

1. Listar Número de Cliente, apellido y nombre, Total Comprado por el cliente 'Total del Cliente', Cantidad de Órdenes de Compra del cliente 'OCs del Cliente' y la Cant. de Órdenes de Compra de todos los clientes 'Cant. Total OC', de todos aquellos clientes cuyo promedio de compra por Orden supere al promedio de órdenes de compra general, tengan al menos 2 órdenes y cuyo zipcode comience con 94.

```
SELECT c.customer_num, fname, lname,
       SUM(unit_price*quantity) 'Total del Cliente',
       COUNT(DISTINCT i.order_num) 'OCs del Cliente',
       (SELECT COUNT(o2.order_num) FROM orders o2) 'Cant Total OC'
FROM customer c JOIN orders o ON (c.customer_num = o.customer_num)
                JOIN items i ON (o.order_num = i.order_num)
WHERE zipcode LIKE '94%'
GROUP BY c.customer_num, fname, lname
HAVING (SUM(unit_price*quantity)/COUNT(DISTINCT i.order_num))
       > (SELECT (SUM(unit_price*quantity)/COUNT(DISTINCT i3.order_num))
          FROM items i3) AND COUNT(DISTINCT i.order_num) >=2
```

2.a Se requiere crear una tabla temporal #ABC_Productos un ABC de Productos ordenado por cantidad de venta en u\$, los datos solicitados son:

Nro. de Stock, Código de fabricante, descripción del producto, Nombre de Fabricante, Total del producto pedido 'u\$ por Producto', Cant. de producto pedido 'Unid. por Producto', para los productos que pertenezcan a fabricantes que fabriquen al menos 10 productos diferentes.

2.a.1

```
SELECT i.stock_num, i.manu_code, description, manu_name,
       SUM(unit_price*quantity) 'u$ por Producto',
       SUM(quantity) 'Unid. por Producto'
INTO #ABC_Productos
FROM items i JOIN manufact m ON (i.manu_code = m.manu_code)
                JOIN product_types s ON (i.stock_num = s.stock_num)
WHERE i.manu_code IN (SELECT s2.manu_code
                     FROM products s2
                     GROUP BY s2.manu_code
                     HAVING COUNT(*) >= 10)
GROUP BY i.stock_num, i.manu_code, description, manu_name
```

2.a.2

```
SELECT i.stock_num, i.manu_code, description, manu_name,
       SUM(unit_price*quantity) 'u$ por Producto',
       SUM(quantity) 'Unid. por Producto'
INTO #ABC_Productos
FROM items i JOIN manufact m ON (i.manu_code = m.manu_code)
                JOIN product_types s ON (i.stock_num = s.stock_num)
                JOIN (SELECT s2.manu_code FROM products s2
                     GROUP BY s2.manu_code HAVING COUNT(*) >= 10) s3
                ON (i.manu_code=s3.manu_code)
GROUP BY i.stock_num, i.manu_code, description, manu_name
```

2.b Listar los datos generados en la tablas #ABC_Productos ordenados en forma descendente por 'u\$ por Producto' y en forma ascendente por stock_num y manu_code.

```
Select * from #ABC_Productos
order by 'u$ por producto' desc, stock_num, manu_code
```

3. En función a la tabla temporal generada en el punto 2, obtener un listado que detalle para cada producto existente en #ABC_Producto, la descripción del producto, el mes en el que fue solicitado, el cliente que lo solicitó (en formato 'Apellido, Nombre'), la cantidad de órdenes de compra 'Cant OC por mes', la cantidad del producto solicitado 'Unid Producto por mes' y el total en u\$ solicitado 'u\$ Producto por mes'.

Mostrar sólo aquellos clientes que vivan en el estado con mayor cantidad de clientes, ordenado por mes y descripción del tipo de producto en forma ascendente y por cantidad de productos por mes en forma descendente.

```
SELECT description, MONTH(order_date) 'Mes', lname+', '+fname 'Apellido y Nombre',
COUNT(DISTINCT i.order_num) 'Cant OC por mes',
SUM(quantity) 'Unid Producto x mes', SUM(quantity*unit_price) 'u$ Producto
x mes'
FROM orders o JOIN customer c ON (o.customer_num=c.customer_num)
JOIN items i ON (o.order_num = i.order_num)
JOIN #ABC_Productos ABC ON (i.stock_num = ABC.stock_num AND
i.manu_code = ABC.manu_code)
WHERE state = (SELECT TOP 1 state from customer
GROUP BY state
ORDER BY COUNT(customer_num) DESC)
GROUP BY description, MONTH(order_date), lname, fname
ORDER BY MONTH(order_date), description, SUM(quantity) DESC
```

4. Dado los productos con nro de stock 5, 6 y 9 del fabricante 'ANZ' listar de a pares los clientes que hayan solicitado el mismo producto, siempre y cuando, el primer cliente haya solicitado más cantidad del producto que el 2do cliente.

Se deberá informar nro de stock, código de fabricante, Nro de Cliente y Apellido del primer cliente, Nro de cliente y apellido del 2do cliente ordenado por stock_num y manu_code

```
SELECT DISTINCT i1.stock_num, i1.manu_code, c1.customer_num, c1.lname, c2.customer_num,
c2.lname
FROM items i1 JOIN orders o1 ON (o1.order_num = i1.order_num)
JOIN customer c1 ON (o1.customer_num = c1.customer_num)
JOIN items i2 ON (i1.stock_num = i2.stock_num
AND i1.manu_code=i2.manu_code)
JOIN orders o2 ON (o2.order_num = i2.order_num)
JOIN customer c2 ON (o2.customer_num = c2.customer_num)
WHERE i1.stock_num IN (5,6,9)
AND i1.manu_code='ANZ'
AND (SELECT SUM(quantity) FROM items i11
JOIN orders o11 ON (i11.order_num=o11.order_num)
WHERE i11.stock_num=i1.stock_num
AND i11.manu_code=i1.manu_code
AND o11.customer_num = c1.customer_num)
>
(SELECT SUM(quantity) FROM items i12
JOIN orders o12 ON (i12.order_num=o12.order_num)
WHERE i12.stock_num=i2.stock_num
AND i12.manu_code=i2.manu_code
AND o12.customer_num = c2.customer_num)
Order by i1.stock_num, i1.manu_code;
```

5. Se requiere realizar una consulta que devuelva en una fila la siguiente información: La mayor cantidad de órdenes de compra de un cliente, mayor total en u\$ solicitado por un cliente y la mayor cantidad de productos solicitados por un cliente, la menor cantidad de órdenes de compra de un cliente, el menor total en u\$ solicitado por un cliente y la menor cantidad de productos solicitados por un cliente. Los valores máximos y mínimos solicitados deberán corresponderse a los datos de clientes según todas las órdenes existentes, sin importar a que cliente corresponda el dato.

```
SELECT MAX(cantOrd) maxCantOrd, MAX(sumPrecio) maxSumPrecio,
       MAX(cantItem) maxCantItem, MIN(cantOrd) minCantOrd,
       MIN(sumPrecio) minSumPrecio, MIN(cantItem) minCantItem
FROM (SELECT o1.customer_num,
            COUNT(DISTINCT i1.order_num) cantOrd,
            SUM(i1.unit_price * i1.quantity) sumPrecio,
            sum(i1.quantity) cantItem
       FROM orders o1 JOIN items i1 ON (o1.order_num = i1.order_num)
       GROUP BY o1.customer_num) subT
```

6. Seleccionar los número de cliente, número de orden y monto total de la orden de aquellos clientes del estado California(CA) que posean 4 o más órdenes de compra emitidas en el 2015. Además las órdenes mostradas deberán cumplir con la salvedad que la cantidad de líneas de ítems de esas ordenes debe ser mayor a la cantidad de líneas de ítems de la orden de compra con mayor cantidad de ítems del estado AZ en el mismo año.

```
SELECT c.customer_num, o.order_num, SUM(i.unit_price*i.quantity) totalOrdenPorCliente
FROM customer c JOIN orders o ON c.customer_num=o.customer_num
                LEFT JOIN items i ON o.order_num=i.order_num
WHERE c.state = 'CA' AND YEAR(o.order_date)=2015
    and c.customer_num IN
        (SELECT c1.customer_num
         FROM customer c1 JOIN orders o1 ON c1.customer_num=o1.customer_num
         WHERE YEAR(o1.order_date)=2015
         GROUP BY c1.customer_num HAVING COUNT(*)>=4)
GROUP BY c.customer_num, o.order_num
having count(i.item_num) > (SELECT top 1 COUNT(i3.item_num)
                           FROM orders o3 JOIN items i3 ON o3.order_num=i3.order_num
                           JOIN customer c3 ON o3.customer_num=c3.customer_num
                           WHERE YEAR(o3.order_date)=2015 AND c3.state='AZ'
                           GROUP BY o3.order_num
                           ORDER BY COUNT(i3.item_num) DESC)

order by 1,2
```

7. Se requiere listar para el Estado de California el par de clientes que sean los que suman el mayor monto en dólares en órdenes de compra, con el formato de salida:

'Código Estado', 'Descripción Estado', 'Apellido, Nombre', 'Apellido, Nombre', 'Total Solicitado' (*)

(*) El total solicitado contendrá la suma de los dos clientes.

```
SELECT TOP 1 s.state, sname, c1.lname+', '+c1.fname, c2.lname+', '+c2.fname,
            totcli1 + totcli2 'Total'
FROM state s JOIN customer c1 ON (s.state = c1.state)
                JOIN customer c2 ON (s.state = c2.state)
                JOIN (SELECT o1.customer_num, SUM(unit_price*quantity) totcli1
                     FROM orders o1 JOIN items i1 ON (o1.order_num = i1.order_num)
                     GROUP BY o1.customer_num) totc1
                     ON (c1.customer_num = totc1.customer_num)
                JOIN (SELECT o2.customer_num, SUM(unit_price*quantity) totcli2
                     FROM orders o2 JOIN items i2 ON (o2.order_num = i2.order_num)
                     GROUP BY customer_num) totc2 ON (c2.customer_num = totc2.customer_num)
WHERE c1.customer_num > c2.customer_num AND s.state = 'CA'
ORDER BY 5 DESC
```

8. Se observa que no se cuenta con stock suficiente para las últimas 5 órdenes de compra emitidas que contengan productos del fabricante 'ANZ'. Por lo que se decide asignarle productos en stock a la orden del cliente que más cantidad de productos del fabricante 'ANZ' nos haya comprado. Se solicita listar el número de orden de compra, número de cliente, fecha de la orden y una fecha de orden "modificada" a la cual se le suma el *lead_time* del fabricante más 1 día por preparación del pedido a aquellos clientes que no son prioritarios. Para aquellos clientes a los que les entregamos los productos en stock, la "fecha modificada" deberá estar en NULL.

Listar toda la información ordenada por "fecha modificada"

```
SELECT DISTINCT o1.order_num, o1.customer_num, o1.order_date,
                NULL 'Fecha Modificada'
FROM orders o1 JOIN items i1 ON o1.order_num = i1.order_num
                JOIN (SELECT TOP 1 o2.customer_num, SUM(quantity) totcliente
                      FROM orders o2 JOIN items i2 ON o2.order_num = i2.order_num
                      WHERE i2.manu_code='ANZ'
                      GROUP BY o2.customer_num
                      ORDER BY 2 DESC) sub1 ON o1.customer_num = sub1.customer_num
WHERE o1.order_num IN (SELECT DISTINCT TOP 5 o2.order_num
                      FROM orders o2 JOIN items i2 ON o2.order_num = i2.order_num
                      WHERE i2.manu_code='ANZ'
                      ORDER BY o2.order_num DESC)

UNION

SELECT DISTINCT o1.order_num, o1.customer_num, o1.order_date,
                o1.order_date + m1.lead_time + 1 'Fecha Modificada'
FROM orders o1 JOIN items i1 ON (o1.order_num = i1.order_num)
                JOIN (SELECT TOP 1 o2.customer_num, SUM(quantity) totcliente
                      FROM orders o2 JOIN items i2
                      ON (o2.order_num = i2.order_num)
                      WHERE i2.manu_code='ANZ'
                      GROUP BY o2.customer_num
                      ORDER BY 2 DESC) sub1
                ON (o1.customer_num <> sub1.customer_num)
                JOIN manufact m1
                ON (i1.manu_code = m1.manu_code)
WHERE o1.order_num IN (SELECT DISTINCT TOP 5 o2.order_num
                      FROM orders o2 JOIN items i2
                      ON (o2.order_num = i2.order_num)
                      WHERE i2.manu_code='ANZ'
                      ORDER BY o2.order_num DESC)
AND i1.manu_code = 'ANZ'
ORDER BY 4
```

9. Listar el numero, nombre, apellido, estado, cantidad de ordenes y monto total comprado de los clientes que no sean del estado de Wisconsin y cuyo monto total comprado sea mayor que el monto total promedio de órdenes de compra.

```
select c.customer_num, c.fname, c.lname, c.state,
       count(distinct o.order_num) cantOrdenes,
       sum(i.unit_price * i.quantity) MontoOrdenes
from customer c join orders o on c.customer_num = o.customer_num
                join items i on o.order_num = i.order_num
where c.state != 'WI'
group by c.customer_num, c.fname, c.lname, c.state
having sum(i.unit_price * i.quantity) >=
       (select sum(i.unit_price * i.quantity)
        / count(distinct o.order_num)
        from orders o join items i on o.order_num = i.order_num);
```