

“Encontrar los proveedores que suministran a todos los proyectos de su misma ciudad”.  
La consulta no se resuelve mediante una división, sino mediante el siguiente razonamiento:

a) Encontramos todos los pares  $\text{codpro}, \text{codpj}$  que cumplan que son de la misma ciudad

$$\rho(\pi_{\text{codpro}, \text{codpj}}(\sigma_{S.\text{ciudad}=J.\text{ciudad}}(S \times J))) = SJ$$

b) Encontramos los suministros  $(\text{codpro}, \text{codpj})$  que cumplen que la ciudad del proveedor es la misma que la del proyecto

$$\rho(\pi_{\text{codpro}, \text{codpj}}(S \bowtie V \bowtie J)) = SVJ$$

c) Si a  $SJ$  le quitamos  $SVJ$  quedarán los pares  $\text{codpro}, \text{codpj}$  que no cumplen la condición y, por tanto, los  $\text{codpro}$  que no cumplen la condición, si a todos los  $\text{codpro}$  les quito esos me quedarán los que sí la cumplen.

$$\pi_{\text{codpro}}(S) - \pi_{\text{codpro}}(SJ - SVJ)$$

En cálculo quedaría de la siguiente manera:

{  $S1.\text{codpro} \mid s(S1)$  and not (exists  $J, S2, V$ ) ( $j(J)$  and  $s(S2)$  and  $spj(V)$  and  $J.\text{ciudad}=S2.\text{ciudad}$  and  $V.\text{codpro}=S2.\text{codpro}$  and  $V.\text{codpj}=J.\text{codpj}$  and  $V.\text{codpro}=S1.\text{codpro}$ ));

En SQL:

```
SELECT S1.codpro FROM proveedor S1 WHERE NOT EXISTS (
    SELECT * FROM proveedor S2, proyecto J WHERE J.ciudad=S2.ciudad AND NOT
    EXISTS(
        SELECT * FROM ventas V WHERE V.codpro=S2.codpro and V.codpj=J.codpj and
        V.codpro=S1.codpro
    )
)
ó
SELECT S1.codpro FROM proveedor S1 WHERE NOT EXISTS (
    SELECT J.codpj FROM proyecto J WHERE J.ciudad=S1.ciudad
    MINUS
    SELECT V.codpj FROM ventas V, proyecto J1 WHERE V.codpro=S1.codpro
        and J1.ciudad=S1.ciudad and V.codpj=J1.codpj
)
```