

## Práctica 6

### Objetivos:

- Unión, intersección y diferencia de sentencias de selección.
- Presentación ordenada de los resultados de una consulta.
- Aplicaciones y utilización de la operación de concatenación externa (*outer join*).

### Ejercicios guiados:

1. Facturas con IVA 16 o sin descuento (descuento cero o nulo).

**Resultado:** 295 filas.

**Ayuda:** Se puede hacer con el operador *union* o sin él (mejor sin el operador).

2. Códigos de los pueblos de la provincia de Castellón (código de provincia '12') donde no hay clientes.

**Resultado:** 519 filas.

**Ayuda:** Utilizar el operador *minus*.

3. Código, descripción, y precio de los artículos cuyo *stock* supera su *stock* mínimo multiplicado por 4. Presentar el resultado ordenado por precio en orden descendente, y dentro de un mismo precio, ordenado por código.

**Resultado:** 75 filas.

**Ayuda:** Operador *order by*.

4. Nombre de las provincias junto con el número de clientes que hay en ellas.

**Resultado:** 56 filas.

**Ayuda:** Utilizar el *join* externo dos veces, ya que puede haber provincias sin pueblos, y pueblos sin clientes.

### Ejercicios de la sesión:

5. Código de cada artículo cuyo *stock* supera las 20 unidades, con un precio superior a 15 euros, y de los que no hay ninguna factura en el último trimestre del año pasado.

**Resultado:** 5 filas.

**Ayuda:** Utilizar *minus*.

6. Para cada cliente de Burriana o de algún pueblo de la provincia de Valencia, mostrar el nombre del cliente y el de su pueblo.

**Resultado:** 15 filas.

**Ayuda:** Para evaluar ambas condiciones se necesitan tablas diferentes por lo que se pueden formular separadamente y luego unir los resultados.

7. Nombre de las provincias con más de 500 pueblos y en las que hay más de 5 pueblos con clientes.

**Resultado:** 3 filas.

**Ayuda:** Utilizar *intersect*.

8. Código de cada cliente y número de líneas de pedido que ha hecho, en el caso de que sean menos de 20 líneas.

**Resultado:** 85 filas.

**Ayuda:** Formular una consulta con dos *joins* externos, ya que hay clientes sin facturas y, por lo tanto, sin líneas de pedido.

9. Código y precio de cada artículo con un precio superior a 50 euros y que en alguna de sus líneas de pedido haya tenido un precio inferior a la mitad de su precio. Cada artículo se debe presentar con el mínimo de dichos precios, y los artículos sin líneas de pedido deben presentarse con un valor igual a cero. Ordenar el resultado por el precio del artículo en orden ascendente, y para aquellos artículos con igual precio, ordenarlos por el valor del precio mínimo por en orden descendente.

**Resultado:** 26 filas.

**Ayuda:** Aunque inicialmente, la consulta no se puede realizar con un *join* externo por las condiciones que hay sobre las columnas de las tablas concatenadas, el operador *nvl* permite localizar los nulos y formular la consulta.

## Ejercicios opcionales:

10. Códigos de los artículos tales que su *stock* esté por debajo del doble de su *stock* mínimo, y el número total de unidades vendidas sea mayor que 50.

**Resultado:** 3 filas.

**Ayuda:** Utilizar el operador *intersect*.

11. Código de cada artículo con un precio superior a 50 euros junto con la cantidad media que se suele hacer en cada línea de pedido de ese artículo. Los artículos sin líneas de pedido deben presentarse con una cantidad media igual a cero.

**Resultado:** 39 filas.

**Ayuda:** A pesar de las condiciones, se puede formular una consulta con un *join* externo.

12. Pueblos de la provincia de Castellón en donde no hay clientes, o donde los clientes que hay no han comprado nada durante el último trimestre del año pasado.

**Resultado:** 523 filas.

**Ayuda:** Los pueblos que buscamos son todos los de Castellón menos los que tienen algún cliente que haya comprado algo durante dicho periodo.

13. Código y nombre de los clientes que son de algún pueblo de Valencia y alguna vez se les ha realizado una factura con más de 9 líneas de factura.

**Resultado:** 4 filas.

**Ayuda:** Dado que las condiciones son sobre tablas diferentes, para no poner tantas tablas en el *from*, es más eficiente evaluarlas por separado y calcular su intersección.