

ALGEBRA RELAZIONALE

Linguaggi per basi di dati

- I linguaggi per la interazione con DBMS sono una componente fondamentale per le basi dati (sino ad ora abbiamo solo fatto riferimento ad SQL):
 - operazioni sullo schema
 - DDL: **data definition language**
 - creazione tabelle, modifica attributi delle tabelle
 - specifica dei vincoli di integrità
 - operazioni sui dati
 - DML: **data manipulation language**
 - interrogazione ("query")
 - aggiornamento (inserimento, cancellazione, modifica)

Linguaggi di interrogazione per basi di dati

- In base a come si specifica una interrogazione, i Data Manipulation Languages si dividono in:
 - Dichiarativi
specificano le proprietà del risultato ("che cosa")
 - Procedurali
specificano le modalità di generazione del risultato ("come")

Linguaggi di interrogazione per il modello relazionale

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

Algebra relazionale

- Costituito da un insieme di operatori definiti su relazioni e che producono relazioni come risultati
- Gli operatori possono essere composti in cascata per formulare interrogazioni complesse

Operatori dell'algebra relazionale

- Unione, intersezione, differenza (operatori insiemistici)
- Ridenominazione, selezione, proiezione (operatori monadici, agiscono su una sola relazione)
- Join: join naturale, prodotto cartesiano, theta-join (operatori su due o più relazioni)

Operatori insiemistici

- Dato che le istanze di relazioni sono insiemi ha senso considerare gli operatori definiti per gli insiemi
- Tuttavia una relazione nel modello relazionale differisce da una generica relazione insiemistica per la presenza dello schema
- Per evitare risultati non coerenti, l'applicazione degli operatori unione, intersezione, differenza deve essere limitata a **relazioni definite sugli stessi attributi**

Unione

- L'unione tra due relazioni R_1 ed R_2 definite sullo stesso insieme di attributi X è indicata con $R_1 \cup R_2$ ed è una relazione su X contenente le n -uple che appartengono ad R_1 o ad R_2 o ad entrambe

Unione

- Siccome in una relazione non possono esserci due n-uple uguali, le n-uple che sono presenti in entrambe le relazioni di partenza vengono rappresentate una sola volta nell'unione

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

Unione

- Grado?
- Cardinalità $\#(R_1 \cup R_2)$ rispetto a $\#(R_1)$ e $\#(R_2)$?

Intersezione

- L'intersezione tra due relazioni R_1 ed R_2 definite sullo stesso insieme di attributi X è indicata con $R_1 \cap R_2$ ed è una relazione su X contenente le n -uple che appartengono sia ad R_1 che ad R_2

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

Intersezione

- Grado?
- Cardinalità?

Differenza

- La differenza tra una relazione R_1 ed una relazione R_2 definite sullo stesso insieme di attributi X è indicata con $R_1 - R_2$ ed è una relazione su X contenente le n -uple di R_1 che non appartengono ad R_2
- La differenza **NON È** commutativa

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

Differenza

- Grado?
- Cardinalità?

Operatori insiemistici e schemi

- Come applicare operatori insiemistici tra relazioni definite su schemi diversi?

Impiegati

Cognome	Ufficio	Stipendio
Rossi	Roma	55
Neri	Milano	64

Operai

Cognome	Fabbrica	Salario
Bruni	Monza	45
Verdi	Latina	55

Impiegati \cup Operai

??

Ridenominazione

- Operatore monadico (con un argomento)
- **Modifica lo schema** di una relazione cambiando il nome di uno o più attributi e lasciando inalterata l'istanza
- Data una relazione R definita su un insieme di attributi X la ridenominazione di $A \subseteq X$ in A' si indica con:

$$\text{REN}_{A' \leftarrow A}(R)$$

Impiegati

Cognome	Ufficio	Stipendio
Rossi	Roma	55
Neri	Milano	64

Operai

Cognome	Fabbrica	Salario
Bruni	Monza	45
Verdi	Latina	55

REN Sede, Retribuzione ← Ufficio, Stipendio (Impiegati)

REN Sede, Retribuzione ← Fabbrica, Salario (Operai)

Cognome	Sede	Retribuzione
Rossi	Roma	55
Neri	Milano	64
Bruni	Monza	45
Verdi	Latina	55

Ridenominazione

- Grado?
- Cardinalità?

Ridenominazione

- Vincoli:
 - Non si può assegnare ad un attributo un nome che compare già tra i nomi degli attributi della relazione

Selezione

- operatore monadico
- produce un risultato che:
 - ha lo stesso schema dell'operando
 - contiene un sottoinsieme delle ennuple dell'operando
 - quelle che soddisfano una **condizione di selezione**

Selezione

- Una condizione di selezione su un insieme di attributi X è una **formula proposizionale** su X ottenuta combinando con i connettivi logici $\text{AND}(\vee)$, $\text{OR}(\wedge)$ e $\text{NOT}(\neg)$ condizioni atomiche del tipo $A\chi B$ o $A\chi c$ dove:
 - χ è un operatore di confronto ($=, <, >, \dots$)
 - A e B sono attributi di X su cui χ abbia senso
 - c è una costante compatibile con il dominio di A

Selezione, sintassi e semantica

- sintassi

$SEL_{Condizione} (Operando)$

- *Condizione*: espressione booleana (come quelle dei vincoli di ennupla)

- semantica

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

- impiegati che guadagnano più di 50

Impiegati

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

SEL_{Stipendio > 50} (Impiegati)

- impiegati che guadagnano più di 50 e lavorano a Milano

Impiegati

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

SEL_{Stipendio > 50 AND Filiale = 'Milano'} (Impiegati)

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

Impiegati

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

SEL Cognome = Filiale(**Impiegati**)

Selezione

- Grado?
- Cardinalità?

Selezione con valori nulli

Impiegati

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

SEL $\text{Età} > 40$ (Impiegati)

- la formula proposizionale è **FALSA** per quelle n-ple in cui qualcuno degli attributi coinvolti ha valore nullo

Un risultato non desiderabile

$$\text{SEL}_{\text{Età}>30}(\text{Persone}) \cup \text{SEL}_{\text{Età}\leq 30}(\text{Persone}) \\ \neq \\ \text{Persone}$$

- Perché? Perché le selezioni vengono valutate separatamente!
- Ma anche

$$\text{SEL}_{\text{Età}>30 \text{ OR } \text{Età}\leq 30}(\text{Persone}) \neq \text{Persone}$$

- Perché? Perché anche le condizioni atomiche vengono valutate separatamente!

Selezione con valori nulli

$SEL_{Età > 40}$ (Impiegati)

- la condizione atomica è vera solo per valori non nulli
- per riferirsi ai valori nulli esistono forme apposite di condizioni:

IS NULL

IS NOT NULL

Selezione con valori nulli

- A questo punto:

$$\text{SEL}_{\text{Età} > 30}(\text{Persone}) \cup \text{SEL}_{\text{Età} \leq 30}(\text{Persone}) \cup \text{SEL}_{\text{Età IS NULL}}(\text{Persone})$$

=

$$\text{SEL}_{\text{Età} > 30 \text{ OR Et\`a} \leq 30 \text{ OR Et\`a IS NULL}}(\text{Persone})$$

=

Persone

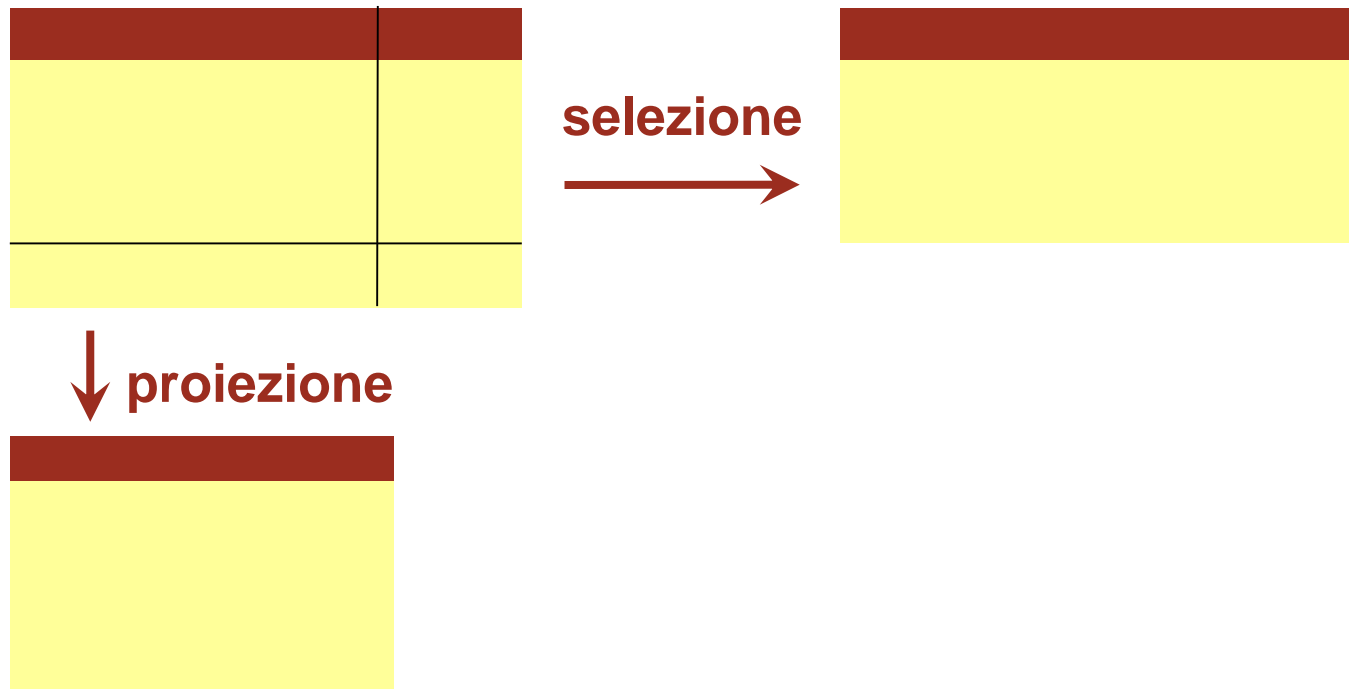
Impiegati

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

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

Selezione e proiezione

- operatori "ortogonali"
- **selezione**: decomposizione orizzontale
- **proiezione**: decomposizione verticale



Proiezione

- operatore monadico
- produce un risultato che
 - È definito su un sottoinsieme Y degli attributi dell'operando
 - contiene n -uple ottenute considerando i valori su Y delle n -uple dell'operando

Proiezione, sintassi e semantica

- Data una relazione $R(X)$ ed un insieme $Y \subset X$ la proiezione di R su Y si indica con $\text{PROJ}_Y(R)$ ed è una relazione R' definita su Y e contenente le n -ple di R considerandone solo i valori su Y :

$$\text{PROJ}_Y(R) = \{t[Y] \mid t \in R\}$$

Impiegati

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

- per tutti gli impiegati:
 - matricola e cognome
 - cognome e filiale

- matricola e cognome di tutti gli impiegati

Matricola	Cognome
7309	Neri
5998	Neri
9553	Rossi
5698	Rossi

PROJ Matricola, Cognome **(Impiegati)**

Proiezione

- Grado?
- Cardinalità?

– cognome e filiale di tutti gli impiegati

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma

PROJ Cognome, Filiale **(Impiegati)**

Cardinalità delle proiezioni

- una proiezione
 - contiene al più tante ennuple quante l'operando
 - può contenerne di meno
- se X è una superchiave di R , allora...
 $PROJ_X(R)$ contiene
esattamente tante ennuple quante R

Selezione e proiezione

- Gli operatori di selezione e proiezione possono essere combinati...

- 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))

**L'ordine dei due operatori
non sempre è commutabile**

Join

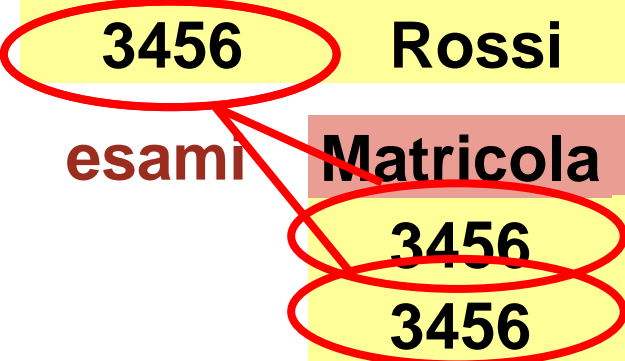
- Combinando selezione e proiezione, possiamo estrarre informazioni da una relazione
- Non possiamo però correlare informazioni presenti in relazioni diverse
- Il JOIN è l'operatore dell'algebra relazionale che permette di correlare dati in relazioni diverse

Join

- Quali esami ha superato Maria Rossi e con quale votazione?

studenti	Matricola	Cognome	Nome	Data di nascita
	6554	Rossi	Mario	05/12/1978
	8765	Neri	Paolo	03/11/1976
	9283	Verdi	Luisa	12/11/1979
	3456	Rossi	Maria	01/02/1978

esami	Matricola	Voto	Corso
	3456	30	Analisi I
	3456	24	Chimica
	9283	28	Analisi I
	6554	26	Fisica



Join naturale

- operatore binario (generalizzabile)
- produce una relazione definita sull'**unione** degli attributi degli operandi (non concatenazione)
- le righe del risultato sono ottenute combinando le tuple degli operandi sulla base di valori uguali in attributi con lo stesso nome

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Mori
B	Bruni

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

- le righe del risultato sono ottenute combinando le tuple degli operandi sulla base di valori uguali in attributi con lo stesso nome

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Mori
B	Bruni

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

- Per capire la definizione formale del JOIN è utile osservare che ogni riga del risultato è costituita dall'unione di due parti a ciascuna delle quali corrisponde una riga nelle tabelle di ingresso

Join, sintassi e semantica

- $R_1(X_1), R_2(X_2)$
- $R_1 \text{ JOIN } R_2$ è una relazione su $X_1 \cup X_2$

$$R_1 \text{ JOIN } R_2 = \{ t \text{ su } X_1 \cup X_2 \mid \exists t_1 \in R_1 \text{ e } t_2 \in R_2 \\ \text{con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$$

JOIN Completo

- Se ogni n-pla delle due relazioni contribuisce al risultato il JOIN si dice **completo**
- Altrimenti il join non è completo

Un join non completo

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

JOIN vuoto

- Un caso particolare del JOIN non completo è il JOIN vuoto che dà come risultato la relazione vuota

Un join vuoto

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

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

JOIN completo

- Il JOIN è completo se ogni n-upla delle due relazioni si combina con **ALMENO UNA** n-upla dell'altra relazione
- Un caso particolare di JOIN completo è quello in cui ogni n-upla delle due relazioni si combina con **TUTTE** le n-uple dell'altra relazione

Un join completo, con $n \times m$ ennuple

Impiegato	Reparto
Rossi	B
Neri	B

Reparto	Capo
B	Mori
B	Bruni

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

Cardinalità del JOIN

- Il join di R_1 e R_2 contiene un numero di n-ple compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$
- Il fatto che il JOIN sia completo non significa che $|R_1 \text{ JOIN } R_2| = |R_1| * |R_2|$

Cardinalità del JOIN

- se gli attributi in comune X sono una chiave di R_2 , allora la cardinalità del JOIN...
- se gli attributi in comune X sono una chiave di R_2 ed esiste tra loro un vincolo di integrità referenziale, allora la cardinalità del JOIN...

[illegible]

R_2 <u>X</u>			

Cardinalità del join, esempi

$$R_1(X_1) \text{ JOIN } R_2(X_2) \quad B = X_1 \cap X_2$$

- in generale

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

- se B è chiave in R_2

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

- se B è chiave in R_2 ed esiste vincolo di integrità referenziale fra B in R_1 e R_2 :

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

Proprietà del JOIN

- Il JOIN naturale soddisfa le proprietà:

- Commutativa:

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

- Associativa:

$$R_1 \text{ JOIN } R_2 \text{ JOIN } R_3 = (R_1 \text{ JOIN } R_2) \text{ JOIN } R_3 = R_1 \text{ JOIN } (R_2 \text{ JOIN } R_3)$$

Riassunto

- Vincolo di integrità referenziale
- Operazioni di aggiornamento e violazione dei vincoli
- Linguaggi per l'interazione con DBMS
 - DDL, DML
 - Linguaggi dichiarativi e procedurali
 - Algebra relazionale: unione, intersezione, differenza, ridenominazione, selezione, proiezione, join naturale, completezza del join

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

- Se il JOIN non è completo alcune n-ple non contribuiscono al risultato: l'informazione nel risultato non contiene alcune delle informazioni negli operandi

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

- **sinistro**: mantiene tutte le ennuple del primo operando, estendendole con valori nulli, se necessario
- **destro**: ... del secondo operando ...
- **completo**: ... di entrambi gli operandi ...

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{LEFT} Reparti

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

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{RIGHT} Reparti

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

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{FULL} Reparti

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

Cardinalità JOIN esterno

- La cardinalità **minima**, da 0 diventa:
 - $|R_1|$ per il LEFT join
 - $|R_2|$ per il RIGHT join
 - $\text{MAX}(|R_1|, |R_2|)$ per il FULL join

JOIN e proiezioni

- Date due relazioni $R_1(X_1)$ ed $R_2(X_2)$ si ha che

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

- Infatti, nel caso che il JOIN non sia completo, al risultato potrebbe partecipare solo un sottoinsieme delle n-ple di R_1 .

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

- Data una relazione $R(X)$ con $X = X_1 \cup X_2$ si ha che:

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

- Si ha il segno = solo nel caso in cui ...
 $X_1 \cap X_2$ è una superchiave per R

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

Join e proiezioni

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

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

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

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

Un caso limite...

- Che succede se eseguo un join naturale tra relazioni che non hanno attributi in comune?
 - Ogni n-pla della prima relazione può combinarsi con ognuna delle n-ple della seconda
 - Il risultato contiene sempre un numero di n-ple pari al prodotto delle cardinalità degli operandi
- In questo caso il JOIN diventa il prodotto cartesiano tra i due operandi

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 solo se seguito da selezione:

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

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

$$R_1 JOIN_{Condizione} R_2$$

Perché "theta-join"?

- La condizione C è spesso una combinazione (**AND**) di condizioni atomiche di confronto $A_1 \vartheta A_2$ dove ϑ è uno degli operatori di confronto ($=, >, <, \dots$)
- se l'operatore è sempre l'uguaglianza ($=$) allora si parla di **equi-join**

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

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
A	Mori
B	Bruni

Impiegati JOIN Reparti

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

JOIN naturale ed equi-join

- Il JOIN naturale può essere ottenuto per mezzo di operatori di ridenominazione, equi-join e proiezione

Join naturale ed equi-join

Impiegati

Impiegato Reparto

Reparti

Reparto Capo

Impiegati JOIN_{nat} Reparti =

PROJ_{Impiegato,Reparto,Capo} (

Impiegati JOIN_{Reparto=Codice} REN_{Codice ← Reparto} (Reparti)
)

JOIN e vincoli di integrità referenziale

- Il vincolo di IR esplicita un legame tra gli attributi di due tabelle diverse.
 - Livello sintattico: valori della tabella interna sono vincolati a comparire nella tabella esterna
 - Livello semantico: equi join tra attributi interessati a vincolo IR esplicita un'informazione non presente né in R_1 né in R_2 se prese separatamente

JOIN e vincoli di integrità referenziale

- Esempio: Lo schema sotto riportato rappresenta informazioni sui dipendenti di un'azienda. Ogni dipendente è affiliato ad un dipartimento. Ogni dipartimento ha un direttore che è uno dei dipendenti

Dipendente(matricola, nome, cognome, **dipartimento**, stipendio)

Dipartimento(codice, descrizione, **direttore**)

Dipendente				
matricola	nome	cognome	dipartimento	stipendio
015234	Carlo	Minetti	D002	20000
034727	Giovanni	Pirli	D002	25000
026777	Franca	Franchi	D001	30000

Dipartimento		
codice	descrizione	direttore
D001	Ricerca	026777
D002	Sviluppo	034727

JOIN e vincoli di integrità referenziale

- Esempio: Lo schema sotto riportato rappresenta informazioni sui dipendenti di un'azienda. Ogni dipendente è affiliato ad un dipartimento. Ogni dipartimento ha un direttore che è uno dei dipendenti

Dipendente(matricola, nome, cognome, **dipartimento**, stipendio)

Dipartimento(codice, descrizione, **direttore**)

- Quale informazione è resa esplicita da
Dipendente JOIN_{dipartimento=codice} Dipartimento
- Quale informazione è resa esplicita da
Dipendente JOIN_{direttore=matricola} Dipartimento

Esempi

Impiegati

<u>Matricola</u>	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

Esempi

- Matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni
- Matricola, nome ed età degli impiegati che guadagnano più di 40 milioni
- Matricole dei capi degli impiegati che guadagnano più di 40 milioni
- Nome e stipendio dei capi degli impiegati che guadagnano più di 40 milioni
- Impiegati che guadagnano più del proprio capo, mostrando matricola, nome e stipendio dell'impiegato e del capo
- Matricole dei capi i cui impiegati guadagnano **tutti** più di 40 milioni

Supervisione

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

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

- Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni

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

- Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni

$SEL_{\text{Stipendio} > 40}(\text{Impiegati})$

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

SEL_{Stipendio>40}(Impiegati)

- Trovare matricola, nome ed età degli impiegati che guadagnano più di 40 milioni

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

- 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

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

- Trovare le matricole dei capi di quegli impiegati che guadagnano più di 40 milioni

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

- Trovare nome e stipendio dei capi di quegli impiegati che guadagnano più di 40 milioni

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

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

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

- Trovare le matricole dei capi i cui impiegati guadagnano **tutti** più di 40 milioni

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

Esercizi

- Estrarre Nome e Cognome degli studenti lavoratori
- Nome e Cognome degli studenti NON lavoratori
- Nome e cognome degli studenti lavoratori che hanno sostenuto almeno un esame con più di 4 crediti
- Nome e Cognome degli studenti che hanno preso almeno un trenta e mai un voto inferiore a 27
- Nome e Cognome degli studenti che hanno dato Analisi I e non Geometria
- Nome e Cognome degli studenti che hanno dato almeno due esami
- Nome e Cognome degli studenti che hanno dato almeno due esami con lo stesso docente

Studenti					StudentiLavoratori			
Corsi	<u>Matricola</u>	Nome	Cognome	Eta	<u>Matricola</u>			
	<u>Codice</u>	Nome	Crediti	Docente				
Esami								
					<u>Corso</u>	<u>Studente</u>	Voto	Data

Esercizi

- Estrarre Nome e Cognome degli studenti lavoratori

```
PROJnome,cognome(  
  RENmat←matricola(StudentiLav)  
  JOINmat=matricola(Studenti))
```

Studenti

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>	<u>Eta</u>
------------------	-------------	----------------	------------

Corsi

<u>Codice</u>	<u>Nome</u>	<u>Crediti</u>	<u>Docente</u>
---------------	-------------	----------------	----------------

StudentiLavoratori

<u>Matricola</u>

Esami

<u>Corso</u>	<u>Studente</u>	<u>Voto</u>	<u>Data</u>
--------------	-----------------	-------------	-------------

Esercizi

- Nome e Cognome degli studenti NON lavoratori

```
PROJnome,cognome(  
RENmat←matricola(PROJmatricola(Studenti) - StudentiLav)  
JOINmat=matricola(Studenti))
```

```
PROJnome,cognome(  
(PROJmatricola(Studenti) - StudentiLav)  
JOIN Studenti)
```

Studenti

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>	<u>Eta</u>
------------------	-------------	----------------	------------

StudentiLavoratori

<u>Matricola</u>

Corsi

<u>Codice</u>	<u>Nome</u>	<u>Crediti</u>	<u>Docente</u>
---------------	-------------	----------------	----------------

Esami

<u>Corso</u>	<u>Studente</u>	<u>Voto</u>	<u>Data</u>
--------------	-----------------	-------------	-------------

Esercizi

- Nome e cognome degli studenti lavoratori che hanno sostenuto almeno un esame con più di 4 crediti

```
PROJnome,cognome(  
(PROJstudente(Esami JOINcorso=codice AND crediti>4 Corsi))  
JOINstudente=matricola(Studenti JOIN StudentiLav))
```

Studenti

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>	<u>Eta</u>
------------------	-------------	----------------	------------

Corsi

<u>Codice</u>	<u>Nome</u>	<u>Crediti</u>	<u>Docente</u>
---------------	-------------	----------------	----------------

StudentiLavoratori

<u>Matricola</u>

Esami

<u>Corso</u>	<u>Studente</u>	<u>Voto</u>	<u>Data</u>
--------------	-----------------	-------------	-------------

Esercizi

- Nome e Cognome degli studenti che hanno preso almeno un trenta e mai un voto inferiore a 27

```
PROJnome,cognome( (Studenti)JOINstudente=matricola  
( PROJstudente(SELvoto=30 (Esami) ) –  
  PROJstudente(SELvoto<27 (Esami) ) ) )
```

Studenti

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>	<u>Eta</u>
------------------	-------------	----------------	------------

Corsi

<u>Codice</u>	<u>Nome</u>	<u>Crediti</u>	<u>Docente</u>
---------------	-------------	----------------	----------------

StudentiLavoratori

<u>Matricola</u>

Esami

<u>Corso</u>	<u>Studente</u>	<u>Voto</u>	<u>Data</u>
--------------	-----------------	-------------	-------------

Esercizi

- Nome e Cognome degli studenti che hanno dato Analisi I e non Geometria

```
PROJnome,cognome( (Studenti) JOINstudente=matricola  
( PROJstudente(Esami JOINcorso=codice AND nome='Analisi I' Corsi) –  
PROJstudente(Esami JOINcorso=codice AND nome='Geometria' Corsi) ))
```

Studenti

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>	<u>Eta</u>
------------------	-------------	----------------	------------

Corsi

<u>Codice</u>	<u>Nome</u>	<u>Crediti</u>	<u>Docente</u>
---------------	-------------	----------------	----------------

StudentiLavoratori

<u>Matricola</u>

Esami

<u>Corso</u>	<u>Studente</u>	<u>Voto</u>	<u>Data</u>
--------------	-----------------	-------------	-------------

Esercizi

- Nome e Cognome degli studenti che hanno dato almeno due esami

```
PROJnome,cognome( (Studenti) JOINstudente=matricola  
RENstud2,corso2←studente,corso(PROJstudente,corso (Esami))  
JOINstud2 = studente AND corso2<>corso(PROJstudente,corso (Esami) ))
```

Studenti

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>	<u>Eta</u>
------------------	-------------	----------------	------------

Corsi

<u>Codice</u>	<u>Nome</u>	<u>Crediti</u>	<u>Docente</u>
---------------	-------------	----------------	----------------

StudentiLavoratori

<u>Matricola</u>

Esami

<u>Corso</u>	<u>Studente</u>	<u>Voto</u>	<u>Data</u>
--------------	-----------------	-------------	-------------

Esercizi

- Nome e Cognome degli studenti che hanno dato almeno due esami con lo stesso docente

Studenti

<u>Matricola</u>	<u>Nome</u>	<u>Cognome</u>	<u>Eta</u>
------------------	-------------	----------------	------------

StudentiLavoratori

<u>Matricola</u>

Corsi

<u>Codice</u>	<u>Nome</u>	<u>Crediti</u>	<u>Docente</u>
---------------	-------------	----------------	----------------

Esami

<u>Corso</u>	<u>Studente</u>	<u>Voto</u>	<u>Data</u>
--------------	-----------------	-------------	-------------