

1) Crear una vista que devuelva:

- a) Código y Nombre (manu_code, manu_name) de los fabricante, posean o no productos (en tabla stock), cantidad de productos que poseen en tabla stock (cant_producto) y la fecha de la última OC que contenga un producto suyo (ult_fecha_orden).
- De los fabricantes que fabriquen productos sólo se podrán mostrar los que fabriquen más de un producto.
 - No se permite utilizar funciones definidas por usuario, ni tablas temporales, ni UNION.
- b) Realizar una consulta sobre la vista que devuelva manu_code, manu_name, cant_producto y si el campo ult_fecha_orden posee un NULL informar 'No Posee Órdenes' si no posee NULL informar el valor de dicho campo.
- No se puede utilizar UNION para el SELECT.

```
create view vrecul as
SELECT m.manu_code, m.manu_name, count(stock_num) cant_productos,
      (SELECT max(order_date) FROM orders o JOIN items i
       ON o.order_num=i.order_num AND i.manu_code=m.manu_code)
      ult_compra
FROM manufact m LEFT JOIN products s ON s.manu_code =
m.manu_code
GROUP BY m.manu_code, m.manu_name
HAVING count(stock_num)=0 OR count(stock_num)>1
```

```
create view vrecul as
SELECT m.manu_code, m.manu_name, count(distinct s.stock_num)
cant_productos,
max(o.order_date) ult_compra
FROM manufact m
LEFT JOIN products s ON s.manu_code = m.manu_code
LEFT JOIN items i ON s.manu_code = i.manu_code AND
s.stock_num=i.stock_num
LEFT JOIN orders o ON i.order_num = o.order_num
GROUP BY m.manu_code, m.manu_name
HAVING count(distinct s.stock_num)=0 OR count(distinct s.stock_num)>1
```

```
create view vrecul as
SELECT m.manu_code, m.manu_name, count(distinct s.stock_num)
cant_productos,
max(o.order_date) ult_compra
FROM manufact m
LEFT JOIN products s ON s.manu_code = m.manu_code
LEFT JOIN items i ON s.manu_code = i.manu_code AND
s.stock_num=i.stock_num
LEFT JOIN orders o ON i.order_num = o.order_num
WHERE m.manu_code IN
(SELECT m2.manu_code FROM manufact m2 JOIN stock s2
ON (m2.manu_code = s2.manu_code)
GROUP BY m2.manu_code
HAVING COUNT(*) >1 OR COUNT(*) = 0)
GROUP BY m.manu_code, m.manu_name
```

```
select manu_code, manu_name, cant_productos,
      case when ult_compra is null then 'No posee Compras'
```

```
                when ult_compra is not null then cast(ult_compra as char)
end
from vrecul
```

2) Desarrollar una consulta muestre un ABC de fabricantes que:

Liste el código de fabricante, el nombre del fabricante, la cantidad de órdenes de compra que contentan sus productos y la suma total los productos vendidos.

Se deberán tener en cuenta sólo los fabricantes cuyo código comience con H y posea 3 letras, y los productos cuya descripción posea el string "tennis" ó el string "ball".

Sólo se podrán mostrar los datos de los fabricantes cuyo total sea mayor que el total de ventas promedio por cada fabricante (Cantidad vendida / Cantidad de fabricantes que tuvieron productos vendidos).

La consulta deberá mostrar los registros ordenados por total vendido de mayor a menor.

```
SELECT m.manu_code,m.manu_name,
       COUNT(DISTINCT o.order_num) cantidadOrdenes,
       SUM(total_price) totalComprado
FROM   manufact m JOIN items i ON m.manu_code=i.manu_code
JOIN orders o ON o.order_num=i.order_num
JOIN products s ON (i.manu_code=s.manu_code AND
i.stock_num=s.stock_num)
WHERE description LIKE '%tennis%' OR description LIKE '%ball%'
AND    m.manu_code LIKE 'H__'
GROUP BY m.manu_code,m.manu_name
HAVING SUM(total_price)
> (select SUM(total_price)/count(DISTINCT i.manu_code)
    from items i)
ORDER BY 4 DESC
```

3) Crear una vista que devuelva

Mostrar los datos (customer_num,lname,company) de los clientes, posean o no órdenes de compra y la cantidad de órdenes de compra, la fecha de la última OC el total en u\$s (total_price)comprado y el total general Comprado por todos los clientes.

De los clientes que posean órdenes sólo se podrán mostrar los clientes que tengan alguna orden que posea productos que son fabricados por más de dos fabricantes. Mostrar los clientes que posean menos de 5 órdenes de compra.

Ordenar el reporte primero por los clientes que tengan órdenes por cantidad de órdenes descendente y luego por los clientes que no tengan órdenes

No se permite utilizar funciones, ni tablas temporales.

```
CREATE VIEW v_parcial AS
select c.customer_num, c.lname, c.company, sname,
       0 cantidad_ordenes, null ultima_compra, 0 total_ordenes,
       (select sum(total_price) FROM items) total_general
from customer c
inner join state s on c.state =s.code
where customer_num not in (select distinct customer_num from orders)
UNION
select c.customer_num, c.lname, c.company, sname, MAX(order_date),
       count(distinct o.order_num), sum(i.total_price),
       (select sum(total_price) FROM items)
```

```

from customer c
inner join orders o on c.customer_num=o.customer_num
inner join items i on o.order_num = i.order_num
inner join state s on c.state =s.code
where c.customer_num in
    (select DISTINCT o2.customer_num from orders o2
     INNER JOIN items i2 ON o2.order_num=i2.order_num
     WHERE i2.stock_num IN (SELECT stock_num FROM stock
     GROUP BY stock_num HAVING count(*) <=2))
group by c.customer_num,c.lname,c.company,sname
having count(distinct o.order_num) < 5

SELECT * FROM v_parcial
order by 5 DESC, 1

```

4) Crear una vista que devuelva

El top 5 de los productos (description) que fueron más comprados en cada estado (state) con la cantidad vendida y su precio total, teniendo en cuenta que solo se mostrará el estado en el que tuvo mayor cantidad de ventas de un mismo producto.

Ordenarlo por la cantidad vendida descendiente.

No se permite utilizar funciones, ni tablas temporales.

```

CREATE VIEW productMasComprados AS

SELECT TOP 5 s.description, c.state, SUM(i.quantity) cantidadVendida,
SUM(i.total_price) totalVendido
FROM products s JOIN items i ON (s.stock_num = i.stock_num)
JOIN orders o ON (i.order_num = o.order_num)
JOIN customer c ON (o.customer_num = c.customer_num)
GROUP BY s.description, c.state
HAVING SUM(i.quantity) > (SELECT TOP 1 SUM(il.quantity) FROM stock s1
                        JOIN items il ON (s1.stock_num =
                        il.stock_num)
                        JOIN orders o1 ON (il.order_num =
o1.order_num)
                        JOIN customer c1 ON (o1.customer_num =
c1.customer_num)
                        WHERE c1.state = c.state AND
s1.description != s.description
                        GROUP BY c1.state, s1.description
                        ORDER BY SUM(il.quantity) DESC)

SELECT * FROM productMasComprados
order by cantidadVendida DESC

```