

1) π model (σ speed ≥ 3.00 (PC))

2) τ price desc (π price σ hd > 100 PC)

3) π maker (Product $\bowtie$ σ hd ≥ 100 (Laptop))

4) π maker (Product $\bowtie$ Laptop) - π maker (Product $\bowtie$ PC)

5) π X.model, Y.model ((ρ X (PC)) $\bowtie$ X.model > Y.model $\wedge$ X.ram = Y.ram $\wedge$ X.speed = Y.speed (ρ Y (PC)))

6) π PC.model, PC.price, equivlaptop $\leftarrow$ coalesce(Laptop.model, 'none'),
laptopprice $\leftarrow$ coalesce(Laptop.price, 'none') (PC $\bowtie$ PC.speed = Laptop.speed
Laptop)

7) π maker, model (Product $\bowtie$ (π model PC - π PC.model (PC $\bowtie$
PC.speed = Laptop.speed Laptop))) $\hat{=}$
 π maker, model (Product $\bowtie$ PC $\bowtie$ (π speed PC - π speed Laptop))

8)

$\gamma_{\text{type;avg(price)} \rightarrow \text{total(Product} \bowtie (\pi_{\text{model, price}}(\text{PC}) \cup \pi_{\text{model, price}}(\text{Laptop}) \cup \pi_{\text{model, price}}(\text{Printer})))$

9) $A = \pi_{\text{speed, model}}(\text{PC}) \cup \pi_{\text{speed, model}}(\text{Laptop})$
 $\pi_{\text{maker}}(\gamma_{\text{max(speed)} \rightarrow \text{speed}}(A) \bowtie A \bowtie \text{Product})$

Εναλλακτικά:

$$A = \pi_{\text{speed, model}}(\text{PC}) \cup \pi_{\text{speed, model}}(\text{Laptop})$$
$$\pi_{\text{maker}}(\text{Product} \bowtie$$
$$(A - \pi_{\text{r1.speed, r1.model}}(\sigma_{\text{r1.speed} < \text{r2.speed}}(\rho_{\text{r1}}(A) \times \rho_{\text{r2}}(A)))))$$

10) $\pi_{\text{maker, type}}(\text{Product}) \div \pi_{\text{type}}(\text{Product})$

Εναλλακτικά: $(\pi_{\text{maker}}(\sigma_{\text{type}='pc'}(\text{Product}))) \cap$
 $(\pi_{\text{maker}}(\sigma_{\text{type}='laptop'}(\text{Product}))) \cap$
 $(\pi_{\text{maker}}(\sigma_{\text{type}='printer'}(\text{Product})))$

Ωστόσο, η εναλλακτική αυτή έχει το μειονέκτημα ότι χρειάζεται τροποποίηση η έκφραση κάθε φορά που αλλάζουν τα δεδομένα της σχέσης Product όσον αφορά τους τύπους των προϊόντων. Για

παράδειγμα, αν προστεθούν προϊόντα ενός νέου τύπου (π.χ. 'scanner') θα χρειαστεί η έκφραση να επεκταθεί με: Π (π maker (σ type='printer' (Product)))

```
11) PC_Prices = πmaker, price(Product ⋈ PC)
Laptop_Prices = πmaker, price(Product ⋈ Laptop)
PC_Avg_Prices = γmaker; avg(price) → pc_price(PC_Prices)
Laptop_Avg_Prices = γmaker; avg(price) → lp_price(Laptop_Prices)
PC_Avg_Prices ⋈ Laptop_Avg_Prices
```

```
12) Fast_PC_Makers = (π maker (σ speed > 2.5 (Product ⋈ PC)))
MAxRAM = (γ maker; max(ram) → maxram(Product ⋈ PC))
AvgPrice = (γ maker; avg(price) → avgprice(Product ⋈ PC))
π maker, maxram, avgprice(MAxRAM ⋈ AvgPrice ⋈ Fast_PC_Makers)
```

Εναλλακτικά:

```
Fast_PC_Makers = (π maker (σ speed > 2.5 (Product ⋈ PC)))
γ maker; avg(price) → avgprice, max(ram) → maxram (PC ⋈ Product ⋈ Fast_PC_Makers)
```