

**CUESTIONES SQL****(1 punto)**

- 1.- Cuando se realiza una venta directa a un cliente no habitual, no hace falta almacenar todos sus datos, por lo que el valor de codcli en la factura correspondiente es igual a NULL. Según éste comentario, ¿la siguiente sentencia SQL responde a la consulta “facturación del año 1995, sin tener en cuenta descuentos e iva, que incluya las ventas directas y las ventas realizadas a los clientes de la provincia de Castellón?”. Razona la respuesta sin ambigüedad. (0.5 puntos)

```
SELECT SUM(cant*precio)
FROM lineas_fac lin, facturas fac, clientes cli
WHERE lin.codfac = fac.codfac
AND ((fac.codcli = cli.codcli AND cli.codpostal like '12%')
OR (fac.codcli IS NULL)) AND to_char (fecha,'YY') = '95';
```

- 2.- ¿Qué errores produce la siguiente consulta?. Explica, sin ambigüedad, porqué se produce cada uno. (0.5 puntos)

```
SELECT nombre, descrip, SUM(cant)
FROM clientes cli, facturas fac, lineas_fac lin, articulos art
WHERE cli.codpue = pue.codpue AND cli.codcli = fac.codcli
AND fac.codfac = lin.codfac AND lin.codart = art.codart
AND EXISTS ( SELECT *
FROM pueblos pue, provincias pro
WHERE pue.codpro = pro.codpro
AND UPPER(pro.nombre) = 'CASTELLON')
GROUP BY codcli, codart
HAVING UPPER(nombre) like '%GARCIA%'
AND UPPER(descrip) like 'BOMBILLAS%';
```

**CUESTIONES SQL****(1 punto)**

- 1.- La información almacenada en la base de datos para los clientes no es completa, pudiendo existir clientes de los cuales no se tiene constancia de la localidad en la que viven. Según este comentario, ¿la siguiente sentencia SQL responde a la consulta “facturación del primer trimestre del año 1996 de los clientes de la provincia de Castellón?”. Razona la respuesta sin ambigüedad. (0.5 puntos)

```
SELECT SUM(precio * cant)
FROM facturas fac, lineas_fac lin, clientes cli, pueblos pue, provincias prv
WHERE prv.nombre = 'CASTELLON' AND prv.codpro = pue.codpro
      AND pue.codpue = cli.codpue AND cli.codcli = fac.codcli
      AND fac.codfac = lin.codfac
      AND TO_NUMBER(TO_CHAR (fecha,'YY')) = 96
      AND TO_NUMBER(TO_CHAR (fecha,'Q')) = 1;
```

- 2.- ¿Qué error podría aparecer al ejecutar la siguiente consulta?. Explica, sin ambigüedad, porqué se produce. (0.5 puntos)

```
SELECT art.descrip
FROM articulos art
WHERE 0 < ( SELECT fac.dto
            FROM facturas fac, lineas_fac lin
            WHERE fac.codfac = lin.codfac
            AND lin.codart = art.codart
            AND MONTHS_BETWEEN (SYSDATE, fecha) < 24 );
```

**CUESTIONES SQL****(1 punto)**

- 1.- ¿La siguiente sentencia SQL responde a la consulta “artículos que han aparecido más de una vez en una factura”? Razona la respuesta sin ambigüedad. (0.5 puntos)

```
SELECT DISTINCT art.codart, art.descrip
FROM  articulos art, lineas_fac lin, facturas fac
WHERE art.codart = lin.codart AND lin.codfac = fac.codfac
GROUP BY art.codart, fac.codfac
HAVING SUM(lin.cant * lin.precio) <> MAX (lin.cant * lin.precio)
ORDER BY 2 ;
```

- 2.- ¿Qué errores produce la siguiente consulta?. Explica, sin ambigüedad, porqué se produce cada uno. (0.5 puntos)

```
SELECT nombre
FROM  pueblos pue, articulos art
WHERE EXISTS
      ( SELECT *
        FROM facturas fac, lineas_fac lin, clientes cli
        WHERE fac.codfac = lin.codfac AND lin.codart = art.codart
              AND cli.codcli = fac.codcli AND cli.codpue = pue.codpue )
GROUP BY codpue, codart
HAVING SUM(cant) > 10 AND UPPER(descrip) LIKE 'CLAVO%' ;
```

**CUESTIONES SQL****(0.5 puntos)**

- 1.- La siguiente sentencia SQL intenta obtener "los clientes que han comprado más de 10 artículos diferentes en \_\_\_\_\_". Indica el periodo de tiempo relacionado con la consulta, así como si la consulta responde realmente al enunciado. (0.5 puntos)

```
SELECT cli.nombre
FROM clientes cli, facturas fac, lineas_fac lin
WHERE cli.codcli = fac.codcli
      AND fac.codfac = lin.codfac
      AND to_char(fac.fecha,'YYQ') = '964'
GROUP BY cli.codcli, cli.nombre
HAVING COUNT(*) > 10
ORDER BY 1;
```

**CUESTIONES SQL****(0.5 puntos)**

- 1.- La siguiente sentencia SQL intenta obtener "los provincias, ordenadas alfabeticamente, que tienen más de 100 clientes y más de 10 pueblos con clientes". Indica si aparece algún error en la consulta, y si ésta responde realmente al enunciado. Razona la respuesta. (0.5 puntos)

```
SELECT  prv.nombre
FROM    provincias prv, pueblos pue, clientes cli
WHERE   prv.codpro = pue.codpro
        AND   pue.codpue = cli.codpue
GROUP BY prv.codpro
HAVING  COUNT(codpue) > 10 AND COUNT(codcli) > 100
ORDER BY 1;
```

**CUESTIONES SQL****(0.5 puntos)**

- 1.- La siguiente sentencia SQL intenta obtener "el nombre y el número de facturas con un iva nulo que tiene cada uno de los clientes de Castellon que han hecho alguna compra". Indica si aparece algún error en la consulta, y si ésta responde realmente al enunciado. Razona la respuesta. (0.5 puntos)

```
SELECT cli.nombre, COUNT(*) - COUNT(fac.iva)
FROM provincias prv, pueblos pue, clientes cli, facturas fac
WHERE prv.codpro = pue.codpro
      AND pue.codpue = cli.codpue
      AND cli.codcli = fac.codcli
GROUP BY prv.codpro, cli.codcli
HAVING UPPER(prv.nombre) = 'CASTELLON'
ORDER BY 1;
```

**CUESTIONES SQL****(0.5 puntos)**

- 1.- La siguiente sentencia SQL intenta obtener, ordenadamente, "los clientes cuyos abonos durante el año pasado han sumado mas de 10 unidades, mostrando el número de unidades que han devuelto". Indica si aparece algún error en la consulta, y si ésta responde realmente al enunciado. Razona la respuesta. (0.5 puntos)

```
SELECT cli.nombre, -SUM (lin.cant)
FROM clientes cli, facturas fac, lineas_fac lin
WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
AND lin.cant < 0
AND TO_NUMBER(TO_CHAR (fac.fecha,'YY')) =
      TO_NUMBER(TO_CHAR (SYSDATE,'YY')) - 1
AND SUM (lin.cant) < -10
GROUP BY cli.codcli
ORDER BY 1;
```

**CUESTIONES SQL****(0.5 puntos)**

- 1.- La siguiente sentencia SQL intenta obtener, ordenadamente, "los clientes que durante el año pasado realizaron más de 25 facturas en las que se incluían artículos cuya descripción contenía el texto BASE. Razona la respuesta. (0.5 puntos)

```
SELECT cli.codcli, cli.nombre, COUNT(fac.codfac)
FROM clientes cli, facturas fac, lineas_fac lin, articulos art
WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
      AND lin.codart = art.codart AND UPPER(art.descrip) LIKE '%BASE%'
      AND TO_NUMBER(TO_CHAR(fac.fecha,'YY')) =
          TO_NUMBER(TO_CHAR(SYSDATE,'YY')) - 1
GROUP BY cli.codcli, fac.codfac
HAVING COUNT(fac.codfac) > 25
ORDER BY 2;
```



**CUESTIONES SQL****(0.5 puntos)**

- 1.- Verifica si la siguiente sentencia SQL obtiene, ordenadamente, "los artículos que durante el año pasado han tenido una venta mayor de 50 unidades". Razona la respuesta.  
(0.5 puntos)

```
SELECT  art.codart, art.descrip
FROM    facturas fac, lineas_fac lin, articulos art
WHERE   fac.codfac = lin.codfac AND lin.codart = art.codart
GROUP BY art.codart
HAVING  SUM(DISTINCT lin.cant) > 50
        AND  TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
              TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-1
ORDER BY 1, 2;
```

**CUESTIONES SQL****(0.5 puntos)**

- 1.- Verifica si la siguiente sentencia SQL obtiene, ordenadamente, "los artículos que sólo se han comprado en una única provincia, que también debe visualizarse". Razona la respuesta. (0.5 puntos)

```
SELECT  art.descrip, prv.nombre PROVINCIA
FROM    articulos art, lineas_fac lin, facturas fac, clientes cli, pueblos pue,
        provincias prv
WHERE   art.codart = lin.codart AND lin.codfac = fac.codfac
        AND fac.codcli = cli.codcli AND cli.codpue = pue.codpue
        AND pue.codpro = prv.codpro
GROUP BY art.codart, art.descrip
HAVING  COUNT(prv.codpro) = 1
ORDER BY 1;
```

**CUESTIONES SQL****(0.5 puntos)**

- 1.- Verifica si la siguiente sentencia SQL obtiene, "el porcentaje del total de los pueblos donde se han realizado ventas durante el año en curso.". Razona la respuesta. (0.5 puntos)

```
SELECT  COUNT(DISTINCT cli.codpue) / COUNT(DISTINCT pue.codpue)
FROM    pueblos pue, clientes cli, facturas fac
WHERE   cli.codcli = fac.codcli
AND     TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
        TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'));
```

**CUESTIÓN SQL****(0.5 puntos)**

- 1.- Verifica si la siguiente sentencia SQL obtiene, "el número de pueblos que durante el año pasado tuvieron clientes que realizaron compras.". Razona la respuesta. (0.5 puntos)

```
SELECT COUNT (codpue)
FROM clientes cli
WHERE EXISTS
( SELECT fac.codcli
  FROM facturas fac
  WHERE fac.codcli = cli.codcli
    AND TO_NUMBER(TO_CHAR(fac.fecha,'YY')) =
      TO_NUMBER(TO_CHAR(SYSDATE,'YY')) - 1 )
```

**CUESTIÓN SQL****(0.5 puntos)**

- 1.- Verifica si la siguiente sentencia SQL obtiene, "los artículos que durante el año pasado siempre se vendieron con un descuento superior el 15%.". Razona la respuesta, indicando los errores que puedan aparecer y su solución. (0.5 puntos)

```
SELECT  art.descrip
FROM    articulos art, lineas_fac lin, facturas fac
WHERE   art.codart = lin.codart AND lin.codfac = fac.codfac
        AND    TO_NUMBER(TO_CHAR (fac.fecha,'YY')) =
                TO_NUMBER(TO_CHAR (SYSDATE,'YY')) - 1
GROUP BY art.codart
HAVING  MIN (lin.dto) > 15
ORDER BY 1;
```

**CUESTIÓN SQL****(0.5 puntos)**

- 1.- Verifica si la siguiente sentencia SQL obtiene, "los clientes a los que se les ha aplicado más de un descuento diferente en las facturas del año pasado". Razona la respuesta, indicando los errores que puedan aparecer y su solución. (0.5 puntos)

```
SELECT cli.nombre
FROM clientes cli, facturas fac
WHERE cli.codcli = fac.codcli
AND TO_NUMBER(TO_CHAR (fac.fecha,'YY')) =
TO_NUMBER(TO_CHAR (SYSDATE,'YY')) - 1
GROUP BY cli.codcli
HAVING COUNT (fac.dto) > 1
ORDER BY 1;
```