

Esercizi

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 |

Esercizi

- 1) Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40
- 2) Trovare matricola, nome ed età degli impiegati che guadagnano più di 40
- 3) Trovare i capi degli impiegati che guadagnano più di 40
- 4) Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 40
- 5) Trovare gli impiegati che guadagnano più del proprio capo, mostrando matricola, nome e stipendio dell'impiegato e del capo
- 6) Trovare le matricole dei capi i cui impiegati guadagnano **tutti** più di 40

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

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

2) Trovare matricola, nome ed età degli impiegati che guadagnano più di 40

$\pi_{\text{Matricola, Nome, Età}} (\sigma_{\text{Stipendio} > 40}(\text{Impiegati}))$

3) Trovare i capi degli impiegati che guadagnano più di 40

$\pi_{\text{Capo}}(\text{Supervisione} \bowtie_{\text{Impiegato=Matricola}} (\sigma_{\text{Stipendio}>40}(\text{Impiegati})))$

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

$$\pi_{\text{Nome, Stipendio}} \left(\begin{array}{l} \text{Impiegati} \bowtie_{\text{Matricola}=\text{Capo}} \\ \pi_{\text{Capo}}(\text{Supervisione} \bowtie_{\text{Impiegato}=\text{Matricola}} \\ (\sigma_{\text{Stipendio} > 40}(\text{Impiegati}))) \end{array} \right)$$

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

$\pi_{\text{Matr, Nome, Stip, MatrC, NomeC, StipC}}$

$(\sigma_{\text{Stipendio} > \text{StipC}}($

$\rho_{\text{Matr, Nome, Stip, Età}} \leftarrow \text{MatrC, NomeC, StipC, EtàC}(\text{Impiegati})$

$\bowtie_{\text{MatrC} = \text{Capo}}$

$(\text{Supervisione} \bowtie_{\text{Impiegato} = \text{Matricola}} \text{Impiegati}))$

6) Trovare le matricole dei capi i cui impiegati guadagnano **tutti** più di 40

$$\pi_{\text{Capo}}(\text{Supervisione}) -$$
$$\pi_{\text{Capo}}(\text{Supervisione}$$
$$\bowtie_{\text{Impiegato}=\text{Matricola}}$$
$$(\sigma_{\text{Stipendio} \leq 40}(\text{Impiegati})))$$

Algebra relazionale:

Equivalenza di espressioni

- Due espressioni sono **equivalenti** se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati

$$\begin{aligned} \pi_{\text{Capo}}(\text{Supervisione} \bowtie_{\text{Impiegato=Matricola}} (\sigma_{\text{Stipendio}>40}(\text{Impiegati}))) \\ \equiv \\ \pi_{\text{Capo}}(\text{Supervisione} \bowtie_{\text{Impiegato=Matricola}} (\pi_{\text{Matricola}} (\sigma_{\text{Stipendio}>40}(\text{Impiegati})))) \end{aligned}$$

Esecuzione e ottimizzazione delle interrogazioni

```
SELECT COGNOME, NOME
FROM IMPIEGATO
WHERE STIPENDIO > ( SELECT MAX(STIPENDIO)
                     FROM IMPIEGATO
                     WHERE DIP=5);
```



BLOCCO ESTERNO

```
SELECT COGNOME, NOME
FROM IMPIEGATO
WHERE STIPENDIO > C
```

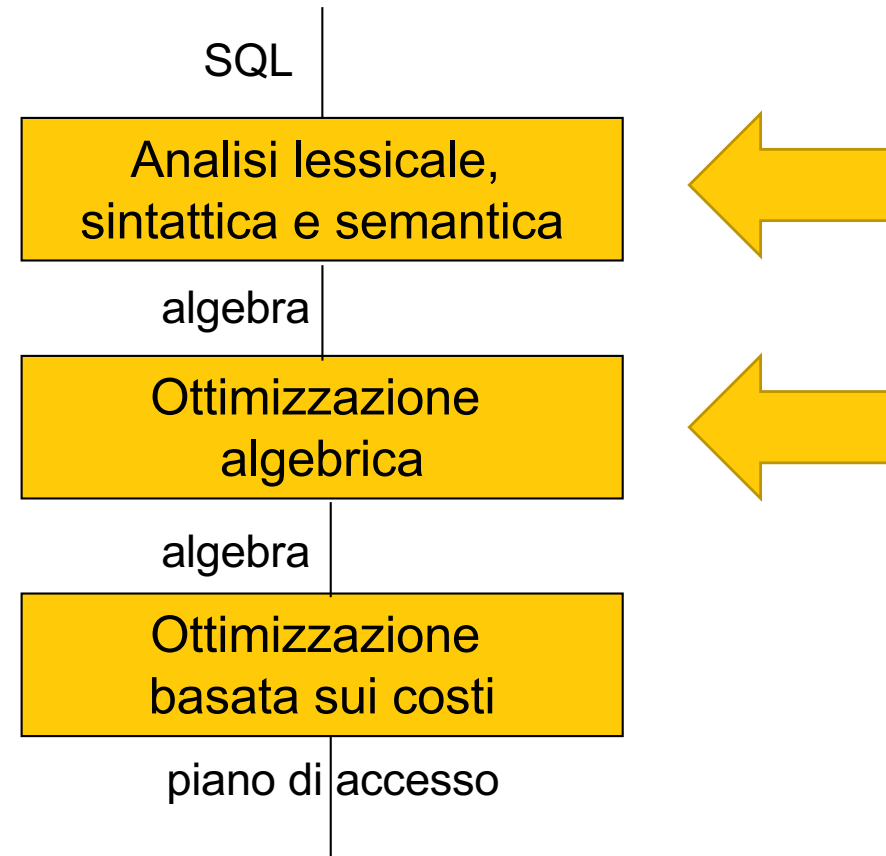
BLOCCO INTERNO

```
C= SELECT MAX(STIPENDIO)
    FROM IMPIEGATO
    WHERE DIP=5
```

BLOCCO ESTERNO

$$\pi_{\text{COGNOME, NOME}}(\sigma_{\text{STIPENDIO} > C}(\text{IMPIEGATO}))$$

BLOCCO INTERNO

$$C = F_{\text{MAX STIPENDIO}}(\sigma_{\text{DIP}=5}(\text{IMPIEGATO}))$$


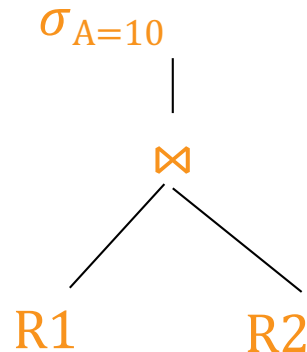
Ottimizzazione algebrica

- L'ottimizzazione algebrica si basa sulla nozione di equivalenza:
 - il DBMS cerca di eseguire espressioni equivalenti a quelle date, ma **meno «costose»!**
- Il termine **ottimizzazione** è improprio perché il processo utilizza euristiche
 - Vedremo alcune euristiche comunemente adottate per ottimizzare l'espressione algebrica
- Prima introduciamo una modellazione ad albero delle espressioni algebriche

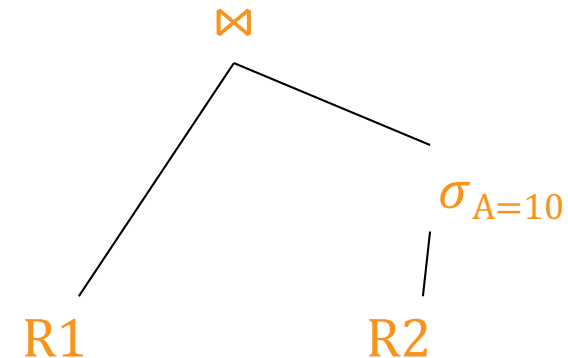
Albero di interrogazioni (query tree)

- Struttura ad albero per rappresentare un'espressione algebrica:
 - Nodi foglia sono le relazioni di input
 - Nodi interni rappresentano le operazioni dell'algebra relazionale
- L'albero rappresenta la sequenza di operazioni relazionali da eseguire partendo dai nodi foglia

$$\sigma_{A=10}(R_1 \bowtie R_2)$$



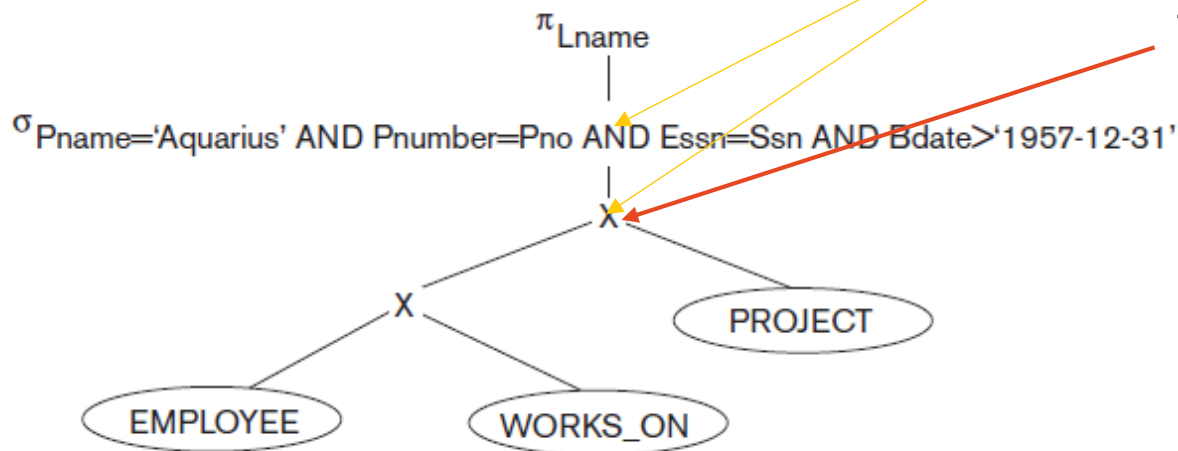
$$R_1 \bowtie (\sigma_{A=10}(R_2))$$



Albero di interrogazioni (query tree)

- Struttura ad albero per rappresentare un'espressione algebrica:
 - Nodi foglia sono le relazioni di input
 - Nodi interni rappresentano le operazioni dell'algebra relazionale
- L'albero rappresenta la sequenza di operazioni relazionali da eseguire partendo dai nodi foglia

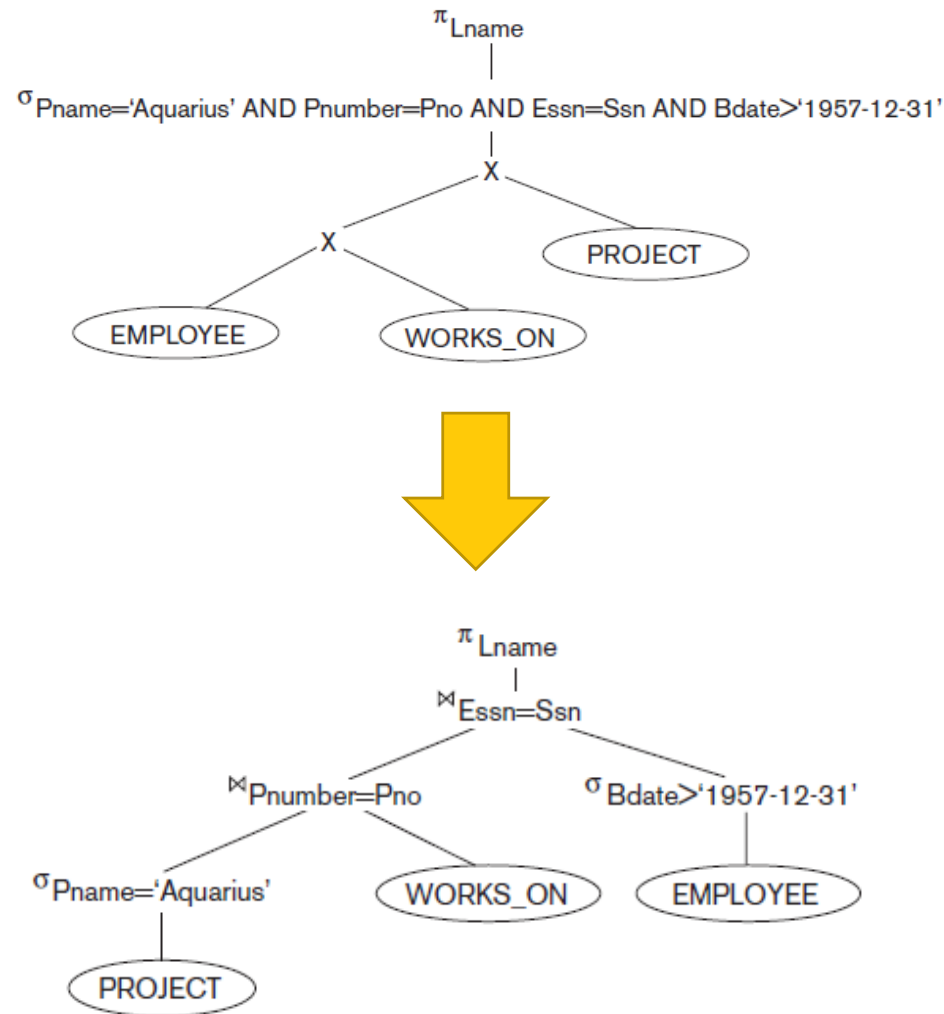
```
SELECT E.Lname
FROM EMPLOYEE E, WORKS_ON W, PROJECT P
WHERE P.Pname='Aquarius' AND P.Pnumber=W.Pno AND
      E.Essn=W.ssn AND E.Bdate>'1957-12-31'
```



- Per ogni operazione, il DBMS offre più algoritmi che la implementano (si vedranno successivamente!)
- Ogni operazione produce file temporanei che possono essere anche molto grandi!

Ottimizzazione algebrica

- Dato un albero di interrogazione l'ottimizzatore ha il compito di trasformarlo in un albero finale che sia più efficiente da eseguire



Ottimizzazione algebrica

- Vedremo alcune euristiche comunemente adottate per ottimizzare l'espressione algebrica

Regole di trasformazione per espressioni algebriche

Regole di trasformazione espressioni algebriche

1) Un operatore di selezione con condizioni in congiunzione può essere spezzato in una sequenza di singole operazioni di σ

$$\sigma_{\theta_1 \wedge \theta_2}(E) \equiv \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

2) L'operatore di selezione è commutativo

$$\sigma_{\theta_1}(\sigma_{\theta_2}(E)) \equiv \sigma_{\theta_2}(\sigma_{\theta_1}(E))$$

3) In una sequenza di proiezioni, solo l'ultimo operatore è significativo

$$\Pi_{L_1}(\Pi_{L_2}(\dots(\Pi_{L_n}(E))\dots)) \equiv \Pi_{L_1}(E)$$

4) $\Pi_L(\sigma_\theta(E))$, $L \subseteq U_E$

se la condizione di θ è applicata solo sugli attributi L , è possibile commutare σ_θ con Π

$$\Pi_L(\sigma_\theta(E)) \equiv \sigma_\theta(\Pi_L(E))$$

Regole di trasformazione espressioni algebriche

5) Theta (natural) join è commutativo:

$$E1 \bowtie_{\theta} E2 \equiv E2 \bowtie_{\theta} E1$$

6) Natural-join è associativo:

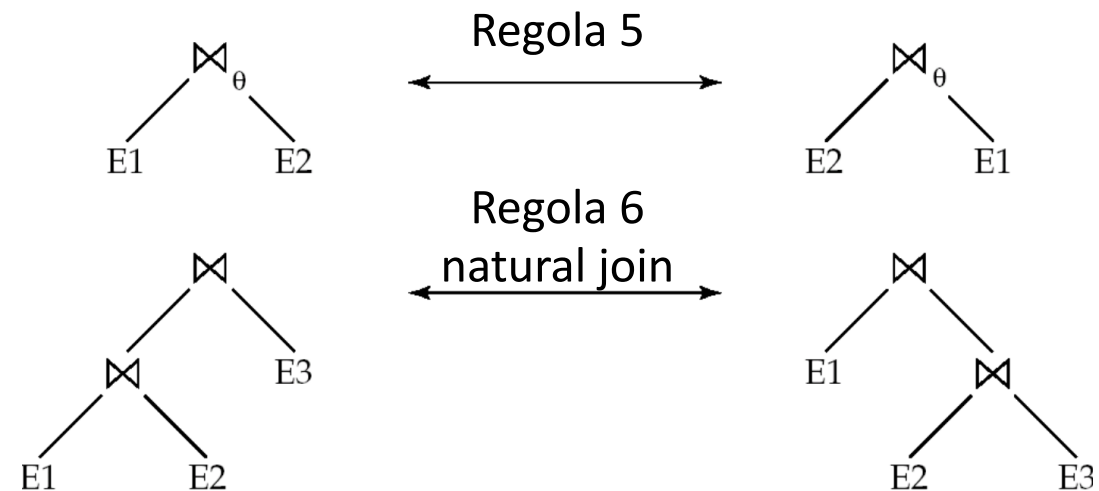
$$(E1 \bowtie E2) \bowtie E3 \equiv E1 \bowtie (E2 \bowtie E3)$$

6.a Anche il Theta joins può essere associativo nel seguente modo

$$(E1 \bowtie_{\theta_1} E2) \bowtie_{\theta_2 \wedge \theta_3} E3 \equiv$$

$$E1 \bowtie_{\theta_1 \wedge \theta_3} (E2 \bowtie_{\theta_2} E3)$$

Se θ_2 è solo su attributi di $E2$ e $E3$ (e non $E1$)



Regole di trasformazione espressioni algebriche

7) $\sigma_{\theta_0}(E_1 \bowtie_{\theta} E_2)$

- L'operatore di selezione può precedere/commutare un theta join/natural join nei seguenti casi

(a) quando θ_0 è definito solo sugli attributi di una delle due relazioni del theta/natural join

$$\sigma_{\theta_0}(E_1 \bowtie_{\theta} E_2) \equiv (\sigma_{\theta_0}(E_1)) \bowtie_{\theta} E_2$$

(b) quando la condizione è in congiunzione $\theta_0 = \theta_1 \wedge \theta_2$, e θ_1 è definito sugli attributi di E_1 e θ_2 su attributi di E_2

$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_{\theta} E_2) \equiv (\sigma_{\theta_1}(E_1)) \bowtie_{\theta} (\sigma_{\theta_2}(E_2))$$

-L'operatore di selezione può essere combinato con il prodotto cartesiano e theta join

- $\sigma_{\theta}(E_1 \times E_2) \equiv E_1 \bowtie_{\theta} E_2$

- $\sigma_{\theta_0}(E_1 \bowtie_{\theta_1} E_2) \equiv E_1 \bowtie_{\theta_0 \wedge \theta_1} E_2$

Regole di trasformazione espressioni algebriche

8) $\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2)$ con $L_1 \subseteq U_{E_1}$ e $L_2 \subseteq U_{E_2}$

l'operatore di proiezione può precedere un theta join come segue:

(a) se θ si applica solo ad attributi in $L_1 \cup L_2$:

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) \equiv \Pi_{L_1}(E_1) \bowtie_{\theta} \Pi_{L_2}(E_2)$$

(b) Più in generale,

- sia L_3 l'insieme di attributi E_1 coinvolti nella condizione di join θ , *ma non* in $L_1 \cup L_2$,
- sia L_4 l'insieme di attributi E_2 coinvolti nella condizione di join θ , *ma non* in $L_1 \cup L_2$,

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) \equiv \Pi_{L_1 \cup L_2}(\Pi_{L_1 \cup L_3}(E_1) \bowtie_{\theta} \Pi_{L_2 \cup L_4}(E_2))$$

Regole di trasformazione espressioni algebriche

9. L'operatore unione e intersezione sono commutativi

$$E_1 \cup E_2 \equiv E_2 \cup E_1$$

$$E_1 \cap E_2 \equiv E_2 \cap E_1$$

10. L'operatore unione e intersezione sono associativi

$$(E_1 \cup E_2) \cup E_3 \equiv E_1 \cup (E_2 \cup E_3)$$

$$(E_1 \cap E_2) \cap E_3 \equiv E_1 \cap (E_2 \cap E_3)$$

11. La selezione può precedere \cup , \cap and $-$

$$\text{a. } \sigma_{\theta}(E_1 \cup E_2) \equiv \sigma_{\theta}(E_1) \cup \sigma_{\theta}(E_2)$$

$$\text{b. } \sigma_{\theta}(E_1 \cap E_2) \equiv \sigma_{\theta}(E_1) \cap \sigma_{\theta}(E_2)$$

$$\text{c. } \sigma_{\theta}(E_1 - E_2) \equiv \sigma_{\theta}(E_1) - \sigma_{\theta}(E_2)$$

12. L'operatore di proiezione può precedere l'unione

$$\Pi_L(E_1 \cup E_2) \equiv (\Pi_L(E_1)) \cup (\Pi_L(E_2))$$

Ottimizzazione algebrica

- Dato un albero di interrogazione l'ottimizzatore ha il compito di trasformarlo in un albero finale che sia più efficiente da eseguire
- Utilizzando le trasformazioni appena introdotte, l'algoritmo di ottimizzazione algebrica esegue, a grandi linee, i seguenti passi:
 1. Scompone ogni operatore σ con condizione in congiunzione in una sequenza $\sigma_1 \dots \sigma_n$
 2. Sfruttando la proprietà commutativa del σ , sposta ogni operatore σ_j quanto più possibile verso il basso dell'albero di interrogazione
 3. Sfruttando la proprietà commutativa del Π , sposta ogni operatore Π_j quanto più possibile verso il basso dell'albero di interrogazione, mantenendo solo gli attributi necessari per eseguire le operazioni successivi (eventualmente anche scomponendo un operatore Π in più operatori Π)
 4. Sfruttando la proprietà commutative e associative riposiziona i nodi in modo che le operazioni σ_θ più selettive siano posizionate per essere eseguite per prima di altri operatori $(\sigma, \Pi, X, \bowtie_\theta)$
 5. Combina i prodotti cartesiani e le selezioni per formare join

Ottimizzazione algebrica: esercizio

- Quali regole posso applicare?

