



| BASES DE DATOS - 20/02/2024 (EXAMEN A) | |   |
|--|---------------|---|
| DA1D1E | 2ª EVALUACIÓN | |
| NOMBRE | | |

INDICACIONES:

- NO ESTÁN ORDENADOS EN NIVEL DE DIFICULTAD.
- TODAS LAS PREGUNTAS SE HACEN CON UNA SOLA SENTENCIA SQL Y UTILIZANDO ÚNICAMENTE LOS DATOS PROPORCIONADOS EN EL ENUNCIADO DE CADA EJERCICIO.
- En las consultas que no se pide un orden específico, con el fin de poder comprobar mejor los resultados, se puede poner el orden que se desee (o ninguno).
- Crear un usuario EXAMEN, password EXAMEN. Crea una conexión con este usuario y lanza el script de Viveros.sql.
- Las 11 primeras selects valen 0,9 puntos cada una. La select del ejercicio nº 12 vale 0,1.

Ejercicio 1.

Haz un listado de los productos cuyo precio sea mayor que 10€ y para los que haya menos de 400 unidades de existencias.

```
Select * from PRODUCTO p2
where p2.STOCK < 400
AND p2.PRECIOVENTA > 10
```

Ejercicio 2.

Muestra un listado con el nombre de los productos, la cantidad de pedidos de cada producto y el número total de unidades que se ha vendido de cada uno para los productos. El listado debe estar ordenado por el número total de unidades vendidas de forma descendente.

```
SELECT p.NOMBRE , count(dp.CODPEDIDO), sum(dp.CANTIDAD)
FROM PRODUCTO p, DETALLE_PEDIDO dp
WHERE p.CODPRODUCTO = dp.CODPRODUCTO
GROUP BY p.NOMBRE ORDER BY 3 desc
```

Ejercicio 3.

Devuelve un listado que muestre todos los empleados (nombre y apellidos) y en caso que tenga clientes asociados, queremos ver el nombre y teléfono de estos. Ordena el listado alfabéticamente por el nombre del empleado.

```
SELECT e.NOMBRE, c.NOMBRECLIENTE
```

```
FROM EMPLEADO e
LEFT JOIN cliente c ON c.CODEMPLEADOVENTAS = e.CODEMPLEADO ;
```

Ejercicio 4.

Para los pedidos del 2020, muestra un listado con el nombre de los clientes a los que no se les ha entregado a tiempo un pedido. Muestra nombre del cliente, código de pedido y la cantidad de días que se ha retrasado el pedido.

```
SELECT c.NOMBRECLIENTE , p.CODPEDIDO , p.FECHAENTREGA -
p.FECHAPREVISTA
FROM cliente c, PEDIDO p
WHERE c.CODCLIENTE = p.CODCLIENTE
AND to_char(p.FECHAPEDIDO,'yyyy') = 2020
AND p.FECHAPREVISTA < p.FECHAENTREGA
```

Ejercicio 5.

Haz un listado de los clientes en el que vemos su nombre y dos columnas, una que me indique si ha hecho pedidos (S / N). El listado se debe ordenar por nombre.

| | ABC NOMBRECLIENTE | ABC PEDIDOS |
|---|---------------------------|-------------|
| 1 | Agrojardin | S |
| 2 | Aloha | N |
| 3 | Beragua | S |
| 4 | Cabuts Golf Management SL | N |
| 5 | Campofeliz | N |
| 6 | Chingona Jardines S.L. | S |
| 7 | Club Golf El Canal | N |
| 8 | DaraDistribuciones | N |
| 9 | Dardena S.A. | S |

```
SELECT c.NOMBRECLIENTE , 'S' PEDIDOS
from CLIENTE c
WHERE c.CODCLIENTE IN (SELECT DISTINCT pe.CODCLIENTE FROM pedido pe)
UNION all
SELECT c.NOMBRECLIENTE , 'N' PEDIDOS
from CLIENTE c
WHERE c.CODCLIENTE not IN (SELECT DISTINCT pe.CODCLIENTE FROM pedido pe)

ORDER BY 1
```

Ejercicio 6.

Para países en los que haya más de un cliente queremos ver, el país, el total de distintos productos que han pedido y el total del ingreso para ese país.

```
SELECT c.PAIS ,  
       (SELECT count(DISTINCT dp.CODPRODUCTO)  
        FROM DETALLE_PEDIDO dp, PEDIDO p, cliente c3  
        WHERE dp.CODPEDIDO= p.CODPEDIDO  
        AND p.CODCLIENTE=c3.CODCLIENTE  
        AND c3.PAIS = c.pais ) distintos_productos,  
       nvl((SELECT sum(dp.CANTIDAD*dp.PRECIOUNIDAD)  
        FROM DETALLE_PEDIDO dp, PEDIDO p, cliente c3  
        WHERE dp.CODPEDIDO= p.CODPEDIDO  
        AND p.CODCLIENTE=c3.CODCLIENTE  
        AND c3.PAIS = c.pais),0) total_ingreso  
FROM cliente c  
GROUP BY c.PAIS  
HAVING count(1) > 1
```

Ejercicio 7.

Queremos saber el número total de productos para los que hay más de 5€ de margen de beneficio por cada unidad.

```
SELECT count(*)  
FROM PRODUCTO p  
WHERE p.PRECIOVENTA - p.PRECIOPROVEEDOR > 5
```

Ejercicio 8.

Queremos ver, agrupado por país y ciudad, el total de pagos. Ordena la consulta por país ascendente y luego por el total de pagos descendente.

```
SELECT c.PAIS , c.CIUDAD , sum(p.IMPORTETOTAL)  
FROM CLIENTE c, Pago p  
WHERE c.CODCLIENTE = p.CODCLIENTE  
GROUP BY c.PAIS , c.CIUDAD  
ORDER BY 1, 3 desc
```

Ejercicio 9.

Queremos un informe que dé el tipo de producto y el producto más caro (precio de venta) de ese tipo de producto.

```
SELECT t.TIPO ,t.DESCRIPCION_TEXTO, p.NOMBRE , p.PRECIOVENTA
```

```

FROM TIPOPRODUCTO t , PRODUCTO p
WHERE p.TIPOPRODUCTO = t.TIPO
AND p.PRECIOVENTA = (SELECT max(p2.precioventa)
                      FROM PRODUCTO p2
                      WHERE p2.TIPOPRODUCTO=t.TIPO)

```

Ejercicio 10.

Haz un listado de los años para los que hubo una facturación de más de 71.000€ (suponiendo que se factura en el año de la fecha del pedido) en productos que no sean de tipo 'Aromáticas'.

```

SELECT to_char(p.FECHAPEDIDO, 'yyyy') anyo, sum(dp.CANTIDAD*dp.PRECIOUNIDAD)
FROM DETALLE_PEDIDO dp, pedido p, PRODUCTO p2
WHERE p.CODPEDIDO = dp.CODPEDIDO
AND dp.CODPRODUCTO = p2.CODPRODUCTO
AND p2.TIPOPRODUCTO != 'Aromáticas'
GROUP BY to_char(p.FECHAPEDIDO, 'yyyy')
HAVING sum(dp.CANTIDAD*dp.PRECIOUNIDAD)> 71000

```

Ejercicio 11.

Devuelve las tiendas donde no trabaja ninguno de los empleados que tengan algún cliente que haya realizado la compra de algún producto del tipo 'Frutales'.

```

select *
from tienda t
where t.CODTIENDA not IN ( select distinct e2.CODTIENDA
                           from cliente c, PEDIDO p , DETALLE_PEDIDO dp,
                           PRODUCTO p2 , empleado e2
                           WHERE c.CODCLIENTE = p.CODCLIENTE
                           AND p.CODPEDIDO = dp.CODPEDIDO
                           AND dp.CODPRODUCTO = p2.CODPRODUCTO
                           AND c.codempleadoventas = e2.CODEMPLEADO
                           AND p2.TipoProducto='Frutales')

```

Ejercicio 12 - Bonus track.

Queremos ver los empleados que trabajan en tiendas cuyo promedio de pagos está por encima del promedio general de pagos. Además del nombre de los empleados queremos ver el total de clientes que tiene asignado cada empleado.

```

SELECT e3.NOMBRE , (SELECT count(1) FROM cliente c3 WHERE
c3.CODEMPLEADOVENTAS = e3.CODEMPLEADO) total_clientes
FROM EMPLEADO e3,
      (SELECT t.CODTIENDA tienda

```

```

FROM tienda t, EMPLEADO e , CLIENTE c, Pago p
WHERE t.CODTIENDA = e.CODTIENDA
AND e.CODEMPLEADO = c.CODEMPLEADOVENTAS
AND c.CODCLIENTE = p.CODCLIENTE
GROUP BY t.CODTIENDA
HAVING avg(p.IMPORTETOTAL) > (SELECT avg(p2.IMPORTETOTAL)
                                FROM tienda t2, EMPLEADO e2 , CLIENTE c2, Pago p2
                                WHERE t2.CODTIENDA = e2.CODTIENDA
                                AND e2.CODEMPLEADO = c2.CODEMPLEADOVENTAS
                                AND c2.CODCLIENTE = p2.CODCLIENTE )) tabla
WHERE e3.CODTIENDA = tabla.tienda

```