

1.

Listar Número de Cliente, apellido y nombre, Total Comprado por el cliente 'Total del Cliente', la Cant. de Órdenes de Compra solicitadas por el cliente 'OCs del Cliente' y la Cant. de Órdenes de Compra solicitadas por todos los clientes 'Cant. Total OC'.

Para los clientes cuyo promedio comprado por Orden supere al promedio comprado por orden de compra general y hayan solicitado por lo menos 2 órdenes y el zipcode comience con 94.

```
SELECT c.customer_num, fname, lname,
       SUM(total_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(total_price*quantity)/COUNT(DISTINCT i.order_num))
       > (SELECT (SUM(total_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(total_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(total_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 por 'u\$ por Producto', stock_num y manu_code de Mayor a menor por precio y menor a mayor por stock_num y manu_code.

```
SELECT stock_num, manu_code, "u$ por Producto"
FROM #ABC_Productos
ORDER BY 3,1 DESC, 2 DESC
```

3. En función a la tabla temporal generada en el punto 2, obtener un listado que detalle por 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 de producto solicitado 'Unid Producto por mes' y el total en u\$ solicitado 'u\$ Producto por mes'.

Sólo se deberán mostrar los clientes que vivan en el estado con mayor cantidad de clientes. Deberá estar ordenado por mes y producto.

```
SELECT description, MONTH(order_date) 'Mes', lname+', '+fname 'Apellido y Nombre',
COUNT(DISTINCT i.order_num) 'Cant OC por mes',
SUM(quantity) 'Unid Producto por mes',
SUM(total_price) 'u$ Producto por 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 state from customer
GROUP BY state
HAVING count(state) =
(SELECT TOP 1 COUNT(customer_num)
FROM customer c3 GROUP BY state
ORDER BY 1 DESC))
GROUP BY description, MONTH(order_date), lname, fname
ORDER BY 1,2,3
```

Opción más fácil

```
SELECT description, MONTH(order_date) 'Mes', lname+', '+fname 'Apellido y Nombre',
COUNT(DISTINCT i.order_num) 'Cant OC por mes', SUM(quantity) 'Unid Producto
por mes',
SUM(total_price) 'u$ Producto por 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 1,2,3
```

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

```
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)

```