



BASI DI DATI

ALGEBRA RELAZIONALE

Polese G. Caruccio L. Breve B.

a.a. 2023/2024

Richiami di Linguaggi per Basi di Dati

- Operazioni sullo schema:
 - DDL: Data Definition Language
- Operazioni sui dati:
 - DML: Data Manipulation Language
 - ✓ Interrogazione (query)
 - ✓ Aggiornamento (Inserimento, Modifica, Cancellazione)

Linguaggi di interrogazione

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

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")

Algebra Relazionale

- Insieme di operatori:
 - Su relazioni
 - Producono come risultato una relazione
 - Possono essere composti
- Gli operatori dell'algebra relazionale sono:
 - Unione, intersezione e differenza
 - Ridenominazione
 - Selezione
 - Proiezione
 - Join (naturale, prodotto cartesiano, theta-join)

Operatori insiemistici

- Le relazioni sono insiemi
- I risultati devono essere relazioni
- È possibile applicare **unione**, **intersezione** e **differenza** alle relazioni, ma solo se non definite sugli stessi attributi (proprietà di **union compatibility**).

Unione

Laureati

Matricola	Nome	Età
7274	Rossi	42
9297	Neri	33
9826	Verdi	45

Quadri

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9826	Verdi	45

Laureati \cup Quadri

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9826	Verdi	45
9297	Neri	33

Intersezione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9826	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9826	Verdi	45

Intersezione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9826	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9826	Verdi	45

Laureati \cap Quadri

Matricola	Nome	Età
7432	Neri	54
9826	Verdi	45

Differenza

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9826	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9826	Verdi	45

Differenza

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9826	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9826	Verdi	45

Laureati - Quadri

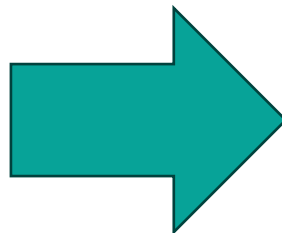
Matricola	Nome	Età
7274	Rossi	42

Ridenominazione

- Operatore monadico (con un argomento)
- "Modifica lo schema" lasciando inalterata l'istanza dell'operando

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco



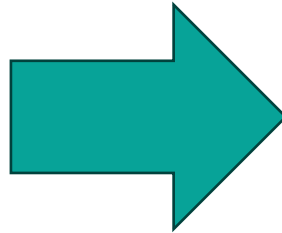
REN Genitore ← Padre (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Ridenominazione per uniformare schemi

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

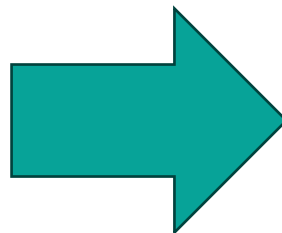


REN Genitore ← Padre (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco



REN Genitore ← Madre (Maternità)

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

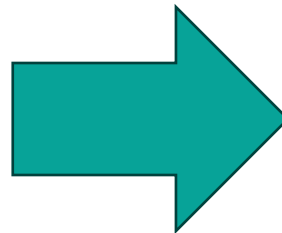
Esempio di unione

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco



REN Genitore ← Padre (**Paternità**)

U

REN Genitore ← Madre (**Maternità**)

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

Esempio di unione

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)**

U

REN Sede, Retribuzione ← Fabbrica, Salario **(Operai)**

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

Selezione

- Operatore monadico. Produce un risultato che:
 - Ha lo stesso schema dell'operando
 - Contiene un sottoinsieme delle ennuple (tuple) dell'operando, cioè quelle che soddisfano una condizione

- Impiegati che:

Matricola	Cognome	Sede	Retribuzione
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Neri	Napoli	64

- Guadagnano più di 50
- Guadagnano più di 50 e lavorano a Milano
- Hanno lo stesso nome della filiale presso cui lavorano

Selezione, sintassi e semantica

- Sintassi:
 - **SEL** (condizione) (Operando), oppure
 - **σ** condizione (Operando)
- *Condizione* : espressione booleana (come quelle dei vincoli di ennupla)
- Semantica
 - Produce una relazione contenente le ennuple dell'operando che soddisfano la condizione

Esempio di selezione (1/3)

- Impiegati che guadagnano più di 50

Impiegati

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

Esempio di selezione (1/3)

- 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), oppure
- `σ stipendio > 50` (Impiegati)

Esempio di selezione (2/3)

- Impiegati che guadagnano più di 50 e lavorano a Milano

Impiegati

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

Esempio di selezione (2/3)

- 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), oppure
- **σ** stipendio > 50 AND Filiale = "Milano" (Impiegati)

Esempio di selezione (3/3)

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

Impiegati

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

Esempio di selezione (3/3)

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

Impiegati

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

- **SEL** Cognome = Filiale (Impiegati), oppure
- **σ** Cognome = Filiale (Impiegati)

Proiezione

- Operatore monadico. Produce un risultato che:
 - Ha parte degli attributi dell'operando
 - Contiene ennuple cui contribuiscono tutte le ennuple dell'operando
 - In pratica, vengono eliminati eventuali duplicati delle ennuple, indotti dalla eliminazione di alcuni attributi

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

Proiezione, sintassi e semantica

- Sintassi:

- $\text{PROJ}_{\text{ListaAttributi}}$ (Operando), oppure
- $\Pi_{\text{ListaAttributi}}$ (Operando)

- Semantica

- Il risultato contiene le ennuple dell'operando ristrette agli attributi nella lista

Esempio di proiezione (1/2)

- Matricola e cognome di tutti gli impiegati

Impiegati

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

Esempio di proiezione (1/2)

- Matricola e cognome di tutti gli impiegati

Impiegati

Matricola	Cognome
7309	Rossi
5998	Neri
5698	Neri

- **PROJ** *Matricola, Cognome* (Impiegati), oppure
- **II** *Matricola, Cognome* (Impiegati)

Esempio di proiezione (2/2)

- Cognome e filiale di tutti gli impiegati

Impiegati

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

Esempio di proiezione (2/2)

- Cognome e filiale di tutti gli impiegati

Impiegati

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma

- **PROJ** *Cognome, filiale* (Impiegati), oppure
- **II** *Cognome, filiale* (Impiegati)

Cardinalità delle proiezioni

- Una proiezione
 - Contiene al più tante ennuple quante ne contiene l'operando
 - Non può contenerne di meno
- Se **X** è una superchiave di **R**, allora **PROJ_X(R)** contiene esattamente tante ennuple quante ne contiene **R**

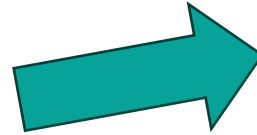
Selezione e proiezione

- Combinando selezione e proiezione possiamo estrarre interessanti informazioni da una relazione
- Sono gli operatori "ortogonali"
 - **Selezione**: decomposizione orizzontale
 - **Proiezione**: decomposizione verticale

Selezione e proiezione in pratica

ATTR1	ATTR2	ATTR3
...
...
...
...
...

Selezione



ATTR1	ATTR2	ATTR3
...
...
...



Proiezione

ATTR1	ATTR3
...	...
...	...
...	...
...	...
...	...

Esempio di selezione e proiezione

- Matricola e cognome degli impiegati che guadagnano più di 50

Impiegati

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

Esempio di selezione e proiezione

- Matricola e cognome degli impiegati che guadagnano più di 50

Impiegati

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

SEL stipendio > 50 (Impiegati)

Esempio di selezione e proiezione

- Matricola e cognome degli impiegati che guadagnano più di 50

Impiegati

Matricola	Cognome
7309	Rossi
5998	Neri
5698	Neri

PROJ matricola, Cognome (**SEL** stipendio > 50 (Impiegati))

Combinare informazioni da più relazioni

- Sebbene la combinazione di selezione e proiezione consenta di estrarre informazioni interessanti da una relazione, essa non consente di correlare informazioni presenti in relazioni diverse
- Occorre un operatore specifico: il JOIN
- L'operatore JOIN permette di correlare dati in relazioni diverse
- È l'operatore più interessante dell'algebra relazionale

Esempio sull'uso del JOIN

- Si consideri il problema della gestione di prove scritte in un concorso pubblico:
 - I compiti sono anonimi, ad ognuno è associata una busta chiusa che deve contenere il nome del candidato
 - Ciascun compito e la relativa busta vengono contrassegnati con uno stesso numero

Esempio sull'uso del JOIN

Risultati

Numero	Voto
1	25
2	13
3	27
4	28

Concorrenti

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

Esempio sull'uso del JOIN

Risultati

Numero	Voto
1	25
2	13
3	27
4	28

Concorrenti

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

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
 - Sull'unione degli attributi degli operandi
 - Con ciascuna ennupla costruita a partire da una ennupla di ognuno degli operandi
 - In pratica, le sue ennuple sono il risultato della combinazione di ennuple degli operandi con valori coincidenti su attributi comuni

JOIN, sintassi e semantica

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

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

Esempio di JOIN

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Mori
B	Bruni

Impiegato	Reparto	Capo
Rossi	A	Mori

Esempio di JOIN

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Mori
B	Bruni

Impiegato	Reparto	Capo
Rossi	A	Mori
Neri	B	Bruni

Esempio di JOIN

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

- Ogni ennupla contribuisce al risultato: **Join 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

- A e C non hanno combinazioni

Un Join vuoto

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
D	Mori
C	Bruni

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

Un Join completo, con n x m ennuple

Impiegato	Reparto
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
B	Bruni

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

Cardinalità del join (1)

- Il join di R_1 e R_2 contiene un numero di ennuple compreso tra zero ed il prodotto $|R_1|$ e $|R_2|$
- Se il join coinvolge una chiave di R_2 , allora il numero di ennuple è compreso fra zero e $|R_1|$
- Se il join coinvolge una chiave di R_2 ed una chiave esterna di R_1 , con vincolo di integrità referenziale sulle tuple di R_2 , allora il numero di ennuple è pari a $|R_1|$

Cardinalità del join (2)

- $R_1(A,B)$, $R_2(B,C)$
- 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 un vincolo di integrità referenziale fra B (in R_1) e R_2 :
 - $|R_1 \text{ JOIN } R_2| = |R_1|$

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 (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 se necessario, con valori nulli
 - **Destro**: ... del secondo operando ...
 - **Completo**: ... di entrambi gli operandi ...

Join esterno sinistro

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegati **JOIN** left Reparti

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

Join esterno destro

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegati **JOIN** right Reparti

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

Join esterno completo

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegati **JOIN** right Reparti

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

Join e proiezioni

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegati **JOIN** Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Join e proiezioni

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

PROJ Impiegato, Reparto(Impiegati JOIN Reparti)

Impiegato	Reparto
Neri	B
Bianchi	B

Proiezioni e join

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

PROJ Impiegato, Reparto(Impiegati)

Impiegato	Reparto
Verdi	A
Neri	B
Bianchi	B

PROJ Reparto, Capo(Reparti)

Reparto	Capo
A	Bini
B	Mori
B	Bruni

Proiezioni e join

Impiegato	Reparto
Verdi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Bini
B	Mori
B	Bruni

(PROJ Impiegato, Reparto(Impiegati)) JOIN (PROJ Reparto, Capo(Reparti))

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

Join e proiezioni

- $R_1(X_1), R_2(X_2)$
 - $\text{PROJ}_{X_1}(R_1 \text{ JOIN } R_2) \subseteq R_1$
- $R(X), X = X_1 \cup X_2$
 - $(\text{PROJ}_{X_1}(R)) \text{ JOIN } (\text{PROJ}_{X_2}(R)) \supseteq R$

Prodotto cartesiano

- Quando si collegano le ennuple su relazioni senza attributi comuni e la condizione di join è sempre soddisfatta.
 - Contiene tutte le possibili combinazioni di ennuple
- Contiene, quindi, un numero di ennuple pari al prodotto delle cardinalità degli operandi
 - Operando1 con cardinalità m
 - Operando2 con cardinalità n
 - Cardinalità prodotto cartesiano: $m \times n$

Esempio prodotto cartesiano

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Codice	Capo
A	Mori
B	Bruni

Impiegati X 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

Selezione su Prodotto Cartesiano

- Il prodotto cartesiano, in pratica, ha senso (quasi) solo se seguito da una selezione
 - $SEL_{Condizione} (R_1 \times R_2)$
- L'operazione viene chiamata **Theta-join** ed è indicata con:
 - $R_1 JOIN_{Condizione} R_2$

Perchè "Theta-join"?

- La condizione C è spesso una congiunzione (AND) di atomi di confronto $A1 \ \theta \ A2$ dove θ è uno degli operatori di confronto ($=, >, <, \dots$)
- Se l'operatore è sempre l'uguaglianza ($=$), allora si parla di **Equi-join**

Esempio Theta-join

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Codice	Capo
A	Mori
B	Bruni

Impiegati X 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

Esempio Theta-join

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Codice	Capo
A	Mori
B	Bruni

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



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

Impiegati

Impiegato	Reparto
-----------	---------

Reparti

Reparto	Capo
---------	------

Impiegati JOIN Reparti

Join naturale ed equi-join

Impiegati

Impiegato	Reparto
-----------	---------

Reparti

Reparto	Capo
---------	------

Impiegati JOIN Reparti

$(\text{Impiegati} \times \text{REN}_{\text{codice}} \leftarrow \text{reparto} (\text{Reparti}))$

Join naturale ed equi-join

Impiegati

Impiegato	Reparto
-----------	---------

Reparti

Reparto	Capo
---------	------

Impiegati JOIN Reparti

$(\text{SEL}_{\text{reparto}=\text{codice}}(\text{Impiegati} \times \text{REN}_{\text{codice} \leftarrow \text{reparto}}(\text{Reparti})))$

Join naturale ed equi-join

Impiegati

Impiegato	Reparto
-----------	---------

Reparti

Reparto	Capo
---------	------

Impiegati JOIN Reparti

$\text{PROJ}_{\text{impiegato, Reparto, Capo}}((\text{SEL}_{\text{reparto=codice}}(\text{Impiegati} \times \text{REN}_{\text{codice} \leftarrow \text{reparto}}(\text{Reparti})))$

Esempi

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

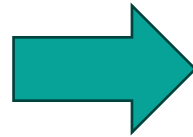
Impiegato	Capo
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

Esempi

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

$\text{PROJ}_{\text{matricola, nome, età}}(\text{SEL}_{\text{stipendio} > 40}(\text{Impiegati}))$

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



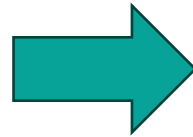
Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

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

Esempi

- Trovare matricola, nome, età degli impiegati che guadagnano più di 40
 - $\text{PROJ}_{\text{matricola, nome, età}}(\text{SEL}_{\text{stipendio} > 40}(\text{Impiegati}))$

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



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

Esempi

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

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

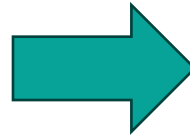
Esempi

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

$\text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato=Matricola}}(\text{SEL}_{\text{stipendio}>40}(\text{Impiegati})))$

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



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

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

Esempi

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

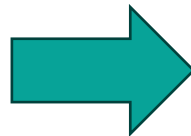
$\text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato=Matricola}}(\text{SEL}_{\text{stipendio}>40}(\text{Impiegati})))$

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

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

Supervisione

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



Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123

$\text{Supervisione JOIN}_{\text{impiegato=Matricola}}(\text{SEL}_{\text{stipendio}>40}(\text{Impiegati}))$

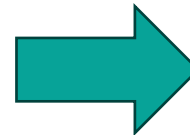
Esempi

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

PROJ_{Capo}(Supervisione JOIN_{impiegato=Matricola} (SEL_{stipendio>40}(Impiegati)))

Supervisione JOIN_{impiegato=Matricola} (SEL_{stipendio>40}(Impiegati))

Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123



Capo
5698
4076
8123

**PROJ_{Capo}(Supervisione
JOIN_{impiegato=Matricola} (SEL_{stipendio>40}(Impiegati)))**

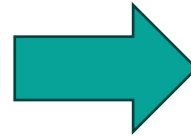
Esempi

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

$\text{PROJ}_{\text{Nome, Stipendio}}(\text{Impiegati JOIN}_{\text{Matricola=Capo}} \text{PROJ}_{\text{Capo}}(\text{Supervisione} \\ \text{JOIN}_{\text{impiegato=Matricola}} (\text{SEL}_{\text{stipendio}>40}(\text{Impiegati}))))$

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



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

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

Esempi

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

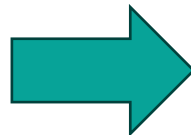
$\text{PROJ}_{\text{Nome, Stipendio}}(\text{Impiegati JOIN}_{\text{Matricola=Capo}} \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato=Matricola}} (\text{SEL}_{\text{stipendio}>40}(\text{Impiegati}))))$

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

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

Supervisione

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



Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123

$\text{Supervisione JOIN}_{\text{impiegato=Matricola}} (\text{SEL}_{\text{stipendio}>40}(\text{Impiegati}))$

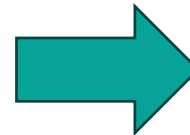
Esempi

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

$\text{PROJ}_{\text{Nome, Stipendio}}(\text{Impiegati JOIN}_{\text{Matricola=Capo}} \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato=Matricola}} (\text{SEL}_{\text{stipendio}>40}(\text{Impiegati}))))$

Supervisione JOIN_{impiegato=Matricola} (SEL_{stipendio>40}(Impiegati))

Matricola	Nome	Età	Stipendio	Impiegato	Capo
7309	Rossi	34	45	7309	5698
5698	Bruni	43	42	5698	4076
4076	Mori	45	50	4076	8123



Capo
5698
4076
8123

$\text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato=Matricola}} (\text{SEL}_{\text{stipendio}>40}(\text{Impiegati})))$

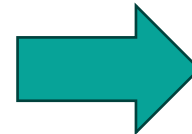
Esempi

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

`PROJNome,Stipendio(Impiegati JOINMatricola=Capo PROJCapo(Supervisione
JOINimpiegato=Matricola (SELstipendio>40(Impiegati))))`

Capo
5698
4076
8123

`PROJCapo(Supervisione
JOINimpiegato=Matricola (SELstipendio>40(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

Impiegati

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

`Impiegati JOINMatricola=Capo PROJCapo(Supervisione
JOINimpiegato=Matricola (SELstipendio>40(Impiegati)))`

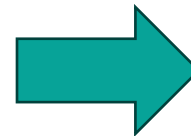
Esempi

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

PROJNome,Stipendio(**Impiegati JOIN**Matricola=Capo **PROJ**Capo(**Supervisione JOIN**impiegato=Matricola (**SEL**stipendio>40(**Impiegati**))))

Impiegati JOINMatricola=Capo **PROJ**Capo(**Supervisione JOIN**impiegato=Matricola (**SEL**stipendio>40(**Impiegati**)))

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



Nome	Stipendio
Bruni	42
Mori	50
Lupi	60

PROJNome,Stipendio(**Impiegati JOIN**Matricola=Capo **PROJ**Capo(**Supervisione JOIN**impiegato=Matricola (**SEL**stipendio>40(**Impiegati**))))

Esempi

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

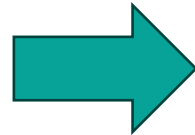
PROJMatricola, Nome, Stipendio, MatrC, NomeC, StipC(**SEL**stipendio>stipC(**REN**MatrC, NomeC, StipC, EtàC
←Matricola, Nome, Stipendio, Età(**Impiegati**) **JOIN**MatrC=Capo
(**Supervisione JOIN**impiegato=matricola **Impiegati**)))

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



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

Supervisione **JOIN**impiegato=Matricola **Impiegati**

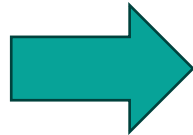
Esempi

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

```
PROJMatricola, Nome, Stipendio, MatrC, NomeC, StipC (SELstipendio > stipC (RENMatrC, NomeC, StipC, EtàC  
← Matricola, Nome, Stipendio, Età (Impiegati) JOINMatrC = Capo  
(Supervisione JOINimpiegato = matricola 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



MatrC	NomeC	EtàC	StipC
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

```
RENMatrC, NomeC, StipC, EtàC ← Matricola, Nome, Stipendio, Età (Impiegati)
```

Esempi

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

$\text{PROJ}_{\text{Matricola, Nome, Stipendio, MatrC, NomeC, StipC}}(\text{SEL}_{\text{stipendio} > \text{stipC}}(\text{REN}_{\text{MatrC, NomeC, StipC, EtàC}} \leftarrow \text{Matricola, Nome, Stipendio, Età}(\text{Impiegati}) \text{ JOIN}_{\text{MatrC=Capo}} (\text{Supervisione JOIN}_{\text{impiegato=matricola}} \text{ Impiegati})))$

Supervisione JOIN_{impiegato=Matricola} Impiegati

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



$(\text{REN}_{\text{MatrC, NomeC, StipC, EtàC}} \leftarrow \text{Matricola, Nome, Stipendio, Età}(\text{Impiegati}) \text{ JOIN}_{\text{MatrC=Capo}} (\text{Supervisione JOIN}_{\text{impiegato=matricola}} \text{ Impiegati}))$

REN_{MatrC, NomeC, StipC, EtàC} ← Matricola, Nome, Stipendio, Età(Impiegati)

MatrC	NomeC	EtàC	StipC
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

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

Esempi

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

```
PROJMatricola, Nome, Stipendio, MatrC, NomeC, StipC (SELstipendio > stipC (RENMatrC, NomeC, StipC, EtàC  
← Matricola, Nome, Stipendio, Età (Impiegati) JOINMatrC=Capo  
(Supervisione JOINimpiegato=matricola Impiegati)))
```

```
(RENMatrC, NomeC, StipC, EtàC ← Matricola, Nome, Stipendio, Età (Impiegati) JOINMatrC=Capo  
(Supervisione JOINimpiegato=matricola Impiegati))
```

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



MatrC	NomeC	EtàC	StipC	Matricola	Nome	Età	Stipendio	Impiegato	Capo
5698	Bruni	43	42	7309	Rossi	34	45	7309	5698

```
SELstipendio > stipC (RENMatrC, NomeC, StipC, EtàC ← Matricola, Nome, Stipendio, Età (Impiegati) JOINMatrC=Capo  
(Supervisione JOINimpiegato=matricola Impiegati))
```

Esempi

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

PROJMatricola, Nome, Stipendio, MatrC, NomeC, StipC(**SEL**stipendio>stipC(**REN**MatrC, NomeC, StipC, EtàC
←Matricola, Nome, Stipendio, Età(**Impiegati**) **JOIN**MatrC=Capo
(**Supervisione JOIN**impiegato=matricola **Impiegati**)))

SELstipendio>stipC(**REN**MatrC, NomeC, StipC, EtàC ←Matricola, Nome, Stipendio, Età(**Impiegati**) **JOIN**MatrC=Capo
(**Supervisione JOIN**impiegato=matricola **Impiegati**)))

MatrC	NomeC	EtàC	StipC	Matricola	Nome	Età	Stipendio	Impiegato	Capo
5698	Bruni	43	42	7309	Rossi	34	45	7309	5698



Matricola	Nome	Stipendio	MatrC	NomeC	StipC
7309	Rossi	45	5698	Bruni	42

PROJMatricola, Nome, Stipendio, MatrC, NomeC, StipC(**SEL**stipendio>stipC(**REN**MatrC, NomeC, StipC, EtàC
←Matricola, Nome, Stipendio, Età(**Impiegati**) **JOIN**MatrC=Capo
(**Supervisione JOIN**impiegato=matricola **Impiegati**)))

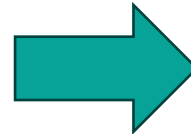
Esempi

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

$\text{PROJ}_{\text{Capo}}(\text{Supervisione}) - \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

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



Matricola	Nome	Età	Stipendio
5998	Bianchi	37	38
9553	Neri	42	35

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

Esempi

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

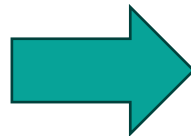
$\text{PROJ}_{\text{Capo}}(\text{Supervisione}) - \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{Impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

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

Matricola	Nome	Età	Stipendio
5998	Bianchi	37	38
9553	Neri	42	35

Supervisione

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



Matricola	Nome	Età	Stipendio	Impiegato	Capo
5998	Bianchi	37	38	5998	5698
9553	Neri	42	35	9553	4076

$\text{Supervisione JOIN}_{\text{Impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati}))$

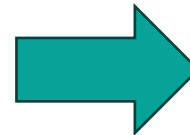
Esempi

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

$\text{PROJ}_{\text{Capo}}(\text{Supervisione}) - \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{Impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

Supervisione JOIN_{Impiegato=Matricola} (SEL_{stipendio≤40}(Impiegati))

Matricola	Nome	Età	Stipendio	Impiegato	Capo
5998	Bianchi	37	38	5998	5698
9553	Neri	42	35	9553	4076



Capo
5698
4076

$\text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{Impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

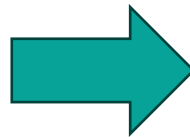
Esempi

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

$\text{PROJ}_{\text{Capo}}(\text{Supervisione}) - \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

Supervisione

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



Capo
5698
4076
8123

$\text{PROJ}_{\text{Capo}}(\text{Supervisione})$

Esempi

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

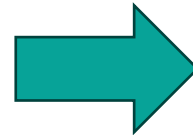
$\text{PROJ}_{\text{Capo}}(\text{Supervisione}) - \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

$\text{PROJ}_{\text{Capo}}(\text{Supervisione})$

Capo
5698
4076
8123

$\text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

Capo
5698
4076



Capo
8123

$\text{PROJ}_{\text{Capo}}(\text{Supervisione}) - \text{PROJ}_{\text{Capo}}(\text{Supervisione JOIN}_{\text{impiegato}=\text{Matricola}} (\text{SEL}_{\text{stipendio} \leq 40}(\text{Impiegati})))$

Equivalenza di espressioni

- Due espressioni sono **equivalenti** se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati
- L'equivalenza è importante in pratica perchè i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "costose"
- **Push selections** (se A è attributo di R2)
 - $SEL_{A=10} (R_1 JOIN R_2) = R_1 JOIN SEL_{A=10} (R_2)$
- Riduce in modo significativo la dimensione del risultato intermedio (e quindi, il costo dell'operazione)

Selezione con valori nulli

Impiegati

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

SEL $Età > 40$ (Impiegati)

- La condizione atomica è vera solo per valori non nulli

Un risultato non desiderabile

- $SEL_{Età > 30} (Persone) \cup SEL_{Età \leq 30} (Persone) \neq Persone$
- Perché? Perché le selezioni vengono valutate separatamente!
- Ma anche
 - $SEL_{Età > 30 \vee Età \leq 30} (Persone) \neq Persone$
- Perché? Perché anche le condizioni atomiche vengono valutate separatamente!

Selezione con valori nulli: soluzioni

- `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`
- Si potrebbe usare (ma non serve) una "logica a tre valori"
 - `(Vero, Falso, Sconosciuto)`

Selezione con valori nulli: soluzioni

- Quindi:

- $SEL_{Età > 30} (Persone) \cup SEL_{Età \leq 30} (Persone)$
 $\cup SEL_{Età \text{ IS NULL}} (Persone)$

=

$$SEL_{Età > 30 \vee Età \leq 30 \vee Età \text{ IS NULL}} (Persone)$$

=

Persone

Selezione con valori nulli: soluzioni

Impiegati

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

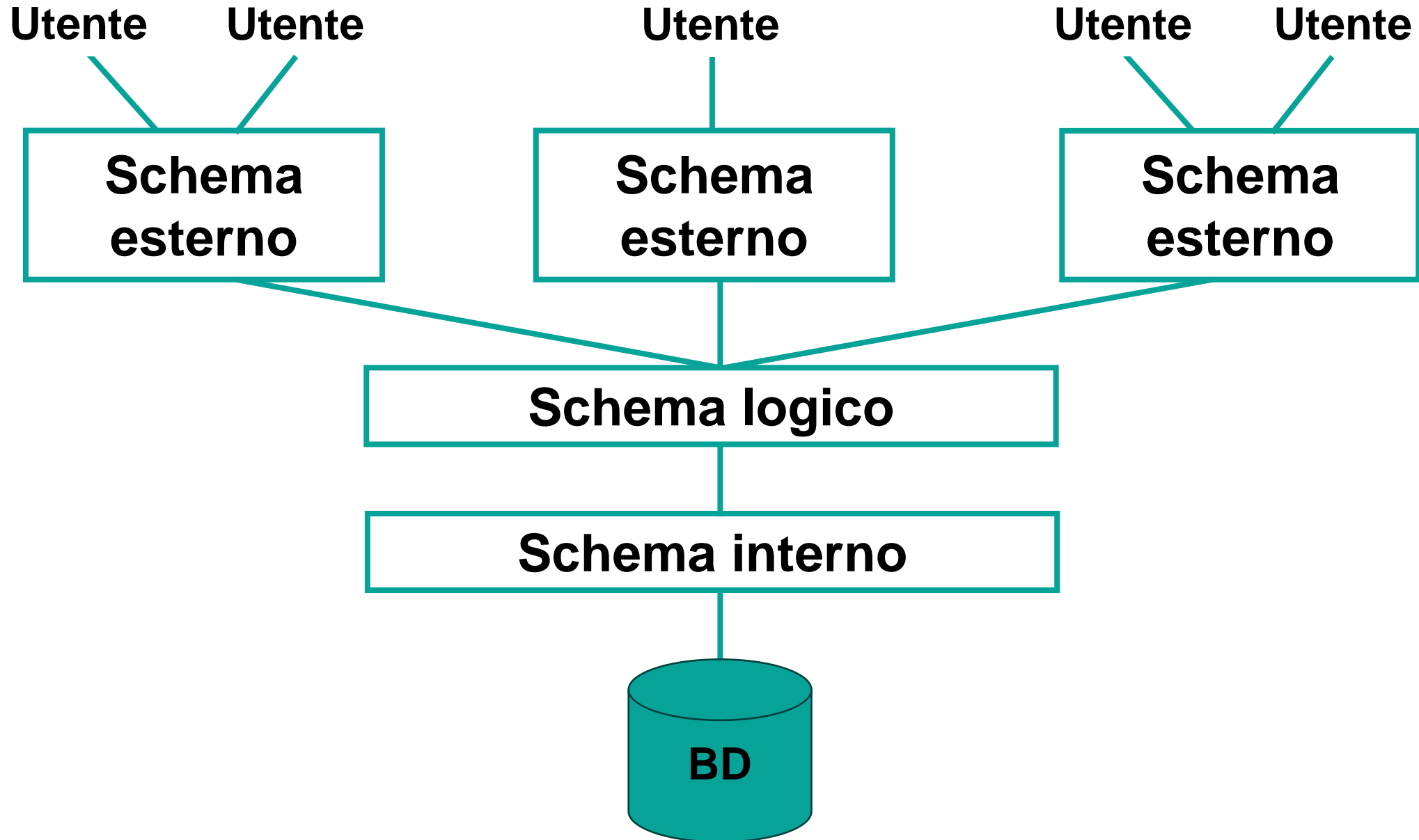
SEL (Età > 40) OR (Età IS NULL) (Impiegati)

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

Viste (relazioni derivate)

- Rappresentazioni diverse per gli stessi dati (schema esterno)
- Relazioni derivate:
 - Relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- Relazioni di base: contenuto autonomo
- Le relazioni derivate possono essere definite su altre derivate, ma...

Architettura standard (ANSI/SPARC) a tre livelli per DBMS



Viste virtuali e materializzate

- Due tipi di relazioni derivate:
- Viste materializzate
- Relazioni virtuali (o viste virtuali)

Viste materializzate

- Relazioni derivate memorizzate nella base di dati
- Vantaggi:
 - Immediatamente disponibili per le interrogazioni
- Svantaggi:
 - Ridondanti
 - Appesantiscono gli aggiornamenti
 - Sono raramente supportate dai DBMS

Viste, esempio

Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B
Bianchi	B

Direzione

Reparto	Capo
A	Mori
B	Bruni
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:
 - `SELCapo = 'Leoni' (Supervisione)`
- Viene eseguita come:
 - `SELCapo = 'Leoni'(
 PROJImpiegato, Capo (Afferenza JOIN Direzione))`

Viste, motivazioni

- Schema esterno: 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)
- Strumento di programmazione:
 - Si può semplificare la scrittura di interrogazioni: espressioni complesse e sotto-espressioni ripetute
- Utilizzo di programmi esistenti su schemi ristrutturati:
 - L'utilizzo di viste non influisce sull'efficienza delle interrogazioni

Viste come strumento di programmazione

- Trovare gli impiegati che hanno lo stesso capo di Rossi

- Senza vista:

- PROJ Impiegato ((Afferenza JOIN Direzione) JOIN

- REN impR,repR ← Impiegato, Reparto (

- SEL Impiegato ='Rossi' (Afferenza JOIN Direzione)))

- Con la vista:

- PROJ Impiegato ((Supervisione) JOIN

- REN impR ← Impiegato(

- SEL Impiegato ='Rossi' (Supervisione)))

Viste e aggiornamenti, attenzione

Afferenza

Impiegato	Reparto
Rossi	A
Neri	B
Verdi	A

Direzione

Reparto	Capo
A	Mori
B	Bruni
C	Bruni

Supervisione

Impiegato	Capo
Rossi	Mori
Neri	Bruni
Verdi	Mori

- Vogliamo inserire, nella vista, il fatto che Lupi ha come capo Bruni, oppure che Belli ha come capo Falchi. Come facciamo?

Viste e aggiornamenti

- "Aggiornare una vista":
 - Modificare le relazioni di base in modo tale che la vista, ricalcolata, rispecchi l'aggiornamento
- L'aggiornamento sulle relazioni di base corrispondente a quello specificato sulla vista deve essere univoco
- In generale però non è univoco!
- Ben pochi aggiornamenti sono ammissibili sulle viste

Operatori Minimali dell'algebra Relazionale

- 5 operatori con la composizione dei quali si ottengono tutti gli altri
- SEL, PROJ, X, U, --
- Ben pochi aggiornamenti sono ammissibili sulle viste
- $A \cap B = A \cup B - (A - B) - (B - A) = A - (A - B) = B - (B - A)$