

Solucións Boletín 3

■ Tarefa 3.1

```

/*****
Seleccionar os artigos de cor negra e mostrar o seu número, nome e peso, así como o
nome do proveedor.
*****/

select ar.art_codigo as Numero,
       ar.art_nome as Articulo,
       ar.art_peso as Peso,
       pr.prv_nome as Proveedor
from artigos as ar join provedores as pr on ar.art_proveedor=pr.prv_id
where ar.art_color='negro';

```

■ Tarefa 3.2

```

/*****
Seleccionar para todos os apelidos, nome e o nome da provincia na que residen. Os dous
primeiros díxitos do código postal (clt_cp) corresponden ao código da provincia na que
reside o cliente. Ordenar o resultado polo nome da provincia, e dentro da provincia,
polos apelidos e nome, alfabeticamente.
*****/

select cl.clt_apelidos, cl.clt_nome, pr.pro_nome
from clientes as cl join provincias as pr on left(trim(cl.clt_cp),2)=pr.pro_id
order by pr.pro_nome, cl.clt_apelidos, cl.clt_nome;

```

■ Tarefa 3.3

```

/*****
Mostrar para cada venda: nome e apelidos do cliente, día, mes, e ano da venda (cada
un nunha columna).
*****/

select cl.clt_nome as Nome_cliente,
       cl.clt_apelidos as Apelidos,
       day(ve.ven_data) as Dia_venta,
       month(ve.ven_data) as Mes_venta,
       year(ve.ven_data) as Ano_venta
from clientes as cl join vendas as ve on cl.clt_id=ve.ven_cliente;

```

■ Tarefa 3.4

```

/*****
Mostrar unha lista que conteña: número de vendas, número de artigos vendidos, suma de
unidades vendidas e a media dos prezos unitarios dos artigos vendidos. Facer os
cálculos coa información contida nas táboas vendas e detalle:vendas
*****/

select count(distinct ve.ven_id) as Numero_vendas,
       count(distinct dv.dev_artigo) as Numero_artigos,
       sum(dv.dev_cantidad) as Suma_unidades,
       round(avg(dv.dev_precio_unitario),2) as Media_precio
from vendas as ve join detalle_vendas as dv on dv.dev_venta = ve.ven_id;

```

■ Tarefa 3.5

```

/*****
Seleccionar para cada artigo o seu número, nome, peso e o nome que corresponde ao peso
(peso_nome), tendo en conta a información contida na táboa pesos, que da un nome aos
pesos en función do intervalo ao que pertence. Ordenar o resultado polo peso do artigo,
de maior a menor.
*****/

select ar.art_codigo,
       ar.art_nome,
       ar.art_peso,
       pe.peso_nome
from artigos as ar join pesos as pe on ar.art_peso between peso_min and peso_max
order by ar.art_peso;

```

■ Tarefa 3.6

```

/*****
Mostrar para cada venda: nome e apelidos do cliente, a data da venda con formato
dd/mm/aa e os días transcorridos dende que se fixo a venda. Ordenar o resultado polo
número de días transcorridos dende a venda.
*****/

select distinct cl.clt_apelidos as Apelidos,
               cl.clt_nome as Nome,
               date_format(ve.ven_data, '%d/%m/%Y') as Data_venta,
               datediff(curdate(),ve.ven_data) as Dias_diferencia
from vendas as ve join clientes as cl on ve.ven_cliente=cl.clt_id
order by Dias_diferencia;
```

■ Tarefa 3.7

```

/*****
Seleccionar os nomes das provincias nas que temos clientes.
*****/

select distinct pro_nome
from provincias as pr
      join clientes as cl on pr.pro_id = left(trim(cl.clt_cp),2);
```

■ Tarefa 3.8

```

/*****
Seleccionar para cada venda:
Datos da venda: identificador e data da venda.
Datos do cliente: nome do cliente (nome e apelidos separados por coma).
Datos do empregado: nome do empregado (nome e apelidos separados por coma).
Mostrar os datos ordenados polos apelidos e nome do cliente
*****/

select ve.ven_id as Numero_venta,
       ve.ven_data as Data_da_venta,
       concat(cl.clt_apelidos,', ',cl.clt_nome) as Apelidos_nome_cliente,
       concat(em.emp_apelidos,', ',em.emp_nome) as Apelidos_nome_empleado
from vendas as ve
      join clientes as cl on ve.ven_cliente = cl.clt_id
      join empregados as em on ve.ven_empleado = em.emp_id
order by Apelidos_nome_cliente;
```

■ Tarefa 3.9

```

/*****
Seleccionar información sobre os artigos vendidos. Para cada liña de detalle interesa:
- Datos do cliente: apelidos e nome separados por coma, nunha única columna.
- Datos do artigo: nome, cantidade, prezo unitario, desconto e o importe final para
o cliente (resultado de multiplicar a cantidade polo prezo unitario e aplicar o
desconto que corresponde).
Mostrar os resultados ordenados polo nome do artigo.
*****/

select concat(cl.clt_apelidos,', ',cl.clt_nome) as Apelidos_nome_cliente,
       ar.art_nome as Artigo,
       dv.dev_cantidade as Cantidade,
       dv.dev_prezo_unitario as Prezo_unitario,
       dv.dev_desconto as '% Desconto',
       round((dv.dev_prezo_unitario*dv.dev_cantidade)*(1-dv.dev_desconto/100),2) as Total
from vendas as ve
      join clientes as cl on ve.ven_cliente = cl.clt_id
      join detalle_vendas as dv on ve.ven_id = dv.dev_venta
      join artigos as ar on dv.dev_artigo = ar.art_codigo
order by Artigo;
```

■ Tarefa 3.10

```
/* Seleccionar o número e nome de departamento, xunto co nome do director, para os departamentos independentes, é dicir, que non dependen de ningún outro departamento.*/
select de.depNumero as Numero,
       de.depNome as Nome,
       em.empNome as Director
from departamento as de join empregado as em on de.depDirector=em.empNumero
where de.depDepende is null;
```

■ Tarefa 3.11

```
/* Mostrar nome (só nome, sen apelidos) e enderezo do centro ao que pertence o departamento no que traballa, dos empregados cun nome (sen ter en conta os apelidos) que empece por 'A'.*/
select trim(right(em.empNome, length(em.empNome)-locate(',', em.empNome))) as Nome,
       ce.cenEnderezo as Enderezo
from empregado as em
     join departamento as de on em.empDepartamento = de.depNumero
     join centro as ce on de.depCentro=ce.cenNumero
where trim(right(em.empNome, length(em.empNome)-locate(',', em.empNome))) like 'A%';
/* Ollo: A función TRIM elimina espazos en branco ao principio e ao final da cadea. Non é o mesmo utilizar right(em.empNome, length(em.empNome)-locate(',', em.empNome)-1) xa que hai un empregado que non ten espazo en branco despois da coma e ten A como segunda letra do nome: SANTOS,SANCHO */
```

■ Tarefa 3.12

```
/* Seleccionar para todos os empregados que non son directores, o nome de departamento no que traballa, o seu nome e salario, o nome e salario do director do seu departamento, e a diferenza do seu salario e o salario do director do departamento. Ordenar o resultado polo nome do departamento.*/
select de.depNome as Departamento,
       em1.empNome as Empregado,
       em1.empSalario as Salario_empregado,
       em2.empNome as Nome_Director,
       em2.empSalario as Salario_director,
       em2.empSalario-em1.empSalario as Diferenza
from empregado as em1 join departamento as de on em1.empDepartamento = de.depNumero
     join empregado as em2 on de.depDirector=em2.empNumero
where em1.empNumero <> em2.empNumero
order by de.depNome;
```

Solucións Boletín 4

■ Tarefa 4.1

```
/*
*****
Para todos os clientes con identificador inferior ou igual a 10, seleccionar os datos das vendas que se lle fixeron. Hai que mostrar para cada venda, o identificador do cliente, apelidos, nome e data de venda. Se a algún deses clientes non se lle fixo ningunha venda, deberá aparecer na lista co seu identificador, nome, apelidos, e o texto 'SEN COMPRAS' na columna da data da venda.
*****
*/
```

```
select distinct cl.clt_id as Codigo_cliente,
               cl.clt_apelidos as Apelidos,
               cl.clt_nome as Nome,
               ifnull(ve.ven_data,'SEN COMPRAS') as Data_compra
from clientes as cl left join vendas as ve on cl.clt_id=ve.ven_cliente
where cl.clt_id<=10;
```

■ Tarefa 4.2

```
/* *****
Seleccionar os nomes das provincias nas que non temos ningún cliente.
***** */

select pro_nome
from provincias as pr
left join clientes as cl on pr.pro_id = left(trim(cl.clt_cp),2)
where clt_nome is null;  /* Vale calquera columna da táboa clientes
```

■ Tarefa 4.3

```
/* *****
Tarefa 2.3. Seleccionar o código (emp_id), apelidos e nome de todos os empregados.
Engadir unha columna na lista de selección, co alias Vendas, na que se mostre o literal
'Si' se o empregado fixo algunha venda, e o literal 'Non' no caso de que aínda non
fixera ningunha venda.
***** */

select distinct emp_id, emp_dni, emp_apelidos, emp_nome,
if(ven_id is null,'Non','Si') as Vendas
from empregados left join vendas on emp_id = ven_empregado;
```

■ Tarefa 4.4

```
/* *****
Obter unha lista de todos os artigos que teñan un prezo de compra superior ao prezo
de compra do artigo con código '0713242'.
***** */

select ar1.art_codigo as Numero,
ar1.art_nome as Nome,
ar1.art_pc as Prezo_compra,
ar2.art_pc as Prezo_artigo_8
from artigos as ar1 join artigos as ar2 on ar1.art_pc>ar2.art_pc
where ar2.art_codigo='0713242';
```

■ Tarefa 4.5

```
/* Mostrar o número, nome e salario de todos os empregados que teñen un salario maior
que o do empregado número 180. Engadir na lista de selección unha columna para mostrar
o salario do empregado número 180.*/

select em1.empNumero, em1.empNome, em1.empSalario, em2.empSalario as 'Empregado 180'
from empregado as em1 join empregado as em2 on em1.empNumero != em2.empNumero
where em2.empNumero=180
and em1.empSalario > em2.empSalario
```

■ Tarefa 4.6

```
/* *****
Seleccionar todos os artigos negros, xunto cos artigos que pesan máis de 5000 gramos,
escribindo dúas consultas, e empregando o operador de unión de consultas
***** */

select art_codigo as Codigo,
art_nome as Nome,
art_peso as Peso,
art_color as Color
from artigos
where art_peso>5000
union
select art_codigo as Codigo,
art_nome as Nome,
art_peso as Peso,
art_color as Color
from artigos
where art_color="negro"
order by Codigo;
```

■ Tarefa 4.7

```

/*****
Para facer un envío de cartas con información dunha nova campaña por correo postal,
seleccionar apelidos, nome, enderezo, código postal e poboación de todos os clientes
e de todos os empregados. Na lista hai que diferenciar se a persoa é cliente ou
empregado. Ordenar o resultado por orden alfabético de apelidos e nome.
*****/

(select 'Cliente' as Persoa,
    cl.clt_apelidos as Apelidos,
    cl.clt_nome as Nome,
    cl.clt_enderezo as Enderezo,
    cl.clt_cp as CP,
    cl.clt_poboacion as Poboacion,
    pr.pro_nome
from clientes as cl
    join provincias as pr on left(trim(cl.clt_cp),2)=pr.pro_id
where clt_baixa is null)
union
(select 'Empregado' as Persoa,
    em.emp_apelidos as Apelidos,
    em.emp_nome as Nome,
    em.emp_enderezo as Enderezo,
    em.emp_cp as CP,
    em.emp_poboacion as Poboacion,
    pr.pro_nome as Provincia
from empregados as em
    join provincias as pr on left(trim(em.emp_cp),2)=pr.pro_id )
order by Apelidos, Nome;
```