

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar, ordenadamente, los clientes que cumplen que todas sus facturas tienen un descuento que sea mayor que la media del descuento de sus líneas. (0.75 puntos)

```
SELECT cli.nombre
FROM clientes cli
WHERE NOT EXISTS
( SELECT *
  FROM facturas fac, lineas_fac lin
  WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
  GROUP BY fac.codfac, fac.dto
  HAVING fac.dto <= AVG(NVL(lin.dto,0)) )
AND EXISTS
( SELECT *
  FROM facturas fac
  WHERE cli.codcli = fac.codcli )
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los clientes junto con los artículos que solicitaron durante el primer semestre del año 95 y no solicitaron esos mismos artículos en el segundo semestre del mismo año. (0.75 puntos)

```
SELECT cli.nombre, art.descrip
FROM clientes cli, articulos art
WHERE EXISTS
( SELECT *
  FROM facturas fac, lineas_fac lin
  WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
    AND art.codart = lin.codart AND to_char(fecha,'YY') = '95'
    AND to_number(to_char(fecha,'MM')) BETWEEN 1 AND 6 )
AND NOT EXISTS
( SELECT *
  FROM facturas fac, lineas_fac lin
  WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
    AND art.codart = lin.codart AND to_char(fecha,'YY') = '95'
    AND to_number(to_char(fecha,'MM')) BETWEEN 7 AND 12 )
ORDER BY 1, 2;
```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT cli.nombre
FROM clientes cli, facturas fac
WHERE cli.codcli = fac.codcli
AND EXISTS
( SELECT *
  FROM lineas_fac lin
  WHERE fac.codfac = lin.codfac
  GROUP BY lin.codfac
  HAVING ( SUM(lin.precio * lin.cant * NVL(lin.dto,0) / 100) *
          (1 + NVL(fac.dto,0)) * (1 + NVL(fac.iva,0))) > 10000 )
GROUP BY cli.codcli, cli.nombre
HAVING COUNT(*) > 3
ORDER BY 1;
```

Mostrar, ordenamente, los clientes que tienen más de tres facturas en las que se han ahorrado más de 10000 pesetas explotando las ofertas de los artículos.

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar el coste total de los artículos que tenemos en stock cuyo precio por unidad sea mayor de 2000 pesetas, de los que se ha solicitado en el último mes menos cantidad que su stock mínimo. (0.75 puntos)

```
SELECT SUM (precio * NVL(stock,0))
FROM  articulos art
WHERE (precio > 2000) AND
      ( ( NVL(stockmin,0) >
        ( SELECT SUM(cant)
          FROM  facturas fac, lineas_fac lin
          WHERE  art.codart = lin.codart AND fac.codfac = lin.codfac
              AND  MONTHS_BETWEEN (SYSDATE,fecha) <= 1 ) )
      OR
      NOT EXISTS
      ( SELECT *
        FROM  facturas fac, lineas_fac lin
        WHERE  art.codart = lin.codart AND fac.codfac = lin.codfac
              AND  MONTHS_BETWEEN (SYSDATE,fecha) <= 1 ) )
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los clientes que han comprado un mismo artículo en más de una factura. (0.75 puntos)

```
SELECT cli.nombre
FROM  clientes cli
WHERE EXISTS
      ( SELECT *
        FROM  facturas fac, lineas_fac lin
        WHERE  cli.codcli = fac.codcli AND fac.codfac = lin.codfac
        GROUP BY  lin.codart
        HAVING COUNT(DISTINCT fac.codfac) > 1 )
ORDER BY 1;
```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT  art.descrip
FROM    articulos art
WHERE   NOT EXISTS
      ( SELECT  *
        FROM    clientes cli, pueblos pue, provincias prv
        WHERE   prv.nombre = 'CASTELLON'
              AND prv.codpro = pue.codpro
              AND pue.codpue = cli.codpue
              AND NOT EXISTS
                    ( SELECT  *
                      FROM    facturas fac, lineas_fac lin
                      WHERE   fac.codfac = lin.codfac
                            AND fac.codcli = cli.codcli
                            AND lin.codart = art.codart ) );

ORDER BY 1;
```

Mostrar, ordenamente, los artículos que han sido vendidos a todos los clientes de Castellón.

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar, ordenadamente, los clientes que han comprado en el año 1996, todos los articulos que tienen un stock menor que el stock mínimo. (0.75 puntos)

```
SELECT cli.nombre
FROM clientes cli,
WHERE NOT EXISTS
( SELECT *
  FROM articulos
  WHERE NVL(art.stock,0) < NVL(art.stock_min,0)
    AND NOT EXISTS
      ( SELECT *
        FROM facturas fac, lineas_fac lin
        WHERE lin.codart = art.codart
          AND fac.codfac = lin.codfac
          AND fac.codcli = cli.codcli
          AND TO_CHAR(fac.fecha,'YY') = '96' ) )
ORDER BY 1 ;
```

- 2.- Mostrar, ordenadamente, los clientes que han solicitado más de dos clases de tornillos durante el mes de Diciembre de 1996.

```
SELECT cli.nombre
FROM clientes cli
WHERE cli.codcli IN
( SELECT fac.codcli
  FROM facturas fac, lineas_fac lin, articulos art
  WHERE fac.codfac = lin.codfac
    AND lin.codart = art.codart
    AND UPPER(art.descrip) LIKE 'TORNILLO%'
  GROUP BY fac.codcli
  HAVING COUNT (DISTINCT lin.codart) > 2 ) ;
```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT cli.nombre, art.descrip
FROM clientes cli, articulos art
WHERE cli.codcli IN
    ( SELECT fac.codcli
      FROM facturas fac, lineas_fac lin
      WHERE fac.codfac = lin.codfac
      GROUP BY fac.codcli
      HAVING SUM(lin.cant) >= ALL
        ( SELECT SUM(lin.cant)
          FROM facturas fac, lineas_fac lin
          WHERE fac.codfac = lin.codfac
          GROUP BY fac.codcli ) )
AND art.codart IN
    ( SELECT lin.codart
      FROM facturas fac, lineas_fac lin
      WHERE fac.codfac = lin.codfac
      AND fac.codcli = cli.codcli
      GROUP BY lin.codart
      HAVING SUM(lin.cant) >= ALL
        ( SELECT SUM(lin.cant)
          FROM facturas fac, lineas_fac lin
          WHERE fac.codfac = lin.codfac
          AND fac.codcli = cli.codcli
          GROUP BY lin.codart ) )
ORDER BY 1, 2 ;
```

Muestra, ordenadamente, los clientes que han pedido más artículos junto con los artículos que más pidieron.

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar, ordenadamente, los nombres de las provincias que tienen el mayor número de pueblos. (0.75 puntos)

```
SELECT  prv.nombre
FROM    provincias prv, pueblos pue
WHERE   prv.codpro = pue.codpro
GROUP BY  prv.codpro, prv.nombre
HAVING  COUNT(*) >= ALL
        ( SELECT  COUNT(*)
          FROM    provincias prv, pueblos pue
          WHERE   prv.codpro = pue.codpro
          GROUP BY  prv.codpro )
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los artículos que siempre aparecen solos en una factura, es decir, no aparece ningún otro artículo en ninguna de sus líneas de las facturas en las que aparece dicho artículo. (0.75 puntos)

```
SELECT  art.descrip
FROM    articulos art
WHERE   NOT EXISTS
        ( SELECT *
          FROM  lineas_fac lin1
          WHERE art.codart = lin1.codart
              AND lin1.codart <> ANY
                ( SELECT  lin.codart
                  FROM    lineas_fac lin2
                  WHERE   lin1.codfac = lin2.codfac ) )
AND     EXISTS
        ( SELECT *
          FROM  lineas_fac lin1
          WHERE art.codart = lin1.codart )
ORDER BY 1;
```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT  art1.descrip, art2.descrip
FROM    articulos art1, articulos art2
WHERE   art1.codart < art2.codart
AND     NOT EXISTS
( SELECT lin.codfac
  FROM  lineas_fac lin
  WHERE lin.codart = art1.codart
MINUS
  SELECT lin.codfac
  FROM  lineas_fac lin
  WHERE lin.codart = art2.codart )
AND     NOT EXISTS
( SELECT lin.codfac
  FROM  lineas_fac lin
  WHERE lin.codart = art2.codart
MINUS
  SELECT lin.codfac
  FROM  lineas_fac lin
  WHERE lin.codart = art1.codart )
AND     art1.codart IN
( SELECT lin.codart FROM lineas_fac lin)
ORDER BY 1, 2;
```

Mostrar, ordenadamente, los pares de artículos que siempre son vendidos juntos, así como las pares de artículos que nunca se han vendido.

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar, ordenadamente por provincias, el nombre de la provincia y el número de clientes que durante el año pasado han facturado más de 100000 ptas, sin tener en cuenta el descuento y el iva. (0.75 puntos)

```
SELECT  prv.nombre, COUNT(*)
FROM    provincias prv, pueblos pue, clientes cli
WHERE   prv.codpue = pue.codpue AND pue.codcli = cli.codcli
AND     cli.codcli IN
        ( SELECT  fac.codcli
          FROM     facturas fac, lineas_fac lin
          WHERE    fac.codfac = lin.codfac
                AND TO_NUMBER(TO_CHAR(fac.codfac,'YY')) =
                  TO_NUMBER(TO_CHAR(SYSDATE,'YY')) - 1
          GROUP BY fac.codcli
          HAVING   SUM(lin.precio * lin.cant) > 100000 )
GROUP BY prv.codpro, prv.nombre
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los clientes que nunca han realizado más de una compra en una fecha determinada. (0.75 puntos)

```
SELECT  cli.nombre
FROM    clientes cli
WHERE   NOT EXISTS
        ( SELECT  *
          FROM     facturas fac1, facturas fac2
          WHERE    fac1.codcli = cli.codcli
                AND fac1.codcli = fac2.codcli
                AND fac1.codfac <> fac2.codfac
                AND fac1.fecha = fac2.fecha )
AND     cli.codcli IN (SELECT codcli FROM facturas)
ORDER BY 1
```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT cli.nombre
FROM clientes cli, facturas fac
WHERE cli.codcli = fac.codcli
AND TO_CHAR(fac.fecha,'YYQ') = '964'
AND cli.codcli NOT IN
( SELECT cli.codcli
  FROM clientes cli, facturas fac
  WHERE cli.codcli = fac.codcli
    AND TO_CHAR(fac.fecha,'YYQ') = '964'
MINUS
SELECT cli.codcli
  FROM clientes cli, facturas fac
  WHERE cli.codcli = fac.codcli
    AND TO_CHAR(fac.fecha,'YYQ') = '963' )
ORDER BY 1;
```

Mostrar, ordenadamente, los clientes que han realizado alguna compra en los dos últimos trimestres del año 1996.

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar, ordenadamente, los clientes que han realizado la devolución de un artículo más de una semana después de la fecha de compra. Se asume que la fecha de compra coincide con la de la factura más proxima en la que aparece el artículo vendido. (0.75 puntos)

```
SELECT DISTINCT cli.nombre
FROM clientes cli, facturas fac1, lineas_fac lin1
WHERE cli.codcli = fac1.codcli
AND fac1.codfac = lin1.codart
AND lin1.cant < 0
AND 7 < ( SELECT fac1.fecha - MAX(fac2.fecha)
          FROM facturas fac2, lineas_fac lin2
          WHERE cli.codcli = fac2.codcli
            AND fac2.codfac = lin2.codfac
            AND fac2.fecha < fac1.fecha
            AND lin2.codart = lin1.codart
            AND lin2.cant > 0 )

ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los artículos que durante el año pasado sufrieron todos los tipos de IVA, {0 ó NULL, 7, 16}, y que cumple que el precio actual es menor que todos los aplicados durante el año pasado. (0.75 puntos)

```
SELECT DISTINCT art.descrip
FROM articulos art
WHERE art.precio < ALL
      ( SELECT lin.precio
        FROM lineas_fac lin, facturas fac
        WHERE lin.codfac = fac.codfac
          AND TO_NUMBER(TO_CHAR(fac2.fecha,'YY')) =
              TO_NUMBER(TO_CHAR(SYSDATE,'YY')) - 1 )

AND 3 =
      ( SELECT COUNT(DISTINCT NVL(fac.iva,0))
        FROM lineas_fac lin, facturas fac
        WHERE lin.codfac = fac.codfac
          AND TO_NUMBER(TO_CHAR(fac2.fecha,'YY')) =
              TO_NUMBER(TO_CHAR(SYSDATE,'YY')) - 1 )

ORDER BY 1
```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT  art.descrip
FROM    articulos art, lineas_fac lin1, facturas fac1, clientes cli1, pueblos pue1,
        provincias prv1
WHERE   art.codart = lin1.codart AND lin1.codfac = fac1.codfac
        AND fac1.codcli = cli1.codcli AND cli1.codpue = pue1.codpue
        AND pue1.codpro = prv1.codpro
        AND UPPER (prv1.nombre) = 'CASTELLON'
        AND MONTHS_BETWEEN (fac1.fecha, SYSDATE) < 1
GROUP BY art.codart, art.descrip
HAVING  SUM(lin1.cant) < ANY
        ( SELECT  SUM(lin2.cant)
          FROM    lineas_fac lin2, facturas fac2, clientes cli2, pueblos pue2,
                provincias prv2
          WHERE   art.codart = lin2.codart AND lin2.codfac = fac2.codfac
                AND fac2.codcli = cli2.codcli AND cli2.codpue = pue2.codpue
                AND pue2.codpro = prv2.codpro
                AND UPPER (prv2.nombre) <> 'CASTELLON'
                AND MONTHS_BETWEEN (fac2.fecha, SYSDATE) < 1
          GROUP BY prv2.codpro )
ORDER BY 1
```

Mostrar, ordenadamente, los artículos que la cantidad total vendida en Castellón en el último mes es menor que la cantidad total vendida en cualquier otra provincia en la que hubo venta durante el mismo periodo.

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar, ordenadamente, los clientes que durante el año pasado han comprado todos los artículos cuya descripción incluye el texto BASE. (0.75 puntos)

```
SELECT DISTINCT cli.nombre
FROM clientes cli, facturas fac
WHERE cli.codcli = fac.codcli AND NOT EXISTS
    ( SELECT *
      FROM articulos art
      WHERE UPPER(art.descrip) LIKE '%BASE%'
        AND NOT EXISTS
            ( SELECT *
              FROM facturas fac, lineas_fac lin
              WHERE cli.codcli = fac.codcli
                AND fac.codfac = lin.codfac
                AND lin.codart = art.codart
                AND TO_NUMBER(TO_CHAR (fac.fecha,'YY')) =
                  TO_NUMBER(TO_CHAR (SYSDATE,'YY'))-1 )
        )
    )
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los meses del año pasado en los que se realizó una venta menor que la media mensual de dicho año. (No tener en cuenta los descuentos ni el IVA) (0.75 puntos)

```
SELECT TO_CHAR(fac.fecha,'MONTH')
FROM facturas fac, lineas_fac lin
WHERE fac.codfac = lin.codfac
  AND TO_NUMBER(TO_CHAR (fac.fecha,'YY')) =
    TO_NUMBER(TO_CHAR (SYSDATE,'YY')) - 1
GROUP BY TO_CHAR(fac.fecha,'MONTH')
HAVING SUM (lin.precio * lin.cant) <
    ( SELECT SUM (lin.precio * in.cant) /
      COUNT (DISTINCT TO_CHAR(fac.fecha,'MONTH'))
      FROM facturas fac, lineas_fac lin
      WHERE fac.codfac = lin.codfac
        AND TO_NUMBER(TO_CHAR (fac.fecha,'YY')) =
          TO_NUMBER(TO_CHAR (SYSDATE,'YY')) - 1 )
ORDER BY 1;
```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT  pro.nombre, COUNT (*)
FROM    provincias pro, pueblos pue, clientes cli, facturas fac
WHERE   pro.codpro= pue.codpro AND pue.codpue = cli.codpue
        AND cli.codcli = fac.codcli
        AND UPPER (pro.nombre) IN
            ('CASTELLON','VALENCIA','ALICANTE')
        AND 10000 <
            ( SELECT  SUM (lin.precio * lin.cant)
              FROM    lineas_fac lin
              WHERE   fac.codfac = lin.codfac )
GROUP BY pro.codpro, pro.nombre
UNION   SELECT  pro.nombre, 0
        FROM    provincias pro
        WHERE   UPPER (pro.nombre) IN
            ('CASTELLON','VALENCIA','ALICANTE')
        AND    NOT EXISTS
            ( SELECT *
              FROM  pueblos pue, clientes cli, facturas fac
              WHERE pro.codpro= pue.codpro
                AND  pue.codpue = cli.codpue
                AND  cli.codcli = fac.codcli
                AND  10000 <
                    ( SELECT  SUM (lin.precio * lin.cant)
                      FROM    lineas_fac lin
                      WHERE   fac.codfac = lin.codfac ) )
GROUP BY pro.codpro, pro.nombre
ORDER BY 1;
```

Mostrar, ordenadamente, las provincias de la comunidad valenciana mostrando el número de facturas con un importe menor de 10000. (No se tienen en cuenta los descuentos ni el IVA).

EJERCICIO 3

(2.25 puntos)

- 1.- Mostrar las provincias de la comunidad valenciana ordenadas respecto del número de pueblos de cada provincia con un nombre compuesto, es decir con un carácter blanco en su nombre. El resultado debe mostrar una fila para cada provincia. (Hay que considerar que una provincia puede no tener ningún pueblo con nombre compuesto) (0.75 puntos)

```

SELECT  pro.nombre, COUNT (*)
FROM    provincias pro, pueblos pue
WHERE   pro.codpro= pue.codpro
        AND    UPPER (pro.nombre) IN
                ('CASTELLON','VALENCIA','ALICANTE')
        AND    INSTR (pue.nombre,'') > 0
GROUP BY  pro.codpro, pro.nombre
UNION   SELECT  pro.nombre, 0
FROM    provincias pro, pueblos pue
WHERE   UPPER (pro.nombre) IN
        ('CASTELLON','VALENCIA','ALICANTE')
        AND    NOT EXISTS
                ( SELECT  *
                  FROM    pueblos pue
                  WHERE   pro.codpro= pue.codpro
                        AND    INSTR (pue.nombre,'') > 0 )

ORDER BY 1;

```

- 2.- Mostrar, ordenadamente, los clientes que han realizado compras durante este año que superan el 80% de las compras realizadas por el resto de clientes durante este mes. (No tener en cuenta los descuentos ni el IVA) (0.75 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,'YY') = TO_CHAR (SYSDATE,'YY')
GROUP BY cli.codcli, cli.nombre
HAVING SUM (lin.precio * lin.cant) >
( SELECT SUM (lin.precio * lin.cant) * 0.8
FROM facturas fac, lineas_fac lin
WHERE fac.codcli <> cli.codcli AND fac.codfac = lin.codfac
AND TO_CHAR (fac.fecha,'MMYY') =
TO_CHAR (SYSDATE,'MMYY') )
ORDER BY 1;

```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT  fac.codfac, cli.nombre, fac.fecha, fac.iva,
        SUM(lin.precio * lin.cant), SUM (lin.cant)
FROM    clientes cli, facturas fac, lineas_fac lin
WHERE   cli.codcli = fac.codcli AND fac.codfac = lin.codfac
        AND   TO_CHAR (fac.fecha,'YY') = TO_CHAR (SYSDATE,'YY')
GROUP BY fac.codfac, cli.codcli, cli.nombre, fac.fecha, fac.iva
HAVING SUM (lin.precio * lin.cant) =
        ( SELECT  MAX(SUM (lin.precio * lin.cant))
          FROM    facturas fac, lineas_fac lin
          WHERE   fac.codfac = lin.codfac
                AND   TO_CHAR (fac.fecha,'YY') =
                      TO_CHAR (SYSDATE,'YY') )
UNION   SELECT  fac.codfac, cli.nombre, fac.fecha, fac.iva,
        SUM(lin.precio * lin.cant), SUM (lin.cant)
FROM    clientes cli, facturas fac, lineas_fac lin
WHERE   cli.codcli = fac.codcli AND fac.codfac = lin.codfac
        AND   TO_CHAR (fac.fecha,'YY') = TO_CHAR (SYSDATE,'YY')
GROUP BY fac.codfac, cli.codcli, cli.nombre, fac.fecha, fac.iva
HAVING SUM (lin.cant) =
        ( SELECT  MAX(SUM (lin.cant))
          FROM    facturas fac, lineas_fac lin
          WHERE   fac.codfac = lin.codfac
                AND   TO_CHAR (fac.fecha,'YY') =
                      TO_CHAR (SYSDATE,'YY') )
ORDER BY 1;
```

Mostrar, ordenadamente, los datos de las facturas que durante este año tengan un importe más alto y de las facturas de este año en las que se hayan vendido más artículos. (Sin tener en cuenta los descuentos ni el IVA)

EJERCICIO 3**(2.25 puntos)**

- 1.- Mostrar, ordenadamente, los pueblos donde la suma de las compras de sus clientes en el último año superó en un 5% las compras de hace dos años. (No tener en cuenta los descuentos ni el IVA) (0.75 puntos)

```

SELECT pue.nombre
FROM pueblos pue, clientes cli, facturas fac, lineas_fac lin
WHERE pue.codpue = cli.codpue AND cli.codcli = fac.codcli
      AND fac.codfac = lin.codfac
      AND TO_NUMBER(TO_CHAR (fac.fecha,'YYYY')) =
          TO_NUMBER(TO_CHAR (SYSDATE,'YYYY')) - 1
GROUP BY pue.codpue, pue.nombre
HAVING SUM (lin.precio * lin.cant) >
    ( SELECT NVL(SUM (lin.precio * lin.cant),0) * 1.05
      FROM clientes cli, facturas fac, lineas_fac lin
      WHERE pue.codpue = cli.codpue AND cli.codcli = fac.codcli
            AND fac.codfac = lin.codfac
            AND TO_NUMBER(TO_CHAR (fac.fecha,'YYYY')) =
                TO_NUMBER(TO_CHAR (SYSDATE,'YYYY')) - 2 )
ORDER BY 1;

```

- 2.- Mostrar, ordenadamente, los artículos que cumplen que nunca se les aplica descuento cuando se venden a los clientes que no son de CASTELLÓN y el descuento a los clientes de CASTELLÓN siempre es mayor que 20. (0.75 puntos)

```

SELECT art.descrip
FROM articulos art
WHERE NOT EXISTS
    ( SELECT *
      FROM clientes cli, pueblos pus, provincias prv
      WHERE cli.codpue = pue.codpue AND pue.codpro = prv.codpro
            AND upper(prv.nombre) <> 'CASTELLON'
            AND NOT EXISTS
                (SELECT *
                 FROM facturas fac, lineas_fac lin
                 WHERE fac.codcli = cli.codcli
                       AND fac.codfac = lin.codfac
                       AND lin.codart = art.codart
                       AND NVL(lin.dto,0) > 0 ) )
AND NOT EXISTS
    ( SELECT *
      FROM clientes cli, pueblos pus, provincias prv
      WHERE cli.codpue = pue.codpue AND pue.codpro = prv.codpro
            AND upper(prv.nombre) = 'CASTELLON'
            AND NOT EXISTS
                (SELECT *
                 FROM facturas fac, lineas_fac lin
                 WHERE fac.codcli = cli.codcli
                       AND fac.codfac = lin.codfac
                       AND lin.codart = art.codart
                       AND NVL(lin.dto,0) < 20 ) )
ORDER BY 1;

```

EJERCICIO 3 (cont)**(2.25 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.75 puntos)

```
SELECT  art.descrip
FROM    articulos art
WHERE   art.precio <
        ( SELECT  NVL(AVG(lin.precio),0)
          FROM    lineas_fac lin, facturas fac
          WHERE   lin.codart = art.codart
                AND fac.codfac = lin.codfac
                AND TO_CHAR(fac.fecha,'MMYYYY') = '121998' )
AND EXISTS
        ( SELECT  *
          FROM    lineas_fac lin, facturas fac
          WHERE   lin.codart = art.codart
                AND fac.codfac = lin.codfac
                AND TO_CHAR(fac.fecha,'MMYYYY') = '121998' )
AND     art.precio >
        ( SELECT  NVL(AVG(lin.precio),0)
          FROM    lineas_fac lin, facturas fac
          WHERE   lin.codart = art.codart
                AND fac.codfac = lin.codfac
                AND TO_CHAR(fac.fecha,'MMYYYY') = '111998' )
AND EXISTS
        ( SELECT  *
          FROM    lineas_fac lin, facturas fac
          WHERE   lin.codart = art.codart
                AND fac.codfac = lin.codfac
                AND TO_CHAR(fac.fecha,'MMYYYY') = '111998' )
ORDER BY 1 ;
```

Mostrar, ordenadamente, los artículos cuyo precio actual sea menor que la media de los precios de sus ventas en Diciembre de 1998, y mayor que la media de los precios de sus ventas en Noviembre de 1998.

EJERCICIO 3**(2 puntos)**

- 1.- Listar, ordenadamente, la descripción de todos los artículos junto con el número de días del año pasado en el que se realizaron ventas de dicho artículo. (0.75 puntos)

```
SELECT art.descrip, COUNT(DISTINCT TO_CHAR(fac.fecha,'DDD')) DIAS
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,'YYYY')) =
            TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
GROUP BY  art.codart, art.descrip
UNION SELECT  art.descrip, 0 DIAS
      FROM  articulos art
      WHERE art.codart NOT IN
            (SELECT  lin.codart
             FROM  lineas_fac lin, facturas fac
             WHERE  lin.codfac = fac.codfac
                   AND  TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
                        TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
            )
ORDER BY  1;
```

- 2.- Mostrar el mes del año pasado en el que se realizaron más ventas (sin tener en cuenta descuentos e iva). (0.75 puntos)

```
SELECT  TO_CHAR(fac.fecha,'MONTH')
FROM  facturas fac, lineas_fac lin
WHERE  fac.codfac = lin.codfac
      AND  TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
            TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
GROUP BY  TO_CHAR(fac.fecha,'MONTH')
HAVING  SUM(lin.precio*lin.cant) =
        (SELECT  MAX(SUM(lin.precio*lin.cant))
         FROM  facturas fac, lineas_fac lin
         WHERE  fac.codfac = lin.codfac
               AND  TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
                    TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
        )
GROUP BY  TO_CHAR(fac.fecha,'MONTH')
);
```

EJERCICIO 3 (cont)**(2 puntos)**

- 3.- ¿A que consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(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,'YYYY') = TO_CHAR (SYSDATE,'YYYY')
GROUP BY cli.codcli, cli.nombre
HAVING SUM (lin.cant) >
( SELECT
SUM (DECODE (cli.codcli,fac.codcli,
DECODE (fac.codfac,lin.codfac,
DECODE (TO_CHAR (fac.fecha,'YYYY'),
TO_CHAR (SYSDATE,'YYYY'),lin.cant,0),0),0))
/
COUNT (DISTINCT
DECODE (cli.codcli,fac.codcli,
DECODE (fac.codfac,lin.codfac,
DECODE (TO_CHAR (fac.fecha,'YYYY'),
TO_CHAR (SYSDATE,'YYYY'),cli.codcli))))
FROM clientes cli, facturas fac, lineas_fac lin
)
ORDER BY 1;
```

Mostrar, ordenadamente, los clientes cuyas compras en unidades durante este año está por encima de la media de las compras en unidades de cada uno de los clientes en ese periodo.

EJERCICIO 3**(2 puntos)**

- 1.- Mostrar, ordenadamente, las provincias donde al menos 8 clientes han superado las 10.000 pesetas de compras, sin tener en cuenta descuentos ni iva, durante el último año. (0.75 puntos)

```
SELECT prv.nombre
FROM provincias prv, pueblos pue, clientes cli
WHERE prv.codpro = pue.codpro AND pue.codpue = cli.codpue
AND 10000 <=
    ( SELECT lin.precio * lin.cant
      FROM facturas fac, lineas_fac lin
      WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
        AND TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
            TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1 )
GROUP BY prv.codpro, prv.nombre
HAVING COUNT(*) >= 8
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los artículos que han sido vendidos a menos de 20 clientes. (0.75 puntos)

```
SELECT art.descrip
FROM articulos art
WHERE 20 <
    ( SELECT COUNT(*)
      FROM clientes cli, facturas fac, lineas_fac lin
      WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
        AND lin.codart = art.codart
      GROUP BY cli.codcli )
AND EXISTS
    ( SELECT *
      FROM clientes cli, facturas fac, lineas_fac lin
      WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
        AND lin.codart = art.codart )
ORDER BY 1;
```

EJERCICIO 3 (cont)**(2 puntos)**

- 3.- ¿A qué consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(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_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
GROUP BY cli.codcli, cli.nombre
HAVING SUM(lin.precio * lin.cant) * 0,5 > any
( SELECT SUM(lin.precio * lin.cant)
FROM facturas fac, lineas_fac lin
WHERE cli.codcli = fac.codcli AND fac.codfac = lin.codfac
AND TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
GROUP BY TO_CHAR(fac.fecha,'MM') )
ORDER BY 1;
```

Mostrar, ordenadamente, los clientes con más del 50% de sus compras del año pasado en un único mes, sin tener en cuenta descuentos e iva.

EJERCICIO 3**(2 puntos)**

- 1.- Mostrar, ordenadamente, los clientes que sólo han hecho compras en un único mes, en todos los años en los que aparece como cliente, debiéndose visualizar también el mes en cuestión y el rango de los años. (0.75 puntos)

```
SELECT cli.nombre, TO_CHAR(MAX(fac.fecha),'MONTH') MES,
      MIN(TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')))) ANYO_INI,
      MAX(TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')))) ANYO_FIN
FROM clientes cli, facturas fac
WHERE cli.codcli = fac.codcli
      AND TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
          TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
GROUP BY cli.codcli, cli.nombre
HAVING COUNT(DISTINCT TO_CHAR(fac.fecha,'MONTH')) = 1 AND
      (COUNT(DISTINCT TO_CHAR(fac.fecha,'YYYY')) =
          MAX(TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) -
              MIN(TO_NUMBER(TO_CHAR(fac.fecha,'YYYY'))))

ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los artículos en el último año que SÓLO han sido comprados en la provincia de Castellón. (0.75 puntos)

```
SELECT art.descrip
FROM articulos art
WHERE art.codart NOT IN
      (SELECT lin.codart
       FROM lineas_fac lin, facturas fac, clientes cli, pueblos pue,
            provincias prv
       WHERE lin.codfac = fac.codfac AND fac.codcli = cli.codcli
            AND cli.codpue = pue.codpue AND pue.codpro = prv.codpro
            AND UPPER(prv.nombre) <> 'CASTELLON'
            AND TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
                  TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1 )
AND art.codart IN
      (SELECT lin.codart
       FROM lineas_fac lin, facturas fac, clientes cli, pueblos pue,
            provincias prv
       WHERE lin.codfac = fac.codfac AND fac.codcli = cli.codcli
            AND cli.codpue = pue.codpue AND pue.codpro = prv.codpro
            AND UPPER(prv.nombre) = 'CASTELLON'
            AND TO_NUMBER(TO_CHAR(fac.fecha,'YYYY')) =
                  TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1)

ORDER BY 1;
```

EJERCICIO 3 (cont)**(2 puntos)**

- 3.- ¿A qué consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.5 puntos)

```
SELECT  art.descrip, MAX(ABS(lin.precio-art.precio)) VENTA_ACTUAL
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,'YYYY')) =
                TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
GROUP BY art.codart, art.descrip
HAVING  MAX(ABS(lin.precio-art.precio)) >=
        ( SELECT  MAX(MAX(ABS(lin.precio-art.precio)))
          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,'YYYY')) =
                        TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - 1
          GROUP BY lin.codart )
ORDER BY 1;
```

Mostrar, ordenadamente, los artículos que durante el último año presentan una mayor diferencia entre el precio el venta y el precio actual, que también debe visualizarse.

EJERCICIO 3**(2 puntos)**

- 1.- Mostrar, ordenadamente, los artículos que están por debajo de su stock mínimo, junto con la fecha de la factura en la que se produjo la venta que provocó que el stock estuviera por debajo del stock mínimo. Hay que tener en cuenta que dicha factura no tiene porque ser la última. (0.75 puntos)

```
SELECT DISTINCT art.descrip, MAX(fac.fecha)
FROM  articulos art, lineas_fac lin, facturas fac
WHERE  art.codart = lin.codart AND lin.codfac = fac.codfac
      AND      nvl(art.stock,0) < nvl(art.stock_min,0)
      AND      nvl(art.stock_min,0) - nvl(art.stock,0) <=
      ( SELECT SUM(lin2.cant)
        FROM  lineas_fac lin2, facturas fac2
        WHERE  art.codart = lin2.codart AND lin2.codfac = fac2.codfac
              AND      fac2.fecha >= fac.fecha
      )
GROUP BY art.codart, art.
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los pueblos, y la provincia a la que pertenece, en los que todos sus clientes han realizado compras superiores a 10000 pesetas, sin tener en cuenta descuentos e iva, durante el año pasado. (0.75 puntos)

```
SELECT pue.nombre, prv.nombre
FROM  provincias prv, pueblos pue, clientes cli
WHERE  prv.codpro = pue.codpro AND pue.codpue = cli.codpue
GROUP BY pue.codpue, pue.nombre, prv.nombre
HAVING COUNT(*) =
      ( SELECT COUNT(COUNT(*))
        FROM  clientes cli, facturas fac, lineas_fac lin
        WHERE  pue.codpue = cli.codpue AND cli.codcli = fac.codcli
              AND      fac.codfac = lin.codfac
              AND      TO_NUMBER(TO_CHAR (fac.fecha,'YYYY')) =
                      TO_NUMBER(TO_CHAR (SYSDATE,'YYYY')) - 1
        GROUP BY cli.codcli
        HAVING SUM(lin.cant * lin.precio) > 10000)
ORDER BY 1;
```

EJERCICIO 3 (cont)**(2 puntos)**

- 3.- ¿A qué consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.5 puntos)

```
SELECT TO_CHAR (fac.fecha,'MONTH')
FROM facturas fac, lineas_fac lin
WHERE fac.codfac = lin.codfac
AND TO_NUMBER(TO_CHAR (fac.fecha,'YYYY')) =
TO_NUMBER(TO_CHAR (SYSDATE,'YYYY')) - 1
GROUP BY TO_CHAR (fac.fecha,'MONTH')
HAVING 3 >
( SELECT COUNT(COUNT(*))
FROM facturas fac2, lineas_fac lin2
WHERE fac2.codfac = lin2.codfac
AND TO_NUMBER(TO_CHAR (fac2.fecha,'YYYY')) =
TO_NUMBER(TO_CHAR (SYSDATE,'YYYY')) - 1
GROUP BY TO_CHAR (fac2.fecha,'MONTH')
HAVING SUM(lin.precio * lin.cant) < SUM(lin2.precio * lin2.cant) )
ORDER BY 1;
```

Mostrar, ordenadamente, los tres meses del año pasado que tienen un mayor volumen de ventas (no tener en cuenta descuentos e iva).

EJERCICIO 3**(2 puntos)**

- 1.- Mostrar, ordenadamente, el nombre de las provincias que durante el año pasado tienen un volumen de ventas superior a la media de las ventas de las provincias que han tenido alguna venta dicho año (no tener en cuenta descuentos e iva). (0.75 puntos)

```
SELECT  prv.nombre
FROM    provincias prv, pueblos pue, clientes cli, facturas fac, lineas_fac lin
WHERE   prv.codpro = pue.codpro AND pue.codpue = cli.codpue
        AND cli.codcli = fac.codcli AND fac.codfac = lin.codfac
        AND TO_NUMBER(TO_CHAR (fac.fecha, 'YYYY')) =
            TO_NUMBER(TO_CHAR (SYSDATE, 'YYYY')) - 1
GROUP BY prv.codpro, prv.nombre
HAVING  SUM(lin.precio * lin.cant) >
( SELECT  SUM(lin.precio * lin.cant) / COUNT(DISTINCT pue.codpro)
  FROM    pueblos pue, clientes cli, facturas fac, lineas_fac lin
  WHERE   pue.codpue = cli.codpue AND cli.codcli = fac.codcli
        AND fac.codfac = lin.codfac
        AND TO_NUMBER(TO_CHAR (fac.fecha, 'YYYY')) =
            TO_NUMBER(TO_CHAR (SYSDATE, 'YYYY')) - 1 )
ORDER BY 1;
```

- 2.- Mostrar, ordenadamente, los clientes a los que durante el año pasado se les aplicaron todos los diferentes iva que se aplicaron durante ese año. (0.75 puntos)

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

EJERCICIO 3 (cont)**(2 puntos)**

- 3.- ¿A qué consulta corresponde la siguiente sentencia SQL?. Responder sin ambigüedad.
(0.5 puntos)

```
SELECT  art.descrip, TO_CHAR (fac.fecha,'YYYY')
FROM    facturas fac, articulos art, lineas_fac lin
WHERE   fac.codfac = lin.codfac AND lin.codart = art.codart
GROUP BY TO_CHAR (fac.fecha,'YYYY'), art.codart, art.descrip
HAVING
    TO_NUMBER (
        TO_CHAR (
            TO_DATE ('3112'|| TO_CHAR(fac.fecha,'YYYY'),
                    'DDMMYYYY'),'WW')) =
        COUNT (DISTINCT TO_CHAR(fac.fecha,'WW'))
ORDER BY 1;
```

Mostrar, ordenadamente, los artículos que se han vendido en todas las semanas de un año, que también se visualiza.