



**Ejercicios Sesión 2:** Repaso de la sentencia SELECT sobre la BD TiendaOnLine

## Soluciones

---

1 Lista los email y nombre y apellidos de los usuarios ordenados por apellidos y nombre

```
select email,nombre,apellidos  
from usuario  
order by apellidos,nombre;
```

---

2 Toda la información (código y nombre) de las provincias de las que se tienen usuarios

```
select pv.*  
from usuario u, provincia pv  
where u.provincia=codp;
```

---

3 Toda la información (código y nombre) de las provincias de las que se tienen usuarios, eliminando duplicados y ordenando por nombre

```
select distinct pv.*  
from usuario u, provincia pv  
where u.provincia=codp  
order by pv.nombre;  
Solución alternativa:  
select distinct pv.*  
from usuario u, provincia pv  
where u.provincia=codp  
order by 2;
```

---

4 Artículos que no tienen marca

```
select * from articulo where marca is null
```

---

5 Email de los usuarios de la provincia de Murcia que no tienen teléfono, acompañado en la salida por un mensaje que diga "No tiene teléfono"

```
select email,'No tiene teléfono'  
from usuario u, provincia pv  
where u.provincia=codp and pv.nombre = 'Murcia'  
and telefono is null;
```

---

6 Fecha y usuario del pedido, código, nombre, marca, pvp y precio de venta de los artículos solicitados en el pedido número 1 que sean televisores.

```
select fecha, usuario, articulo, nombre, marca, pvp, precio
from linped l, articulo a, tv t, pedido p
where l.numpedido=1 and l.numpedido = p.numpedido
and a.cod=l.articulo and a.cod=t.cod
```

---

7 Email de los usuarios cuyo código postal no sea 02012, 02018 o 02032.

```
select email from usuario
where codpos not in ('02012','02018','02032');
```

---

8 Número de pedido, fecha y nombre y apellidos del usuario que solicita el pedido, para aquellos pedidos solicitados por algún usuario de apellido MARTINEZ.

```
select numpedido, fecha, nombre, apellidos
from pedido, usuario
where usuario=email and apellidos like '%MARTINEZ%'
```

---

9 Código, nombre y marca del artículo más caro.

```
select cod, nombre, marca, pvp
from articulo
where pvp = (select max(pvp) from articulo);
Solución alternativa:
select cod, nombre, marca, pvp
from articulo
where pvp >= all (select pvp from articulo);
```

---

10 Código, nombre y pvp de la cámara más cara de entre las de tipo réflex.

```
select a.cod, nombre, pvp
from articulo a, camara c
where a.cod=c.cod
and tipo like '%reflex%'
and pvp=(
    select max(pvp)
    from articulo a, camara c
    where a.cod=c.cod and tipo like '%reflex%');
Solución alternativa:
select a.cod, nombre, pvp
from articulo a, camara c
where a.cod=c.cod and tipo like '%réflex%'
and pvp >= all (select pvp from articulo a, camara c
    where a.cod=c.cod and tipo like '%réflex%');
```

---

11 Marcas de las que no existe ningún televisor en nuestra base de datos.

```
select marca from marca
where marca not in (select marca
                    from articulo a, tv t
                    where a.cod=t.cod);
```

Solución alternativa:

```
SELECT marca
FROM marca m
WHERE NOT EXISTS
  (SELECT 1 FROM articulo a, tv t
   where a.cod=t.cod and a.marca=m.marca);
```

---

12 Código, nombre, tipo y marca de las cámaras de marca Nikon, LG o Sigma.

```
select a.cod, nombre, tipo, marca
from articulo a, camara c
where a.cod=c.cod and marca in ('NIKON','LG','SIGMA');
```

---

13 Código y nombre de los artículos, si además es una cámara, mostrar también la resolución y el sensor.

```
select a.cod, nombre, resolucion, sensor
from articulo a
left join camara c on (a.cod = c.cod);
```

---

14 Muestra las cestas del año 2010 junto con el nombre del artículo al que referencia y su precio de venta al público.

```
select c.*, nombre, pvp
from cesta c
join articulo on (cod=articulo)
where to_char(fecha,'yyyy')= '2010';
```

Solución alternativa:

```
select c.*, nombre, pvp
from cesta c
join articulo on (cod=articulo and to_char(fecha,'yyyy')= '2010' );
```

---

15 Muestra toda la información de los artículos. Si alguno aparece en una cesta del año 2010 muestra esta información.

```
select a.*, c.*
from articulo a
left join cesta c on (articulo=cod and to_char(fecha,'yyyy')= '2010');
```



*A diferencia del ejercicio anterior, es obligatorio incluir la condición de la fecha en la condición del join puesto que queremos la información de todos los artículos y solo las cestas de 2010 en las que estén aquellos. Prueba lo siguiente:*

```
select a.*, c.*  
from articulo a  
left join cesta c on (articulo=cod)  
where to_char(fecha,'yyyy')= '2010';
```

---

16 Cantidad de usuarios de nuestra BD.

```
select count(*) usuarios from usuario;
```

---

17 Obtener la cantidad de provincias distintas de las que tenemos conocimiento de algún usuario.

```
select count(distinct provincia) provincias from usuario;
```

---

18 Tamaño máximo de pantalla para las televisiones.

```
select MAX(pantalla) maxPantalla from tv;
```

---

19 Fecha de nacimiento del usuario más viejo.

```
select MIN(nacido) from usuario;
```

---

20 Obtener el precio total por línea para el pedido 1, en la salida aparecerá los campos numlinea, articulo y el campo calculado total.

```
select linea, articulo, (precio*cantidad) total  
from linped  
where numPedido=1;
```

---

21 Número de pedido, fecha y nombre y apellidos del usuario de las líneas de pedido cuyo total en euros es el más alto.

```
select p.numPedido, fecha, nombre, apellidos  
from pedido p, linped l, usuario u  
where p.usuario=email  
and p.numPedido=l.numPedido  
and (cantidad*precio)=(select max(cantidad*precio) from linped);
```

---

22 ¿Cuántos artículos de cada marca hay?

```
select marca, count(*)  
from articulo  
group by marca;
```

---

23 Dni, nombre, apellidos y email de los usuarios que han realizado más de un pedido.

```
select dni,nombre,apellidos,email  
from usuario U, pedido P  
where U.email=P.Usuario  
group by dni,nombre,apellidos,email  
having count(*)>1;
```

---

24 Pedidos (número de pedido y usuario) que contengan más de cuatro artículos distintos.

```
select P.numPedido,P.usuario, count(distinct articulo)  
from pedido P, linped L  
where P.numPedido=L.numPedido  
group by P.numPedido,P.usuario  
having count(distinct articulo)>4;
```

---

25 Código y nombre de las provincias que tienen más de 50 usuarios (provincia del usuario, no de la dirección de envío).

```
select p.codp, p.nombre  
from provincia p, usuario u  
where p.codp=u.provincia  
group by p.codp, p.nombre  
having count(email)>50;
```

---

26 Cantidad de artículos que no son ni memoria, ni tv, ni objetivo, ni cámara ni pack.

```
select count(*)  
from articulo  
where cod not in (select cod from camara)  
and cod not in (select cod from tv)  
and cod not in (select cod from memoria)  
and cod not in (select cod from objetivo)  
and cod not in (select cod from pack);
```

Solución alternativa:

```
select count(*)  
from articulo  
where cod not in (select cod from camara  
union select cod from tv  
union select cod from memoria  
union select cod from objetivo  
union select cod from pack);
```

---

27 ¿En cuántos pedidos se ha solicitado cada artículo? Si hubiese artículos que no se han incluido en pedido alguno también se mostrarán. Mostrar el código y nombre del artículo junto con las veces que ha sido incluido en un pedido (solo si ha sido incluido, no se trata de la "cantidad").

```
select cod, nombre, count(distinct numpedido)
```

from articulo a  
left join linped l on (a.cod=l.articulo)  
group by cod, nombre;

*El "truco" consiste en contar los números de pedido asociados a algunos artículos. El left join hace que ciertos artículos aparezcan una vez junto con NULL como numpedido. El resto de artículos aparecen tantas veces como líneas tengan en LINPED. Como count(numpedido) solo cuenta los valores distintos de NULL, conseguimos que la cuenta sea 0 en los artículos no pedidos nunca.*

---

28 Pedidos (número de pedido y usuario) de importe mayor a 4000 euros.

```
select p.numpedido,p.usuario  
from pedido p, linped l  
where p.numpedido=l.numpedido  
group by p.numpedido,p.usuario  
having sum(cantidad*precio)>4000;
```

*Cuando nos piden el importe (total) de un pedido, se entiende que es el precio unitario de cada línea de ese pedido multiplicado por la cantidad pedida de ese artículo.*

---

29 Código y precio de los artículos 'Samsung' que tengan pvp y que no tengan pedidos.

```
SELECT distinct a.cod, marca, pvp  
FROM articulo a  
WHERE marca = 'Samsung' AND pvp IS NOT NULL AND a.cod NOT IN  
(select articulo from linped);
```

---

30 Códigos de artículos que están en alguna cesta o en alguna línea de pedido.

```
select articulo from cesta  
union  
select articulo from linped;
```

Solución alternativa 1:

```
select cod  
from articulo  
where cod in (select articulo from cesta)  
or cod in (select articulo from linped);
```

Solución alternativa 2:

```
select cod  
from articulo  
where exists (select 1 from cesta where articulo=cod)  
or exists (select 1 from linped where articulo=cod);
```

---

31 Email y nombre de los usuarios que no han hecho ningún pedido o que han hecho sólo uno.



```
select email, nombre
from usuario
where email not in (select usuario from pedido)
union
select email, nombre
from usuario, pedido
where email=usuario
group by email, nombre
having count(*)=1;
```

---

### 32 Email y nombre de los usuarios que no han pedido ninguna cámara.

```
select email, nombre
from usuario u
where not exists
(select 1
 from linped l, pedido p, camara c
 where email=p.usuario
   and l.numpedido=p.numpedido and c.cod=l.articulo);
```

Solución alternativa:

```
select email, nombre
from usuario u
where email not in
(select usuario
 from linped l, pedido p, camara c
 where l.numpedido=p.numpedido and l.articulo = c.cod);
```

---

### 33 Fecha actual.

```
select sysdate
from dual
```