \text{NumeroEsami}, \text{avg}(\text{Voto}) \rightarrow \text{Media} (\text{Esami})$

Versione completa

- Con attributi di raggruppamento
- Calcolo sulle partizioni

AGG *AttributiRaggruppamento; Funzione (Attributo), ... (Operando)*

o meglio

AGG *AttributiRaggruppamento; Funzione (Attributo) → Nome, ... (Operando)*

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Il voto medio per ciascuno studente

AGG Matricola; avg(Voto) → Media (esami)

Matricola	Media
3456	28
9283	27
6554	26

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Il voto medio per ciascuno studente

AGG $\text{Matricola; avg(Voto)} \rightarrow \text{Media (esami)}$

Matricola	Media
3456	28
9283	27
6554	26

Υ $\text{Matricola; AVG(Voto)} \rightarrow \text{MediaGenerale (Esami)}$

esami	Matricola	Voto	Codice
	3456	30	04
	3456	26	02
	9283	27	01
	6554	26	01
	6554	26	05

- Numero esami e voto medio per ciascuno studente

AGG Matricola; avg(Voto) → Media, count(*) → NumEsami (esami)

Matricola	Media	NumEsami
3456	28	2
9283	27	1
6554	26	2

Aggregazioni e valori nulli

- vengono trattati correttamente (cioè ignorati nelle medie, nelle somme e nei conteggi)

count

- $\text{count}(\ast)$ conta le ennuple
- $\text{count}(A)$ conta le ennuple che hanno il valore di A
 - non i valori distinti di A
 - quindi in assenza di valori il risultato è lo stesso, in presenza di valori nulli per A
 - $\text{count}(A)$ è il numero di ennuple che non hanno il valore nullo

esami

Matricola	Voto	Codice
3456	30	04
3456	NULL	02
9283	27	01
6554	26	01
6554	26	05

- Numero esami e voto medio per ciascuno studente

AGG $\text{Matricola; avg(Voto) \rightarrow Media, count(*) \rightarrow NumEsami (esami)}$

Matricola	Media	NumEsami
3456	30	2
9283	27	1
6554	26	2

- Numero voti e voto medio per ciascuno studente

AGG $\text{Matricola; avg(Voto) \rightarrow Media, count(Voto) \rightarrow NumVoti (esami)}$

Matricola	Media	NumVoti
3456	30	1
9283	27	1
6554	26	2

Viste (relazioni derivate)

- **Relazioni di base:** contenuto autonomo, le relazioni nella base di dati
- **Relazioni derivate:**
 - relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- Le relazioni derivate possono essere definite su altre derivate, ma ...



Viste, esempio

Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Direzione

Reparto	Capo
A	Mori
B	Bruni

- una vista:

Supervisione =

$\text{PROJ}_{\text{Impiegato, Capo}} (\text{Afferenza JOIN Direzione})$

Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

SEL_{Capo='Leoni'} (**Supervisione**)

viene eseguita come

SEL_{Capo='Leoni'}
PROJ_{Impiegato, Capo} (**Afferenza JOIN Direzione**)

Viste, motivazioni

Nota bene:

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

Vantaggi:

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

Viste come strumento di programmazione

- Trovare gli impiegati che hanno lo stesso capo di Rossi (vedremo meglio il concetto più avanti)
- Senza vista:

```
PROJ Impiegato ((Afferenza JOIN Direzione) JOIN  
                REN ImpR,RepR ← Imp,Reparto (  
                SEL Impiegato='Rossi' (Afferenza JOIN Direzione)))
```

- Con la vista:

```
PROJ Impiegato (Supervisione JOIN  
                REN ImpR← Imp (  
                SEL Impiegato='Rossi' (Supervisione)))
```

FINO QUI?