

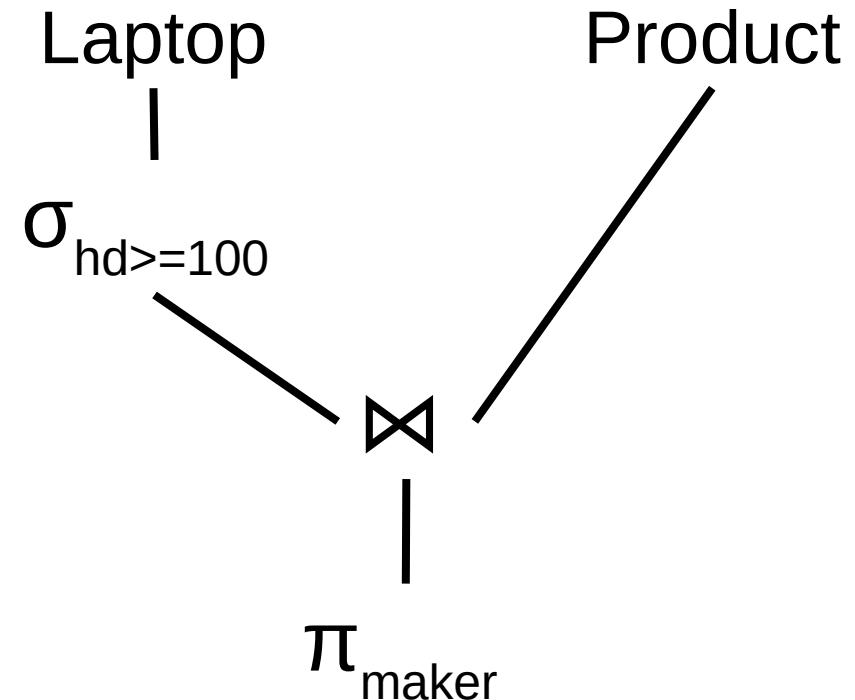
# Relational Algebra Query Execution Tree Examples

# Relational Algebra Operators

- Projection  $\pi$
- Selection  $\sigma$
- Duplicate elimination  $\delta$
- Sorting  $\tau$
- GroupBy aggregations  $\gamma$
- Set operations  $U, \cap, -$
- Product  $\times$
- Join  $\bowtie$
- Every operator takes as input one or two tables and generates as output a table
  - Schema
  - Tuples
- Operators are composable
  - The output of one operator is the input of another operator

# Relational Algebra Expressions $\leftrightarrow$ Query Execution Trees

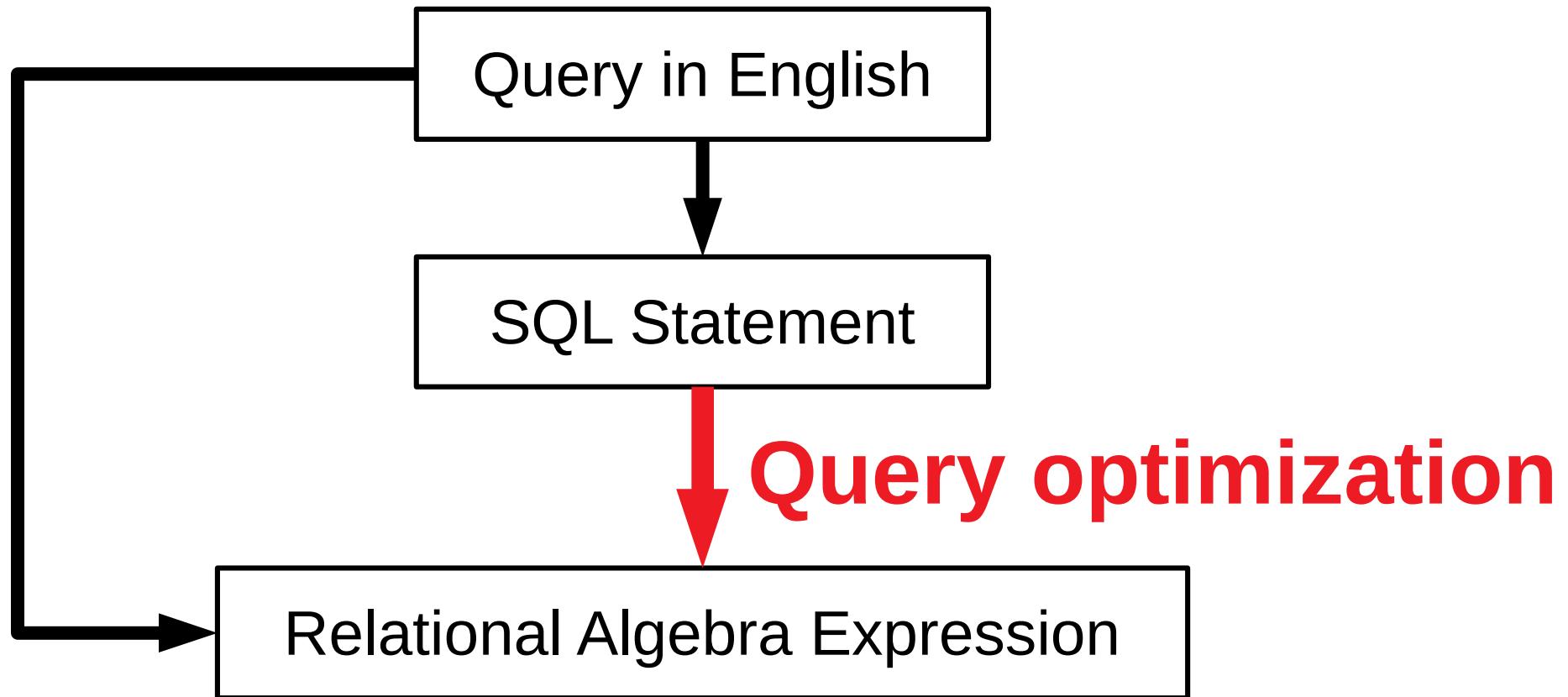
- $S_1(M, S, R, H, Sc, P) = \sigma_{H \geq 100}(\text{Laptop}(M, S, R, H, Sc, P))$
- $S_2(Ma, M, T, S, R, H, Sc, P) = \text{Product}(Ma, M, T) \bowtie S_1(M, S, R, H, Sc, P)$
- $R(\text{maker}) = \pi_{Ma}(S_2(Ma, M, T, S, R, H, Sc, P))$
- $R(\text{maker}) = \pi_{\text{maker}}(\text{Product} \bowtie \sigma_{hd \geq 100}(\text{Laptop}))$



# Relational Algebra $\leftrightarrow$ SQL

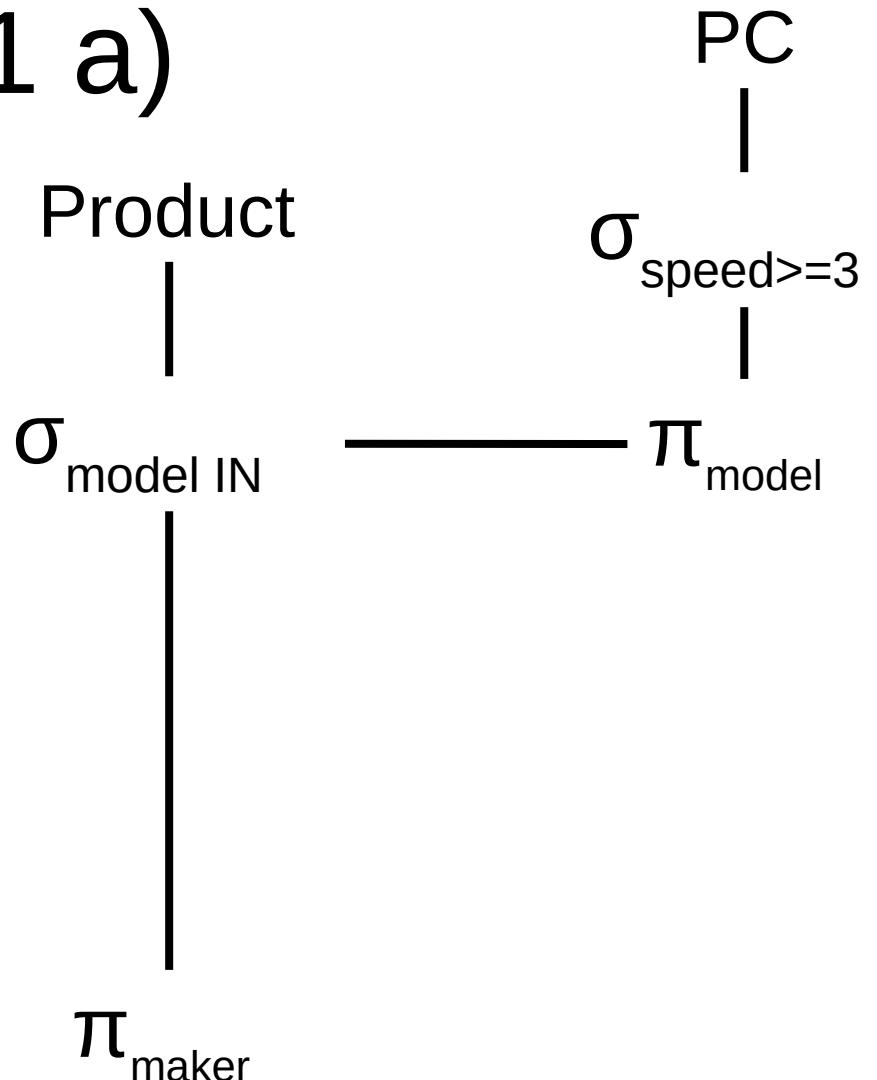
- SELECT  $\leftrightarrow$  Projection  $\pi$
- FROM  $\leftrightarrow$  Input tables
- WHERE  $\leftrightarrow$  Selection  $\sigma$ , Join predicates
- DISTINCT  $\leftrightarrow$  Duplicate elimination  $\delta$
- ORDER BY  $\leftrightarrow$  Sorting  $\tau$
- GROUP BY  $\leftrightarrow$  GroupBy aggregations  $\gamma$
- UNION, INTERSECT, EXCEPT  $\leftrightarrow$  Set operations  $U, \cap, -$
- JOIN  $\leftrightarrow$  Join

# From Queries (Through SQL) To Relational Algebra Expressions



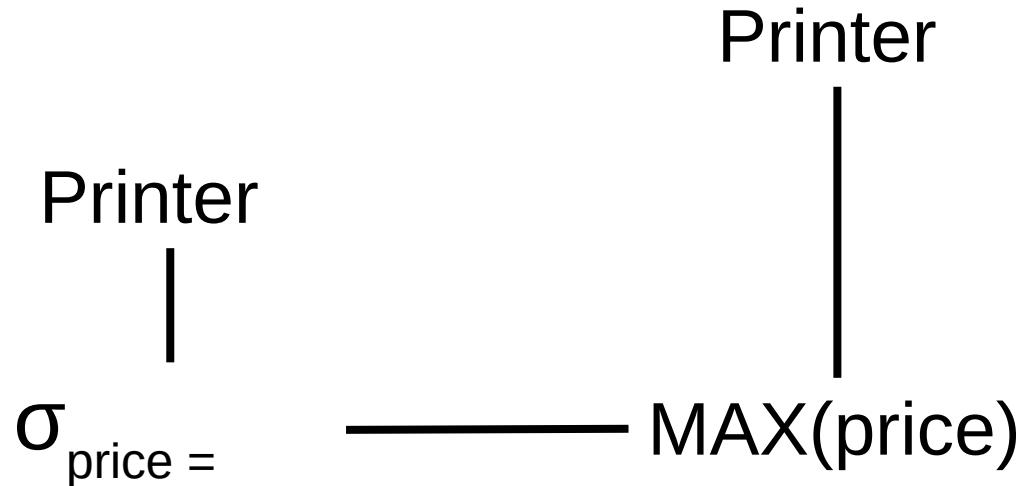
### 6.3.1 a)

select maker  
from Product  
where model in  
(select model  
from PC  
where speed >= 3)



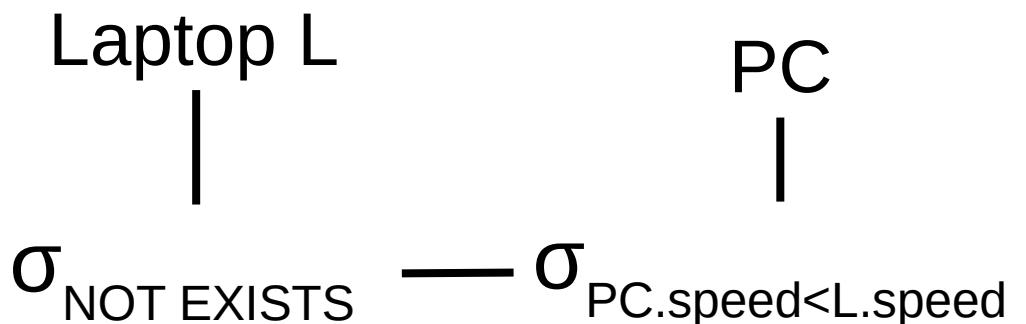
### 6.3.1 b)

```
select *  
from Printer  
where price =  
(select max(price)  
from Printer)
```

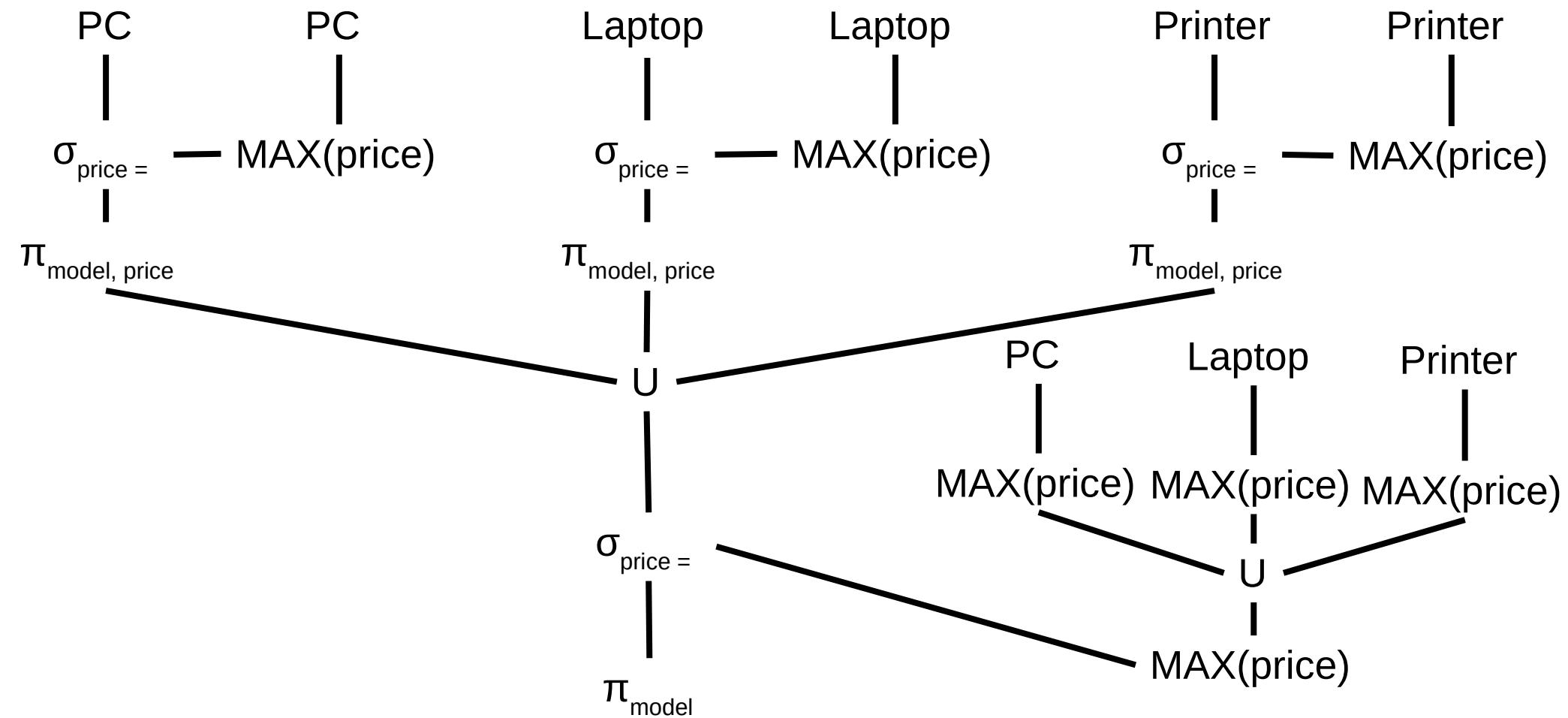


## 6.3.1 c)

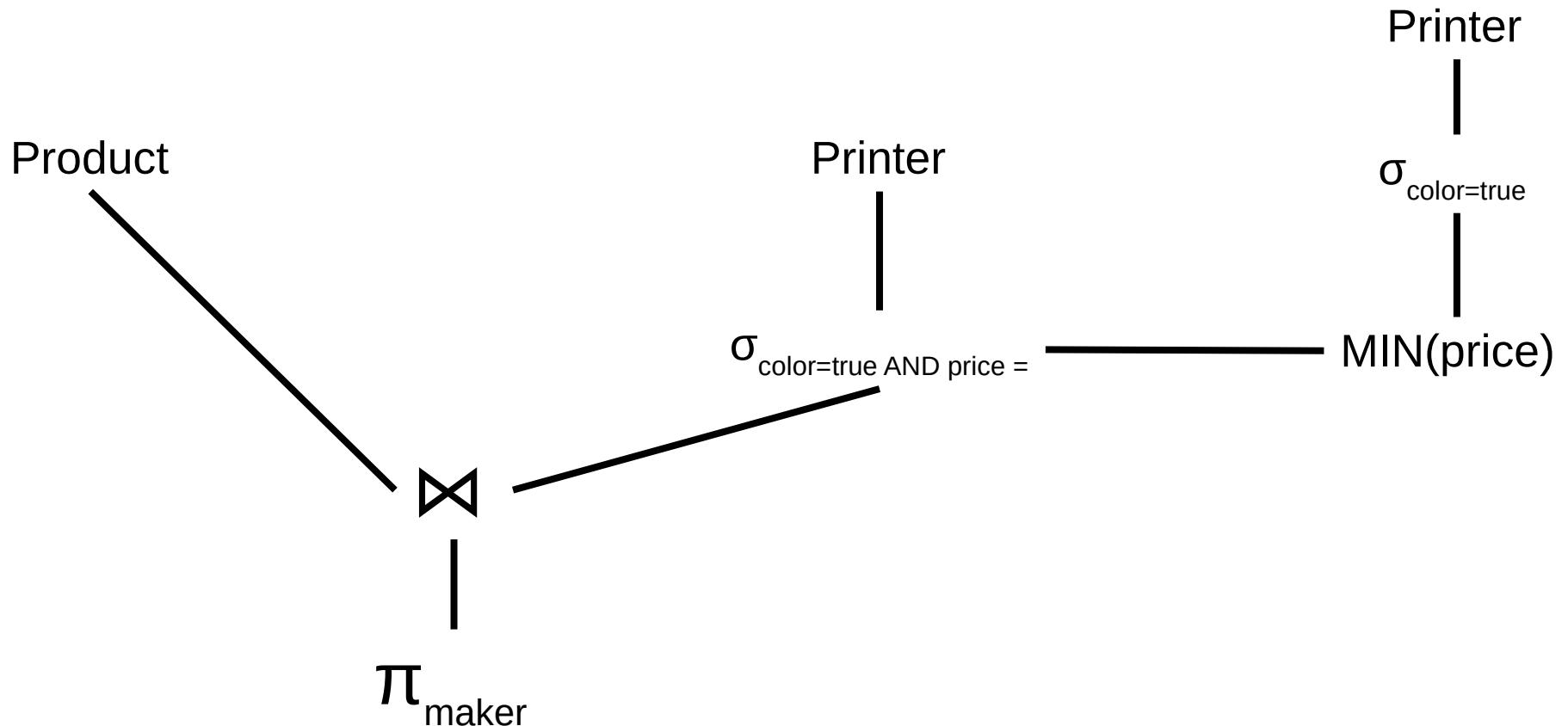
```
select *  
from Laptop L  
where not exists  
(select *  
from PC  
where PC.speed < L.speed)
```



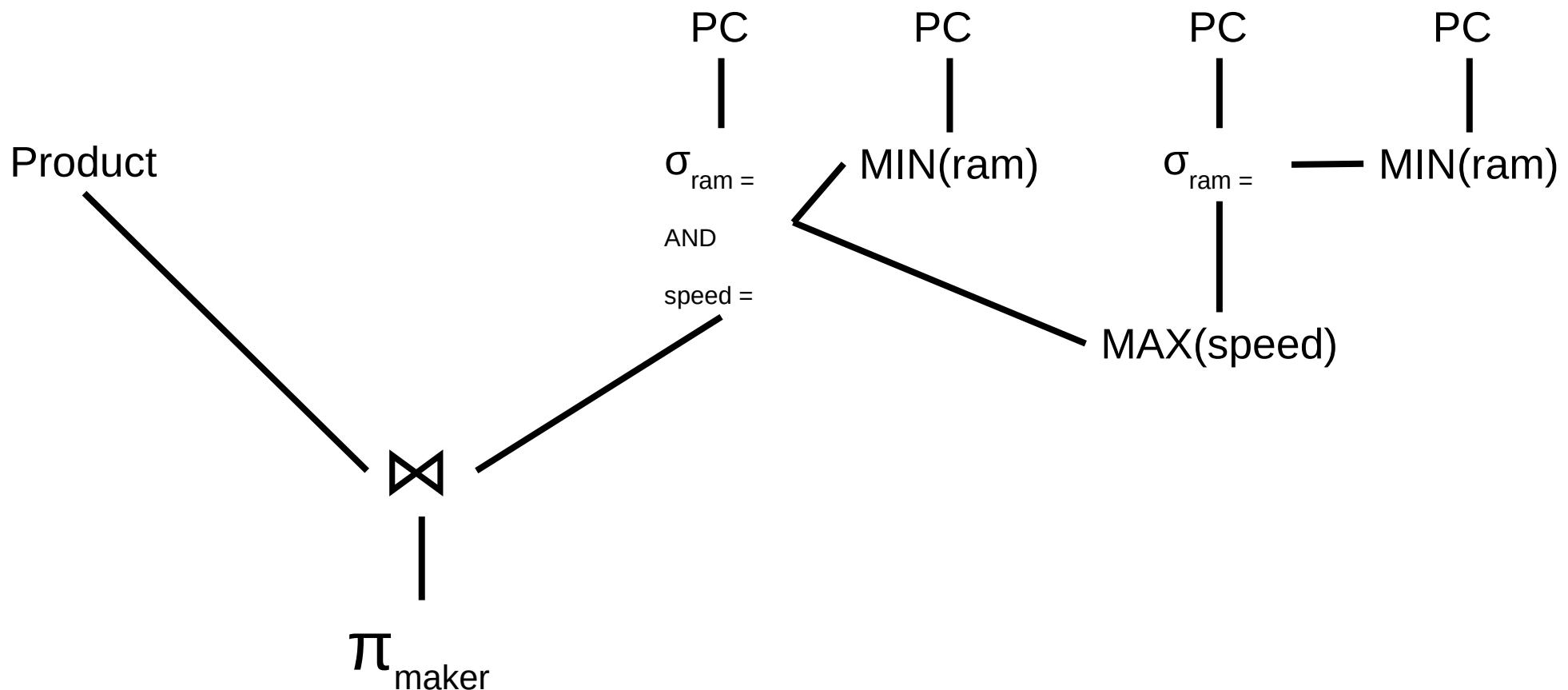
### 6.3.1 d)



### 6.3.1 e)



### 6.3.1 f)



## 6.4.6 a)

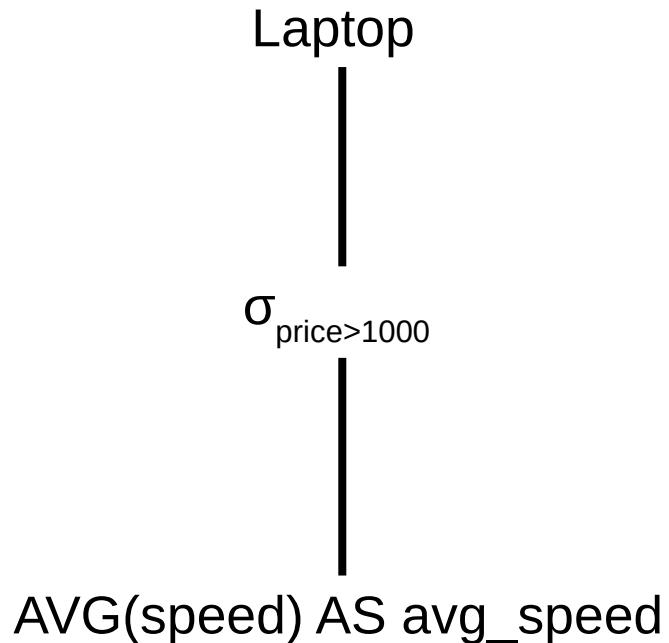
```
select avg(speed) as  
avg_speed
```

```
from pc
```

PC  
|  
AVG(speed) AS avg\_speed

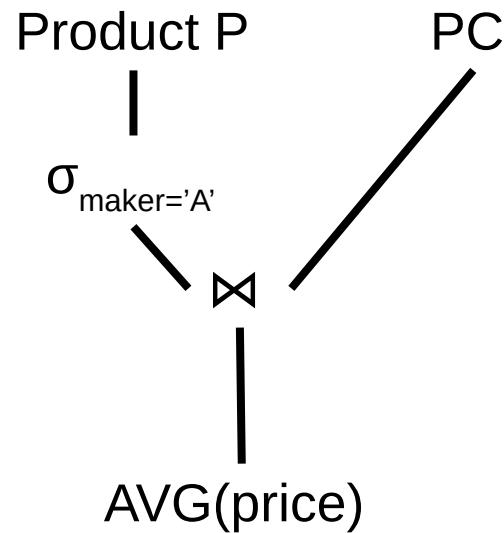
## 6.4.6 b)

```
select avg(speed) as  
avg_speed  
from laptop  
where price > 1000
```

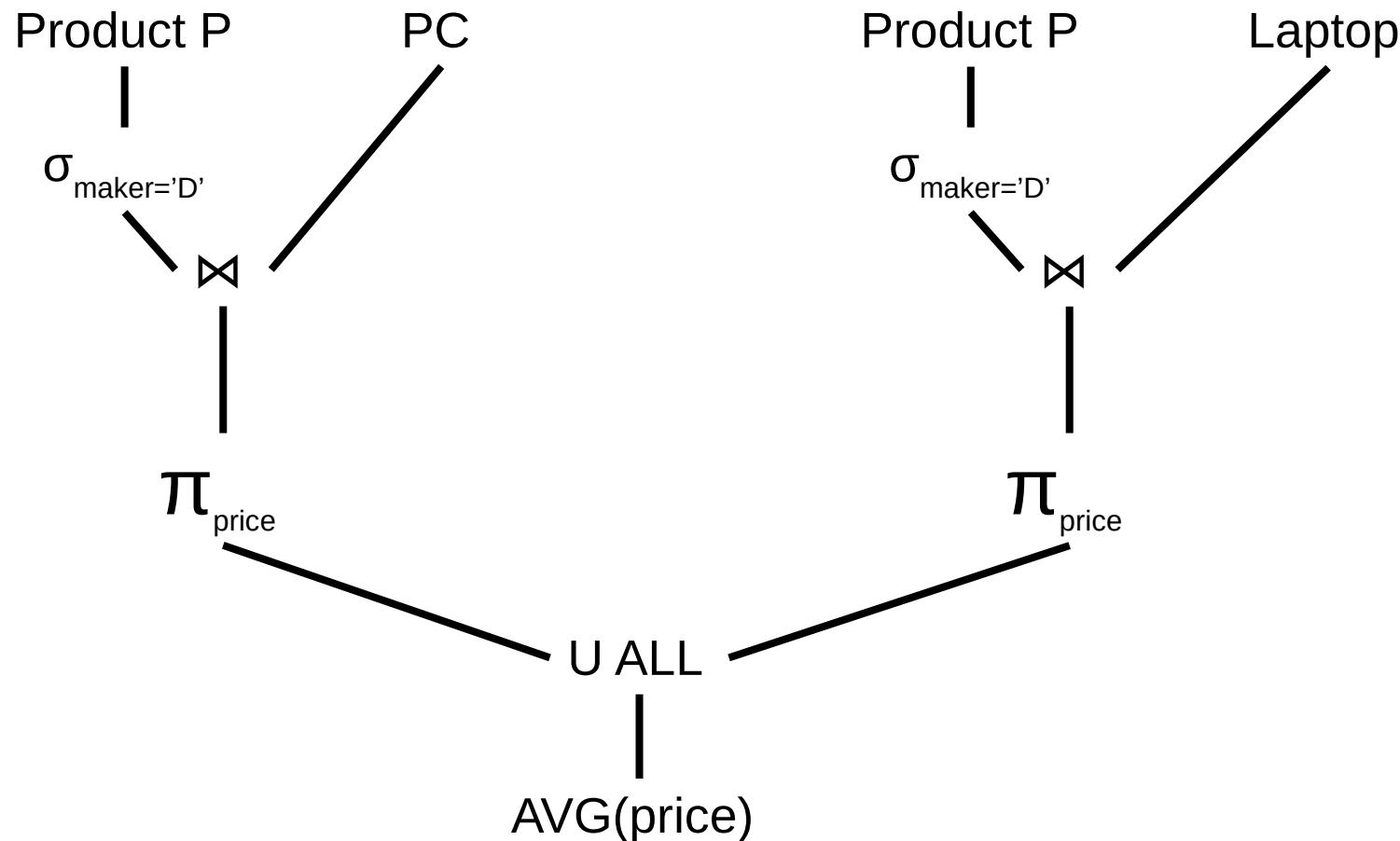


## 6.4.6 c)

```
select avg(price)  
from Product P, PC  
where P.model = PC.model AND  
P.maker = 'A'
```



# 6.4.6 d)



## 6.4.6 e)

```
select speed, avg(price) as  
avg_price
```

```
from pc
```

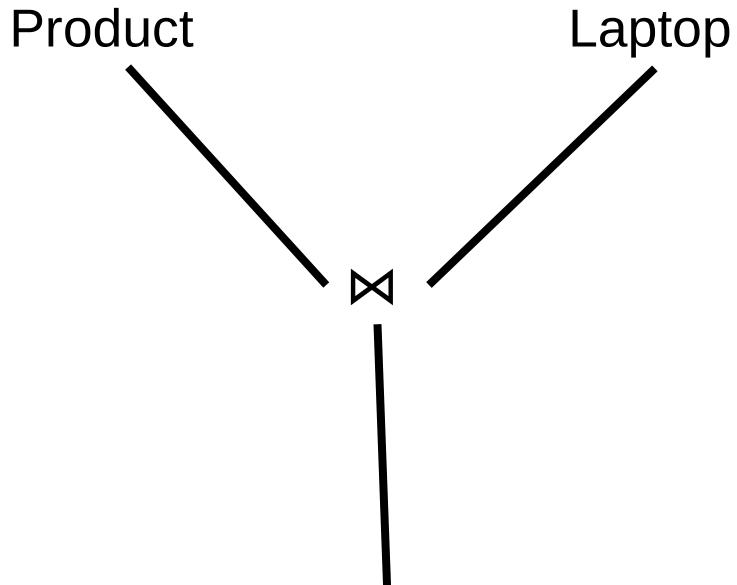
```
group by speed
```

PC

Y<sub>speed, AVG(price) AS avg\_price</sub>

## 6.4.6 f)

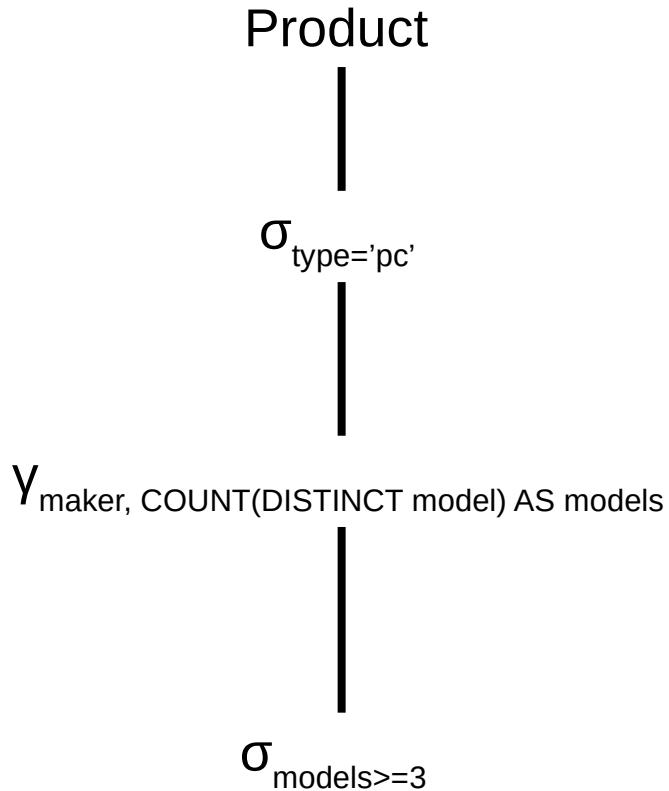
```
select maker, avg(screen) as  
avg_screen  
from Product P, Laptop L  
where P.model = L.model  
group by maker
```



$\gamma_{\text{maker}, \text{AVG}(\text{screen}) \text{ AS } \text{avg\_screen}}$

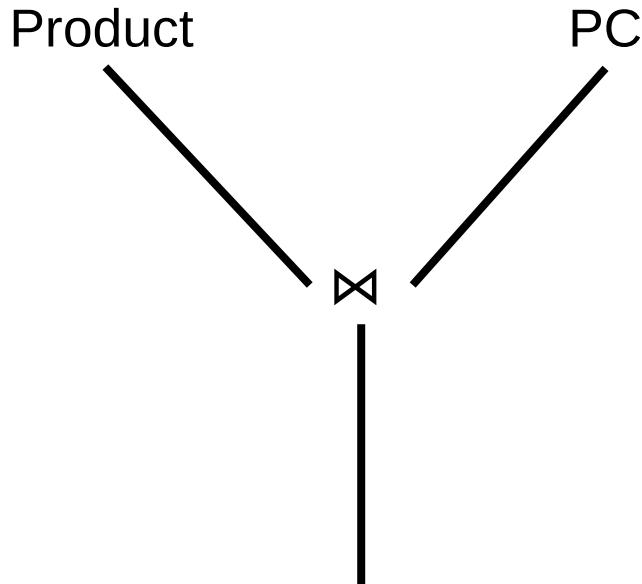
## 6.4.6 g)

```
select maker, count (distinct  
model) as models  
from product  
where type = 'pc'  
group by maker  
having models >= 3
```



## 6.4.6 h)

```
select maker, max(price) as  
max_price  
from Product P, PC  
where P.model = PC.model  
group by maker
```



$\gamma_{\text{maker}, \text{MAX(price)} \text{ AS } \text{max\_price}}$

## 6.4.6 i)

```
select speed, avg(price)  
as avg_price
```

```
from pc
```

```
where speed > 2
```

```
group by speed
```

PC

$\sigma_{\text{speed} > 2}$

$\gamma_{\text{speed, AVG(price) AS avg\_price}}$

## 6.4.6 j)

```
select maker, avg(hd) as avg_hd  
from Product P, PC  
where P.model = PC.model AND  
maker in (select maker  
          from Product  
          where type = 'printer')  
group by maker
```

