

Laboratorio 9

Ottimizzazione Algebrica

rif. paragrafi 5.1 – 5.2, 16.1 – 16.3 libro

Corso di Basi di Dati

A.A. 2012/2013

Presented by
Matteo Lissandrini
matteo.lissandrini@gmail.com

Courtesy of:
Francesco Corcoglioniti
corcoglioniti@disi.unitn.it

Algebra Relazionale Estesa

- Operatori unari

- Proiezione $\pi_{\text{attributo}, \dots, \text{exp} \rightarrow \text{attributo}} (R)$
- Selezione $\sigma_{\text{condizione}} (R)$
- Ridenominazione $\rho_{S(\text{attributo}_1, \dots, \text{attributo}_N)} (R)$
- Rimozione duplicati $\delta (R)$
- Grouping $\gamma_{\text{attributo}, \dots, \text{operatore}(\text{exp}) \rightarrow \text{attributo}, \dots} (R)$
- Sorting $\tau_{\text{attributo}, \dots} (R)$

- Operatori binari insiemistici (varianti bag / set)

- Unione $R \cup_B S, \quad R \cup_S S$
- Intersezione $R \cap_B S, \quad R \cap_S S$
- Differenza $R -_B S, \quad R -_S S$

- Operatori binari di combinazione tuple

- Prodotto cartesiano $R \times S$
- Natural join $R \bowtie S$
- Theta join $R \bowtie_{\theta} S$
- Outer join $R \bowtie_{\text{condizione}}^O S, \quad R \bowtie_{L, \text{condizione}}^O S, \quad R \bowtie_{R, \text{condizione}}^O S$

Proprietà e Leggi Algebriche (1)

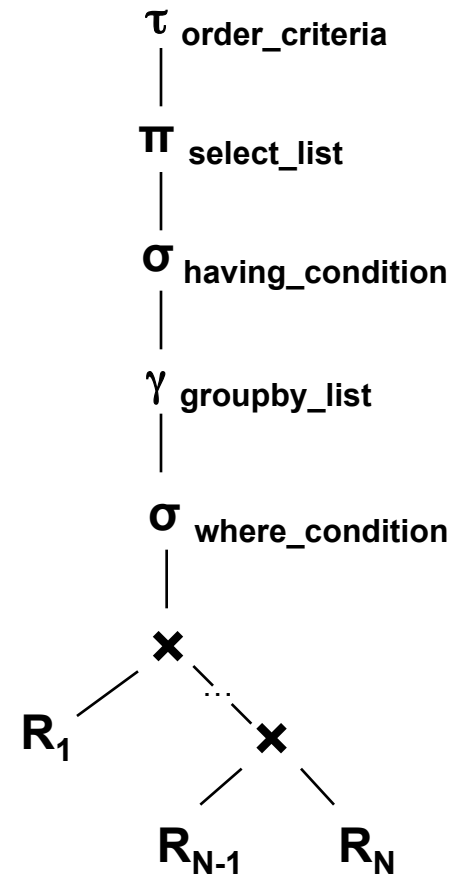
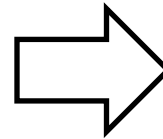
- commutatività e associatività
 - $\cup_B, \cup_S, \cap_B, \cap_S, \bowtie, \times$ sono commutativi e associativi
 - \bowtie_C, \bowtie^0 sono commutativi sempre, associativi in certi casi
- decomposizione σ
 - $\sigma_{c1 \text{ AND } c2}(R) = \sigma_{c1}(\sigma_{c2}(R))$
 - $\sigma_{c1 \text{ OR } c2}(R) = \sigma_{c1}(R) \cup_S \sigma_{c2}(R)$ per R senza duplicati
- pushing σ
 - $\sigma_c(\text{op}(R)) = \text{op}(\sigma_c(R))$ per operatori π, δ, τ ; per γ se la condizione non coinvolge aggregati; per δ a patto di riscrivere la condizione
 - $\sigma_c(R \text{ op } S) = \sigma_c(R) \text{ op } \sigma_c(S)$ per tutti operatori binari tranne outer join
 - $\sigma_c(R \text{ op } S) = \sigma_c(R) \text{ op } S$ per tutti operatori binari tranne outer join e unione

Proprietà e Leggi Algebriche (2)

- pushing π
 - sempre possibile
 - occorre aggiungere attributi usati da operatori superiori nell'albero
- pushing δ
 - possibile sempre tranne che per U_B, \neg_B, π
- pushing γ
 - dipende dagli operatori aggregati utilizzati
- trasformazione prodotto cartesiano
 - $\sigma_c(R \times S) = R \bowtie_c S$
 - $\pi_L(\sigma_c(R \times S)) = R \bowtie S$ per C condizione di uguaglianza tra attributi con stesso nome e L lista di attributi che rimuove con stesso nome

Conversione Query SQL

SELECT	select_list	⑤
FROM	R_1, \dots, R_N	①
WHERE	where_condition	②
GROUP BY	groupby_list	③
HAVING	having_condition	④
ORDER BY	order_criteria	⑥



Ottimizzazione Algebrica

- decomposizione selezioni con AND condizioni
- push selezioni
- push proiezioni
- introduzione proiezioni
- push rimozione duplicati
- sostituzione cartesiano + selezione in join

Esercizio 1 – Prodotti

Sia dato lo schema:

- product (model, maker, type)
- pc (model, speed, ram, hd, price)
- laptop (model, speed, ram, hd, screen, price)
- printer (model, color, type, price)

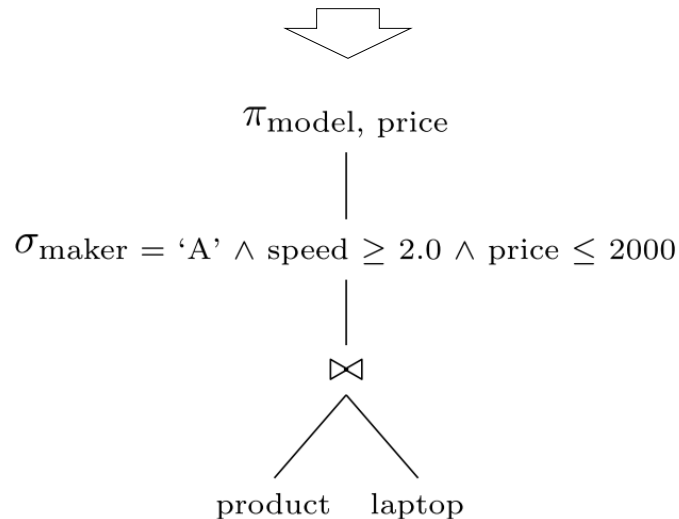
Esprimere in SQL le seguenti query, convertire in algebra relazionale e effettuare l'ottimizzazione algebrica:

1. Elencare modello e prezzo dei laptop prodotti da 'A' di almeno 2.0 GHz e con prezzo inferiore a 2000\$
2. Trovare le dimensioni di HD in comune a due o più laptop
3. Trovare, per ogni produttore, il prezzo medio dei laptop venduti di almeno 15"
4. Elencare, in ordine alfabetico, i produttori che vendono almeno tre diversi modelli di PC veloci almeno 2 GHz
5. Trovare i produttori di laptop con schermo inf. a 14" (**non** usare sotto-query)
6. Trovare il laptop con il prezzo più alto
7. Trovare il prezzo medio di PC e laptop venduti dal produttore 'D'

Esercizio 1 – Soluzioni (1/8)

1. Elencare modello e prezzo dei laptop prodotti da 'A' di almeno 2.0 GHz e con prezzo inferiore a 2000\$

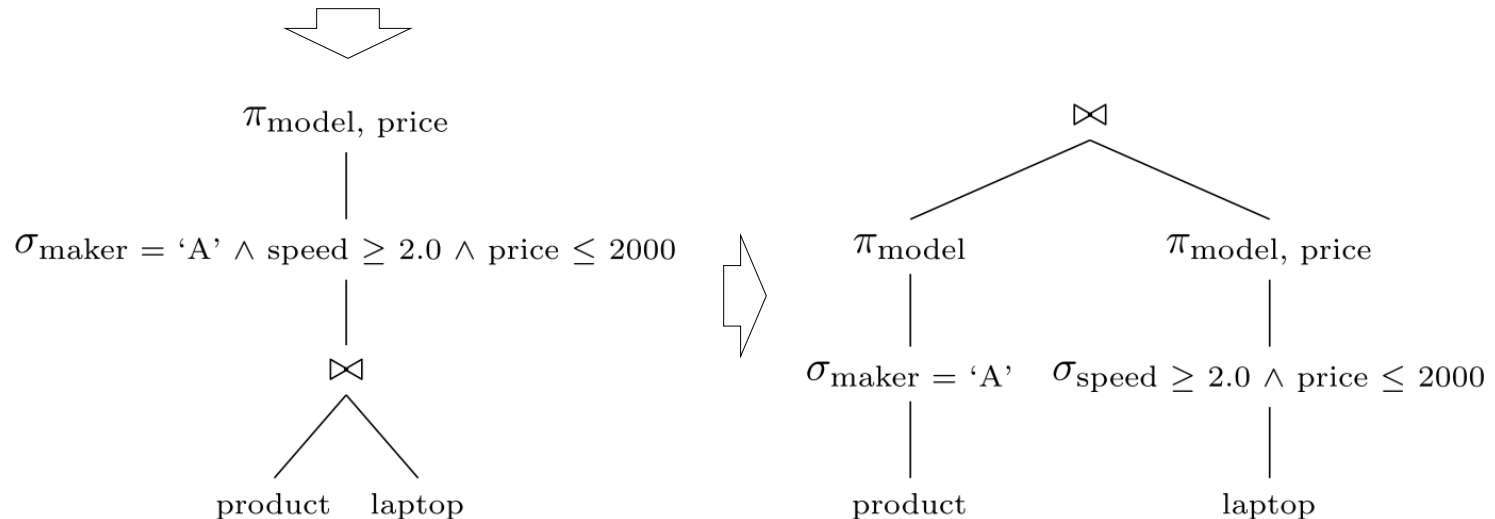
```
SELECT model, price  
FROM product NATURAL JOIN laptop  
WHERE maker = 'A' AND speed  $\geq$  2.0 AND price  $<$  2000
```



Esercizio 1 – Soluzioni (1/8)

1. Elencare modello e prezzo dei laptop prodotti da 'A' di almeno 2.0 GHz e con prezzo inferiore a 2000\$

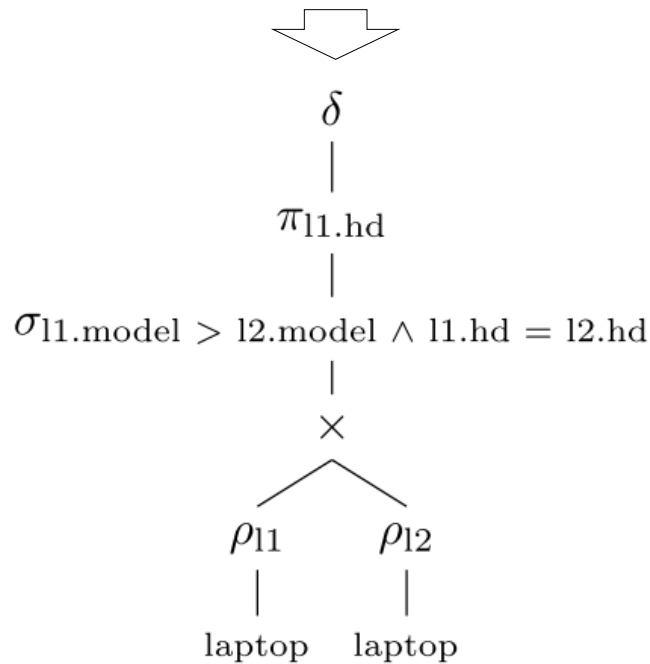
```
SELECT model, price
FROM   product NATURAL JOIN laptop
WHERE  maker = 'A' AND speed >= 2.0 AND price < 2000
```



Esercizio 1 – Soluzioni (2/8)

2. Trovare le dimensioni di HD in comune a due o più laptop

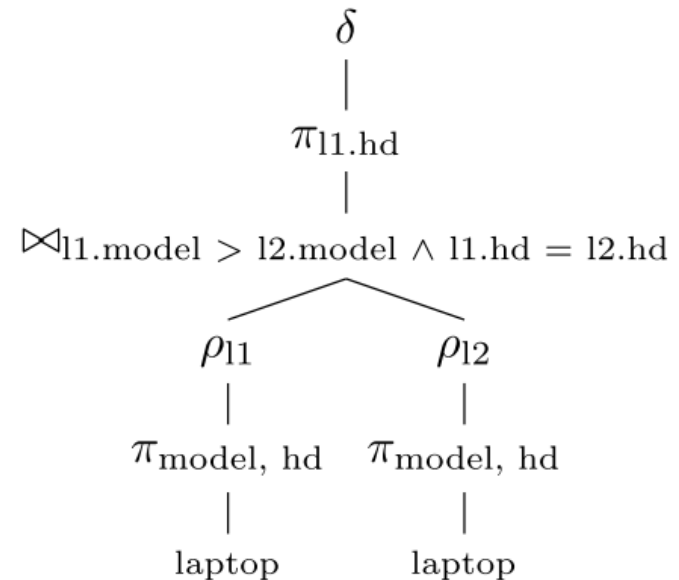
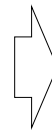
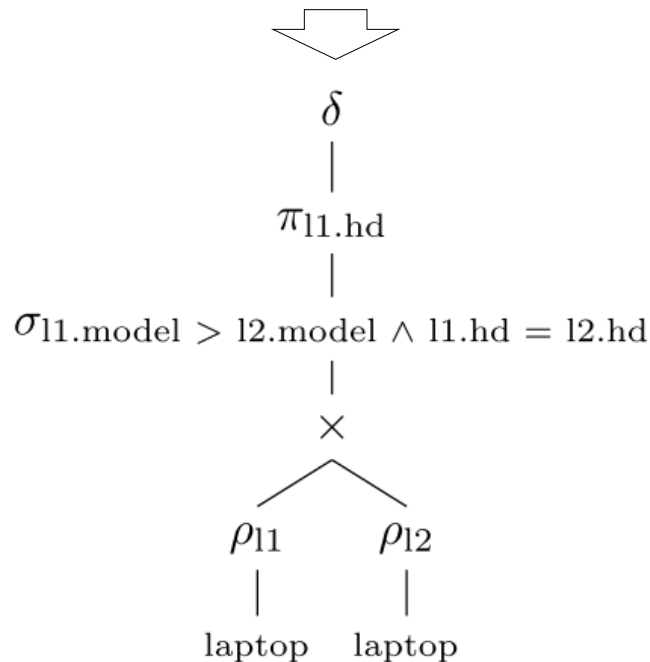
```
SELECT DISTINCT l1.hd  
FROM laptop l1, laptop l2  
WHERE l1.model > l2.model AND l1.hd = l2.hd
```



Esercizio 1 – Soluzioni (2/8)

2. Trovare le dimensioni di HD in comune a due o più laptop

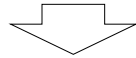
```
SELECT DISTINCT l1.hd  
FROM laptop l1, laptop l2  
WHERE l1.model > l2.model AND l1.hd = l2.hd
```



Esercizio 1 – Soluzioni (3/8)

3. Trovare, per ogni produttore, il prezzo medio dei laptop venduti di almeno 15”

```
SELECT  maker, AVG(price) AS avg_price
FROM    product NATURAL JOIN laptop
WHERE   screen >= 15
GROUP BY maker
```

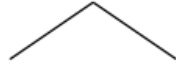


$\gamma_{\text{maker}, \text{AVG}(\text{price}) \rightarrow \text{avg-price}}$

|

$\sigma_{\text{screen} \geq 15}$

|

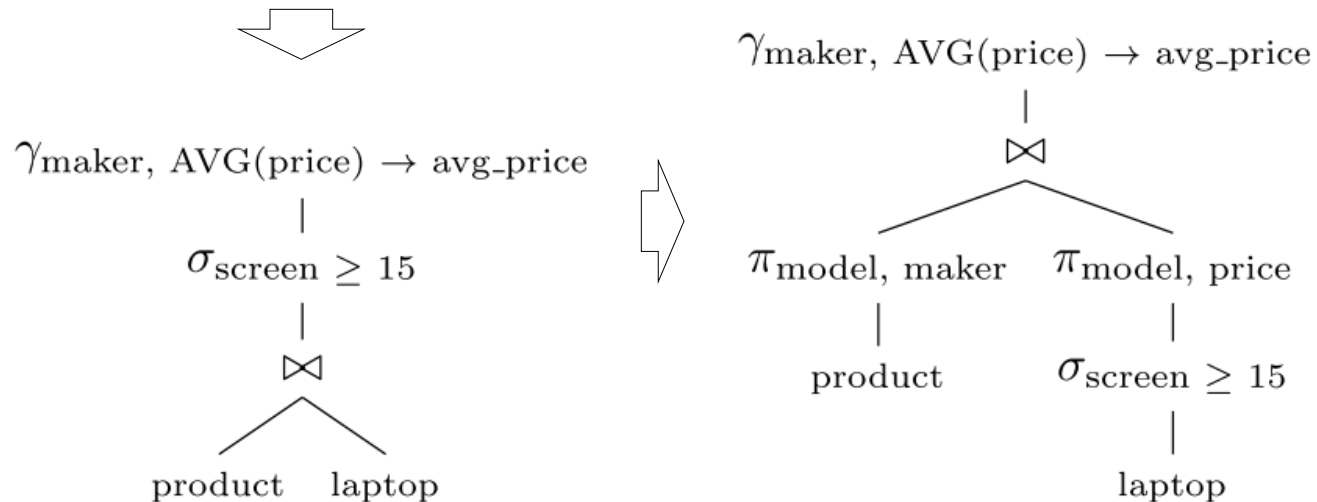


product laptop

Esercizio 1 – Soluzioni (3/8)

3. Trovare, per ogni produttore, il prezzo medio dei laptop venduti di almeno 15”

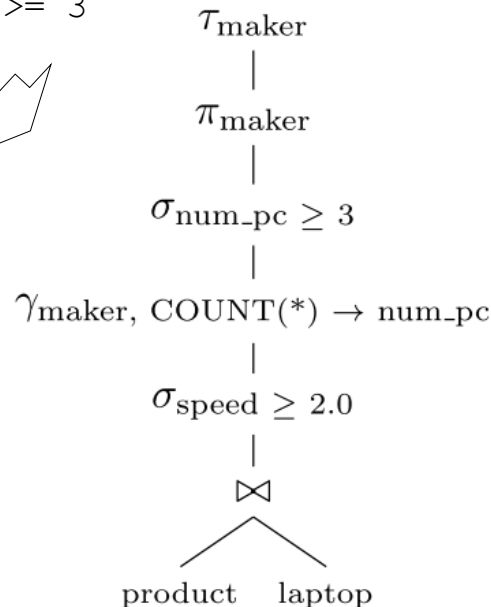
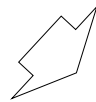
```
SELECT  maker, AVG(price) AS avg_price
FROM    product NATURAL JOIN laptop
WHERE   screen >= 15
GROUP BY maker
```



Esercizio 1 – Soluzioni (4/8)

4. Elencare, in ordine alfabetico, i produttori che vendono almeno tre diversi modelli di PC veloci almeno 2 GHz

```
SELECT  maker
FROM    product NATURAL JOIN pc
WHERE   speed >= 2.0
GROUP BY maker
HAVING  COUNT(*) >= 3
ORDER BY maker
```

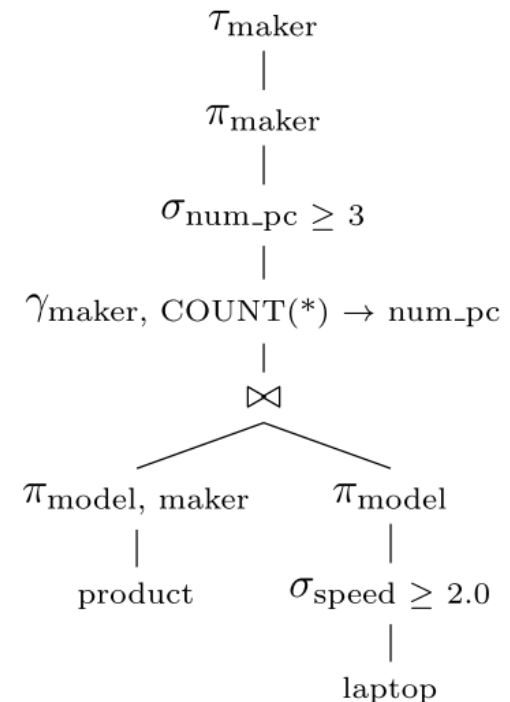
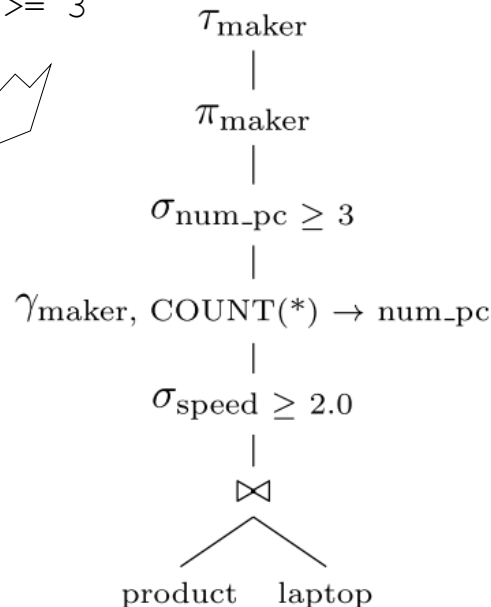


Esercizio 1 – Soluzioni (4/8)

4. Elencare, in ordine alfabetico, i produttori che vendono almeno tre diversi modelli di PC veloci almeno 2 GHz

```

SELECT  maker
FROM    product NATURAL JOIN pc
WHERE    speed >= 2.0
GROUP BY maker
HAVING   COUNT(*) >= 3
ORDER BY maker
    
```



Esercizio 1 – Soluzioni (5/8)

5. Trovare i produttori di laptop con schermo inf. a 14" (**non** usare sotto-query)

```
SELECT DISTINCT maker  
FROM    product NATURAL JOIN laptop  
WHERE screen < 14
```


Esercizio 1 – Soluzioni (6/8)

6. Trovare il laptop con il prezzo più alto

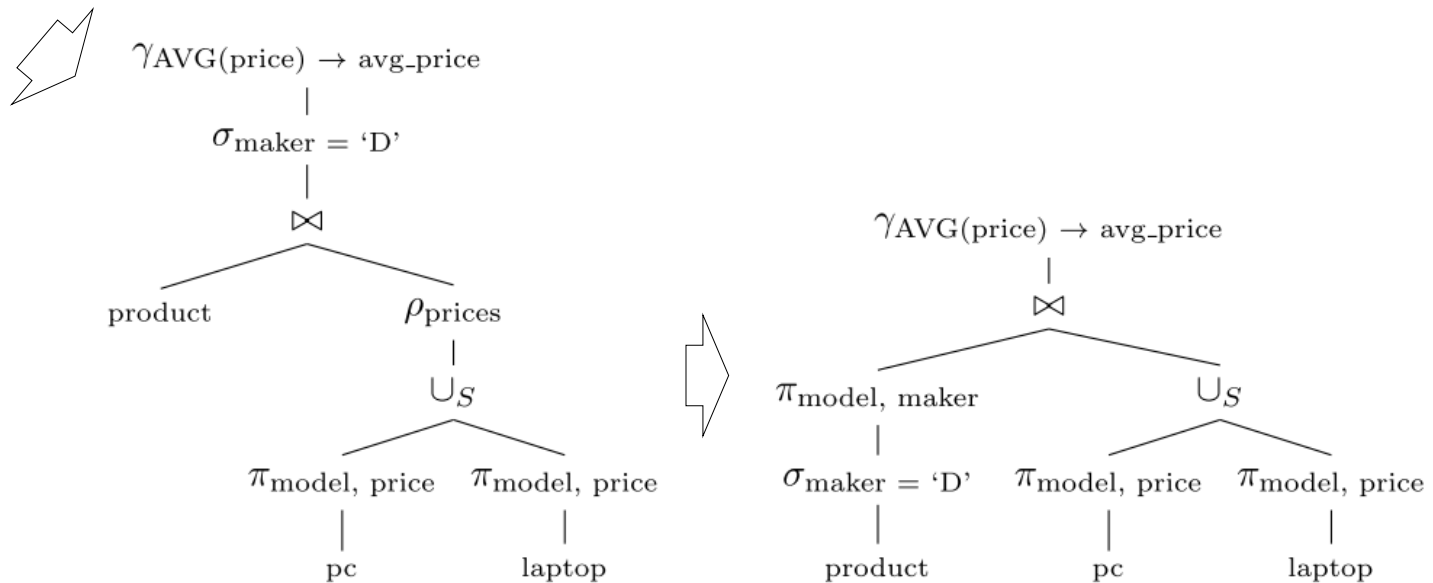
```
SELECT *  
FROM   laptop NATURAL JOIN  
( SELECT MAX(price) AS price FROM   laptop )
```

Esercizio 1 – Soluzioni (7/8)

7. Trovare il prezzo medio di PC e laptop venduti dal produttore 'D'

```

SELECT AVG(price) AS avg_price
FROM   product NATURAL JOIN ( ( SELECT model, price
                                FROM   pc )
                                UNION
                                ( SELECT model, price
                                FROM   laptop ) ) AS prices
WHERE  maker = 'D';
    
```



Esercizio 2 – Navi da battaglia

Sia dato lo schema: `classes` (class, type, country, num_guns, bore, displacement)
`ships` (name, class, launched)
`outcomes` (ship, battle, result)
`battles` (name, date)

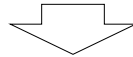
Esprimere in SQL le seguenti query, convertire in algebra relazionale e effettuare l'ottimizzazione algebrica:

1. Elencare le navi varate prima del 1942 e con stazza maggiore di 40000 t
2. Elencare nome e stazza delle corazzate (bb) varate a partire dal 1940 che parteciparono alla battaglia di Guadalcanal
3. Trovare i paesi che, prima del 1942, avevano varato sia corazzate (bb) che incrociatori (bc)
4. Ritornare, anno per anno, il numero di navi varate e la stazza totale (somma delle staze delle singole navi)
5. Trovare, per ogni paese che ha costruito almeno 3 navi, l' anno dell' ultimo varo
6. Trovare nome e paese della classe di corazzata (bb) avente stazza massima

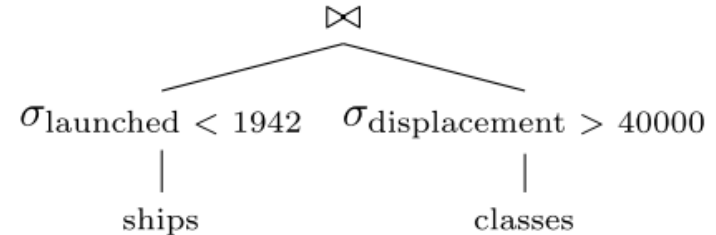
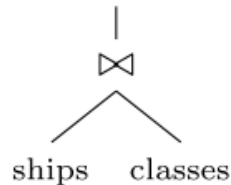
Esercizio 2 – Soluzioni (1/7)

1. Elencare le navi varate prima del 1942 e con stazza maggiore di 40000 t

```
SELECT *  
FROM   ships NATURAL JOIN classes  
WHERE  launched < 1942 AND displacement > 40000
```



$\sigma_{\text{launched} < 1942 \wedge \text{displacement} > 40000}$

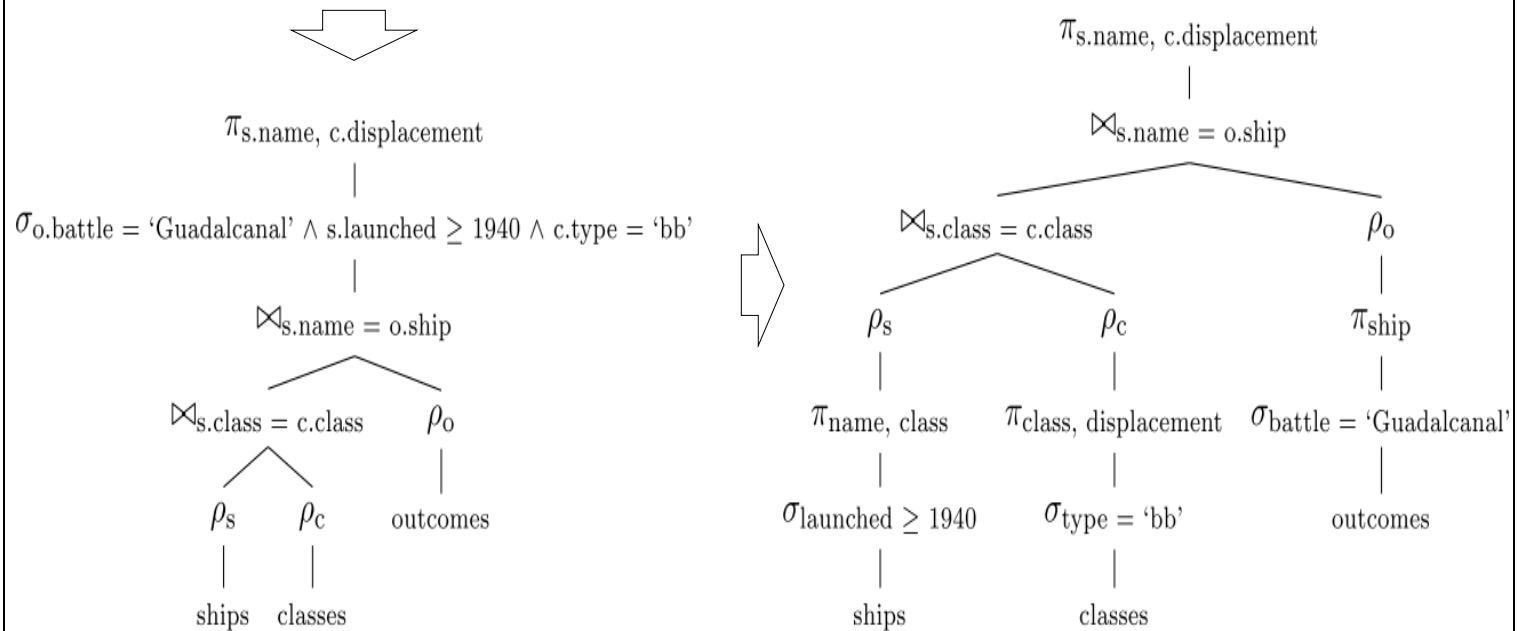


Esercizio 2 – Soluzioni (2/7)

2. Elencare nome e stazza delle corazzate (bb) varate a partire dal 1940 che parteciparono alla battaglia di Guadalcanal

```

SELECT s.name, c.displacement
FROM   ships s JOIN classes c ON s.class = c.class
        JOIN outcomes o ON s.name = o.ship
WHERE  o.battle = 'Guadalcanal' AND s.launched >= 1940 AND type =
        'bb'
    
```



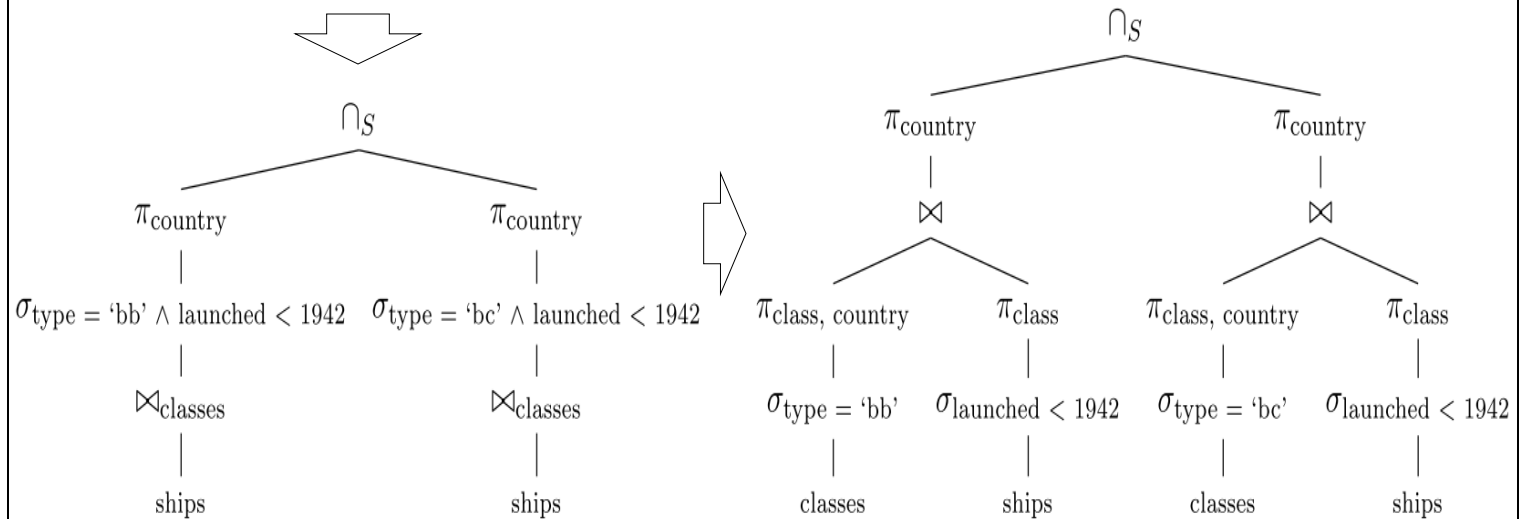
Esercizio 2 – Soluzioni (3/7)

3. Trovare i paesi che, prima del 1942, avevano varato sia corazzate (bb) che incrociatori (bc)

```
( SELECT country
  FROM classes NATURAL JOIN ships
  WHERE type = 'bb' AND launched < 1942 )
```

INTERSECT

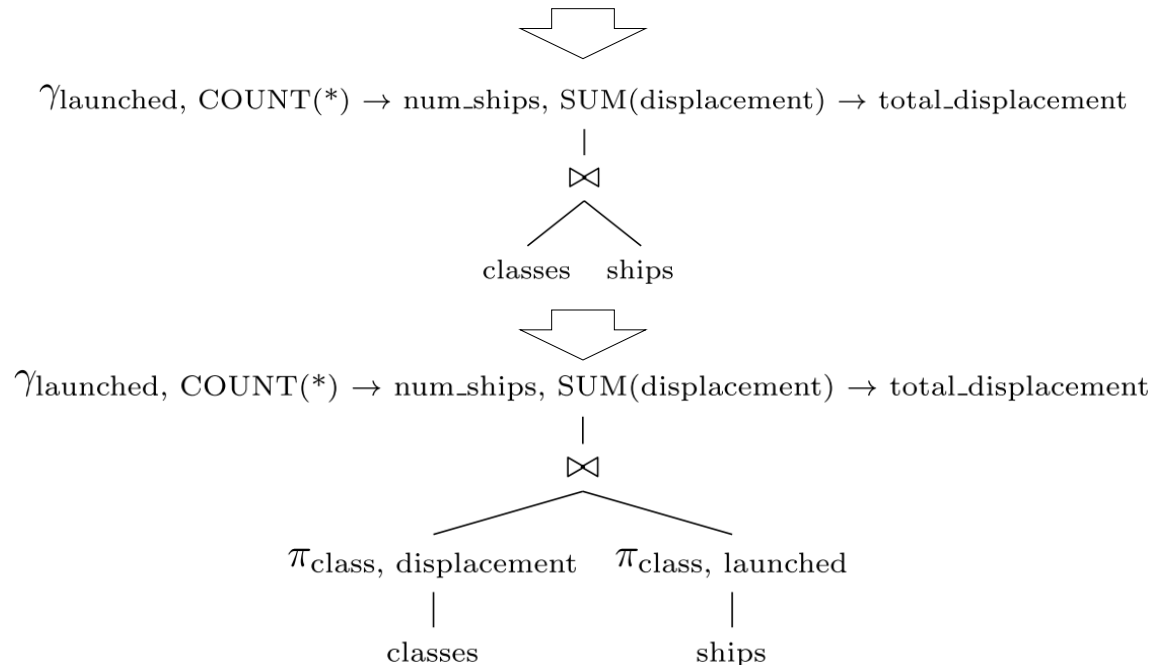
```
( SELECT country
  FROM classes NATURAL JOIN ships
  WHERE type = 'bc' AND launched < 1942 )
```



Esercizio 2 – Soluzioni (4/7)

4. Ritornare, anno per anno, il numero di navi varate e la stazza totale (somma delle staze delle singole navi)

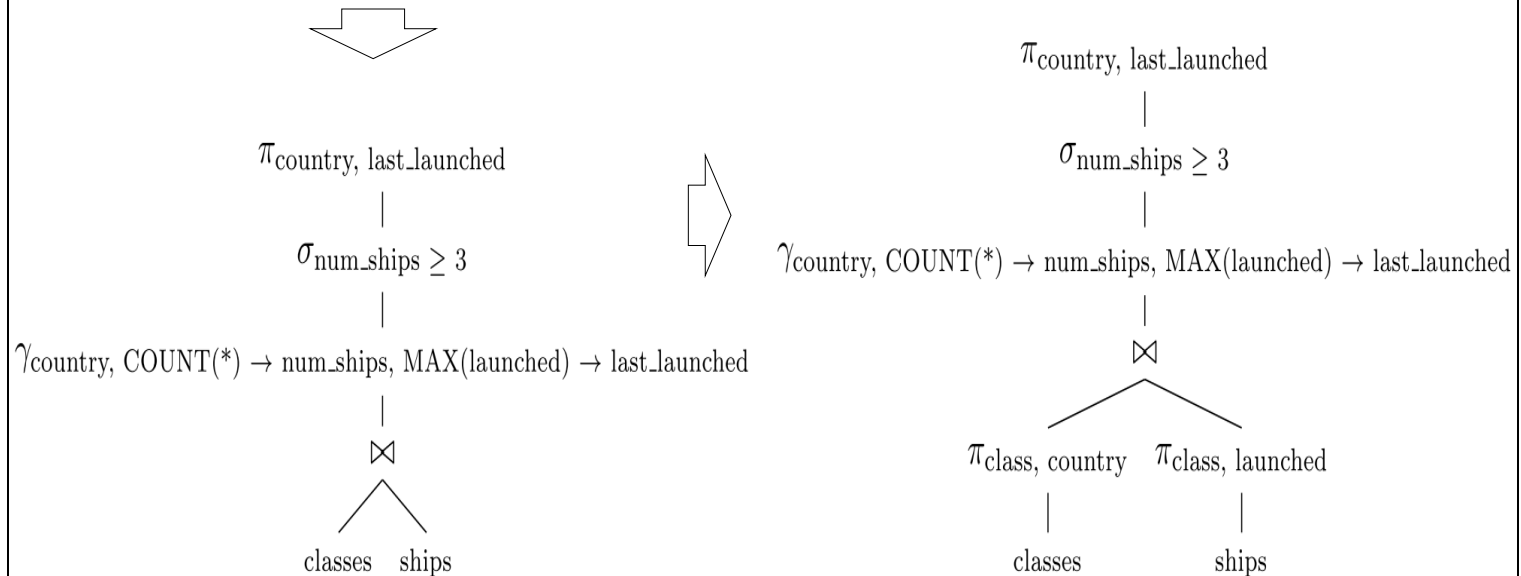
```
SELECT   launched, COUNT(*) AS num_ships,  
          SUM(displacement) AS total_displacement  
FROM     classes NATURAL JOIN ships  
GROUP BY launched
```



Esercizio 2 – Soluzioni (5/7)

5. Trovare, per ogni paese che ha costruito almeno 3 navi, l'anno dell'ultimo varo

```
SELECT   country, MAX(launched) AS last_launched
FROM     classes NATURAL JOIN ships
GROUP BY country
HAVING    COUNT(*) >= 3
```



Esercizio 2 – Soluzioni (6/7)

6. Trovare nome e paese della classe di corazzata (bb) avente stazza massima (**convertire la seguente in teta join**)

```
SELECT class, country
FROM   classes
WHERE  type = 'bb' AND
       displacement = ( SELECT MAX(displacement)
                        FROM   classes
                        WHERE  type = 'bb' )
```

Esercizi dal libro

Exercise 16.2.1: Give examples to show that:

- a) Duplicate elimination (δ) cannot be pushed below projection.
- b) Duplicate elimination cannot be pushed below bag union or difference.
- c) Projection cannot be pushed below set union.
- d) Projection cannot be pushed below set or bag difference.

Exercise 16.2.5: Starting with an expression $\pi_L(R(a, b, c) \bowtie S(b, c, d, e))$, push the projection down as far as it can go if L is:

- a) $a, b, a + d \rightarrow z$.
- b) $b + c \rightarrow x, c + d \rightarrow y$.

Exercise 16.2.6: When it is possible to push a selection to both arguments of a binary operator, we need to decide whether or not to do so. How would the existence of indexes on one of the arguments affect our choice? Consider, for instance, an expression $\sigma_C(R \cap S)$, where there is an index on S .