



BASES DE DATOS - 20/02/2024 (EXAMEN B)		 
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 facturaciones.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 menor que 15€ y para los que haya más de 100 unidades de existencias.

```
Select * from productos p
where p.existencias > 100
and p.precioxunidad < 15;
```

Ejercicio 2.

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) y otra que me indique si ha hecho pagos (S / N). El listado se debe ordenar por el campo de si ha hecho pedidos.

	ABC NOMBRE	ABC PEDIDOS
1	FISSA Fabrica Inter. Salchichas S	N
2	Paris specialites	N
3	The Big Cheese	S
4	Du monde entier	S
5	Die Wandernde Kuh	S
6	QUICK-Stop	S
7	Rattlesnake Canyon Grocery	S

```

SELECT c.NOMBRE , 'S' PEDIDOS
from CLIENTEs c
WHERE c.idcliente IN (SELECT DISTINCT pe.IDCLIENTE FROM pedidos pe)
UNION ALL
SELECT c.NOMBRE , 'N' PEDIDOS
from CLIENTEs c
WHERE c.idcliente not IN (SELECT DISTINCT pe.IDCLIENTE FROM pedidos pe)
ORDER BY 2

```

Ejercicio 3.

Saca los clientes que no han hecho ningún pedido de productos de la categoría de 'Reposteria'

```

SELECT *
FROM clientes c2
WHERE c2.IDCLIENTE NOT IN (SELECT DISTINCT p.IDCLIENTE
    FROM pedidos p, DETALLE_PEDIDOS dp , PRODUCTOS p2 , CATEGORIAS c
    WHERE p.IDPEDIDO = dp.IDPEDIDO
    AND dp.IDPRODUCTO = p2.IDPRODUCTO
    AND p2.IDCATEGORIA = c.IDCATEGORIA
    AND c.NOMBRE like 'Reposteria')

```

Ejercicio 4.

Haz una consulta que nos diga por cada producto(nombre del producto) el pedido más antiguo y la cantidad media que se pide de ese producto (redondeado con 1 decimal).

```

Select pr.nombre, min(p.fechapedido), round(avg(dp.cantidad),1)
from pedidos p, detalle_pedidos dp, productos pr
where p.idpedido = dp.idpedido
and dp.idproducto = pr.idproducto
group by pr.nombre;

```

Ejercicio 5.

Devuelve un listado que muestre todos los clientes (nombre) y en caso que tengan pedidos, queremos ver la fecha del pedido y el destinatario. Ordena el listado alfabéticamente por el nombre del cliente.

```

SELECT c.NOMBRE , p.FECHAPEDIDO , p.DESTINATARIO
FROM CLIENTES c
    LEFT JOIN PEDIDOS p ON c.IDCLIENTE = p.IDCLIENTE
ORDER BY 1 asc

```

Ejercicio 6.

Queremos saber el número total de productos para los que hay más de 100 unidades en existencias.

```
SELECT count(*)  
FROM PRODUCTOS p  
WHERE p.EXISTENCIAS > 100;
```

Ejercicio 7.

Haz una consulta que me devuelva el nombre del cliente, la fecha de pedido y la fecha de entrega para pedidos hechos en los meses de enero y febrero y que hayan tardado menos de 15 días en entregarlos..

```
Select c.nombre, p.FECHAPEDIDO , p.FECHAENTREGA  
from clientes c, pedidos p  
where c.idcliente = p.idcliente  
and to_char(p.fechapedido, 'mm') in ('01', '02')  
AND p.FECHAENTREGA - p.FECHAPEDIDO < 15
```

Ejercicio 8.

Haz un listado de los años para los que hubo una facturación de más de 200.000€ (suponiendo que se factura en el año de la fecha del pedido) en productos que no pertenezcan a la categoría de 'Bebidas'.

```
SELECT to_char(p.FECHAPEDIDO , 'yyyy') anyo,  
sum(dp.PRECIOXUNIDAD*dp.CANTIDAD)  
FROM DETALLE_PEDIDOS dp, pedidos p, PRODUCTOS p2, CATEGORIAS c  
WHERE p.IDPEDIDO = dp.IDPEDIDO  
AND dp.IDPRODUCTO = p2.IDPRODUCTO  
AND c.IDCATEGORIA = p2.IDCATEGORIA  
AND c.NOMBRE != 'Bebidas'  
GROUP BY to_char(p.FECHAPEDIDO, 'yyyy')  
HAVING sum(dp.PRECIOXUNIDAD*dp.CANTIDAD)> 200000
```

Ejercicio 9.

Haz un listado con los proveedores, el producto más barato del proveedor y el precio de este producto.

```
Select pv.nombre, pr.nombre, pr.PRECIOXUNIDAD
```

```

from productos pr, proveedores pv
where pr.idproveedor = pv.idproveedor
and pr.precioxunidad = (Select min(pr2.precioxunidad)
                        from productos pr2
                        where pr.idproveedor = pr2.idproveedor);

```

Ejercicio 10.

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

```

SELECT c.PAIS , c.CIUDAD , sum(p.CARGO)
FROM CLIENTES c, PEDIDOS p
WHERE c.IDCLIENTE = p.IDCLIENTE
GROUP BY c.PAIS , c.CIUDAD
ORDER BY 1, 3 desc

```

Ejercicio 11.

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.IDPRODUCTO)
       FROM DETALLE_PEDIDOS dp, PEDIDOS p, clientes c3
       WHERE dp.IDPEDIDO = p.IDPEDIDO
       AND p.IDCLIENTE =c3.IDCLIENTE
       AND c3.PAIS = c.pais ) distintos_productos,
      nvl((SELECT sum(dp.CANTIDAD *dp.PRECIOXUNIDAD)
           FROM DETALLE_PEDIDOS dp, PEDIDOS p, clientes c3
           WHERE dp.IDPEDIDO = p.IDPEDIDO
           AND p.IDCLIENTE =c3.IDCLIENTE
           AND c3.PAIS = c.pais),0) total_ingreso
FROM clientes c
GROUP BY c.PAIS
HAVING count(1) > 1

```

Ejercicio 12 - Bonus track.

Queremos ver los productos que pertenecen a proveedores cuyo promedio de pedidos (promedio de la cantidad * precio unidad de cada detalle de pedidos) está por encima del promedio general de pedidos (nuevamente promedio de cantidad * precio unidad de detalle de pedidos). Además del nombre de los productos queremos ver el total de pedidos que se han hecho de ese producto.

```

SELECT p.NOMBRE ,
      (SELECT count(dp2.IDPEDIDO) FROM DETALLE_PEDIDOS dp2 WHERE
dp2.IDPRODUCTO = p.IDPRODUCTO ) total_pedidos
FROM PRODUCTOS p,

```

```
(SELECT p.IDPROVEEDOR proveedor
      FROM PROVEEDORES p, PRODUCTOS p2 , DETALLE_PEDIDOS dp
      WHERE p.IDPROVEEDOR = p2.IDPROVEEDOR
      AND p2.IDPRODUCTO = dp.IDPRODUCTO
      GROUP BY p.IDPROVEEDOR
      HAVING avg(dp.PRECIOXUNIDAD*dp.CANTIDAD) > (SELECT
      avg(dp.PRECIOXUNIDAD*dp.CANTIDAD)
      FROM DETALLE_PEDIDOS dp )) Tabla
WHERE p.IDPROVEEDOR = tabla.proveedor
```