

1. Mostrar el código y nombre de las provincias y la cantidad de clientes que contienen utilizando subqueries.
2. Seleccionar producto_cod, producto_desc, de los productos cuyas cantidades totales vendidas superen las 150 unidades. Utilizar un subquery en el FROM para su resolución. De qué otra manera se puede solucionar este ejercicio?
3. Mostrar los datos de las provincias que no posean fabricantes.
4. Mostrar código y tiempo de entrega de todos los fabricantes que **no sean** de 'BA' cuyo plazo entrega sea menor o igual a los plazos de entrega de todos los fabricantes de 'BA'.
5. Mostrar el código y suma vendida de todos los fabricantes que no sean de BA que hayan vendido productos por un monto mayor al de algún fabricante de 'BA'.
6. Seleccionar aquellos clientes cuya facturación supere el promedio facturado de todos los clientes que realizaron compras.
7. Seleccionar aquellos productos que no hayan tenido ventas. Resolverlo de dos maneras diferentes: Con subquery y con un query correlacionado.
8. Seleccionar número de todos los clientes que compraron productos del fabricante 'DOTO'.
9. Seleccionar número, nombre y apellido de todos los clientes que NO compraron productos del fabricante 'DOTO'.
10. Seleccionar número, nombre y apellido de los clientes que hayan comprado TODOS los productos del fabricante 'DOTO'.
11. Tome el ejemplo del apunte de query recursivo, qué debería modificar para que el query muestre todos los "clientes referentes hacia arriba" de un cierto cliente?

-
12. Seleccionar el número, nombre y apellido de los clientes, Monto Total Comprado (p*q) y cantidad de facturas por cliente del producto (producto_cod) con mayor monto total vendido. Mostrar también el monto total del producto mas vendido.

Para el cálculo del mayor producto vendido como de las facturas, se deben considerar **solamente** las facturas posteriores al 15 de marzo del 2021.

Mostrar la información ordenada por el monto total comprado del producto por cliente en forma descendente y por cantidad de facturas en forma ascendente.

Ej. Si el producto más vendido fue el 1010 con ventas totales por \$4.499,91

cliente_num	Nombre	Apellido	MontoTotal ↓	Cantidad Facturas	Monto Producto
122	Cathy	O Brian	3.000,00	3	4,499,91
18	Robert	Duval	1.100,91	1	4,499,91
23	Jack	Sparrow	399,00	1	4,499,91

Notas: No usar Store procedures, ni funciones de usuarios, ni tablas temporales. Una orden puede tener varios Items con el mismo producto.

13. Crear una consulta que devuelva lo siguiente:

Número, Apellido y Nombre del cliente,

Monto total comprado del cliente

Número, Apellido, Nombre del Cliente Referido (ojo, No el referente ¡!)

Monto total comprado por el referido mas un 10% de comisión

Consideraciones.

En el caso que un Cliente no tenga Referidos deberá mostrar los datos del referido en NULL.
Para calcular la comisión del cliente se deberán sumar (cant*precio) de todos los productos comprados por el Cliente Referido cuyo código de producto sea menor a 1010
Se deberá ordenar la salida por el nro de cliente y nro de referido ambos ascendentes.

14.

Crear una consulta que devuelva: Para aquellos **3 provincias de mayor facturación** el Código de Provincia, número cliente, nombre, apellido, promedio de facturas de compra por cliente, total comprado por cliente y total comprado por provincia. Solo tener en cuenta aquellas líneas de facturas que hayan facturado más de \$1000.

Ordenar la consulta por el monto facturado total por provincia en forma descendente y por monto facturado por cliente también en forma descendente.

15.

Desarrollar una consulta que devuelva por cada fabricante dos filas, una con el producto más comprado y otra con el producto menos comprado, indicando el tipo de registro ('1-más vendido', '2-menos vendido').

Los atributos a mostrar son código y nombre del fabricante, código de producto, monto total vendido y la leyenda "más vendido" o "menos vendido".

Ordenar la info por código de fabricante y tipo.

Ejemplo

Fabricante	Nombre	Producto	Monto	Tipo
ALAS	Alanis	1002	6850.00	Mas Vendido
ALAS	Alanis	1002	6850.00	Menos Vendido
BERI	Benemerito	1003	9200.00	Mas Vendido
BERI	Benemerito	1003	3200.00	Menos Vendido
CASA	Camino Salasa	1004	22800.00	Mas Vendido
CASA	Camino Salasa	1013	2000.00	Menos Vendido

Por cada fabricante deberían haber 2 filas, en el caso que tengan un solo producto comprado, el mismo se repetirá como masComprado y menosComprado.

1.

```
select provincia_cod, provincia_desc,  
       (select count(*) from clientes c where c.provincia_cod = p.provincia_cod) total  
from provincias p
```

2.

```
select p.producto_cod, p.producto_desc  
from productos p join (select producto_cod  
                        from facturas_det d  
                        group by producto_cod  
                        having sum(cantidad) > 150) d  
on      p.producto_cod = d.producto_cod
```

```
select p.producto_cod, producto_desc, sum(cantidad)  
from productos p join facturas_det d on p.producto_cod = d.producto_cod  
group by p.producto_cod, producto_desc  
having sum(cantidad) > 150
```

3.

```
select * from provincias  
where provincia_cod not in (select provincia_cod from fabricantes)
```

4.

```
SELECT fabricante_cod, tiempo_entrega  
FROM fabricantes  
where provincia_cod != 'BA' and tiempo_entrega <= ALL  
      (select tiempo_entrega from fabricantes  
       where provincia_cod = 'BA'  
       and tiempo_entrega is not null)
```

5.

```
SELECT f.fabricante_cod, sum(d.cantidad * d.precio_unit) total  
FROM fabricantes f join productos p on f.fabricante_cod = p.fabricante_cod  
join facturas_det d on p.producto_cod = d.producto_cod  
where f.provincia_cod != 'BA'  
group by f.fabricante_cod  
HAVING sum(d.cantidad * d.precio_unit) > any  
      (SELECT sum(d.cantidad * d.precio_unit) total  
       FROM fabricantes f join productos p on f.fabricante_cod = p.fabricante_cod  
       join facturas_det d on p.producto_cod = d.producto_cod  
       where f.provincia_cod = 'BA'  
       group by f.fabricante_cod)
```

6.

```
Select c.cliente_num, sum(cantidad * precio_unit) total  
from clientes c join facturas f on c.cliente_num = f.cliente_num  
join facturas_det d on d.factura_num = f.factura_num  
group by c.cliente_num
```

```

having sum(cantidad * precio_unit) >
    (select sum(cantidad * precio_unit) /
        count(distinct cliente_num)
        from facturas f join facturas_det d
            on d.factura_num = f.factura_num)

```

7.

```

select p.producto_cod, p.producto_desc
    from productos p
    where p.producto_cod not in (select d.producto_cod
                                from facturas_det d)

```

```

select p.producto_cod, p.producto_desc
    from productos p
    where not exists (select d.producto_cod
                      from facturas_det d
                      where p.producto_cod = d.producto_cod)

```

8.

```

select DISTINCT f.cliente_num
    from facturas f join facturas_Det d on f.factura_num = d.factura_num
                    join productos p on d.producto_cod = p.producto_cod
    where p.fabricante_cod = 'DOTO'

```

9.

```

select c.cliente_num, c.nombre, c.apellido
    from clientes c
    where c.cliente_num not in (select f.cliente_num
                                from facturas f join facturas_Det d
                                    on f.factura_num = d.factura_num
                                join productos p
                                    on d.producto_cod = p.producto_cod
                                where p.fabricante_cod = 'DOTO')

```

10. select cliente_num, nombre, apellido

```

    from clientes c
    where not exists (
        select producto_cod
            from productos
            where fabricante_cod = 'DOTO'
        EXCEPT
        select d.producto_cod
            from facturas f join facturas_det d on f.factura_num = d.factura_num
            where f.cliente_num = c.cliente_num );

```

```

select f.cliente_num, c.nombre, c.apellido
    from facturas f join facturas_det fd on f.factura_num = fd.factura_num
    join productos p on fd.producto_cod = p.producto_cod

```

```

join clientes c on f.cliente_num = c.cliente_num
where fd.producto_cod in (select producto_cod
                        from productos
                        where fabricante_cod = 'DOTO')
group by f.cliente_num, c.nombre, c.apellido
having count(distinct fd.producto_cod) = (select count(producto_cod)
                                         from productos
                                         where fabricante_cod = 'DOTO')

```

11. Dé vuelta la condición.

```

12.
select c.cliente_num, c.nombre, c.apellido,
       sum(d.cantidad * d.precio_unit) MontoTotal,
       count(distinct f.factura_num) Cantidad, b.total
from clientes c join facturas f on c.cliente_num = f.cliente_num
                join facturas_det d on d.factura_num = f.factura_num
                join (select top 1 d.producto_cod, sum(d.cantidad*d.precio_unit) total
                      from facturas_det d join facturas f on d.factura_num = f.factura_num
                      where f.fecha_emision >= '2021-03-15'
                      group by d.producto_cod
                      order by 2 desc) b on d.producto_cod = b.producto_cod
where f.fecha_emision > '2021-03-15'
group by c.cliente_num, c.nombre, c.apellido, b.total
order by MontoTotal DESC, Cantidad

```

```

13.
SELECT c1.cliente_num, c1.nombre, c1.apellido, SUM(d1.cantidad * d1.precio_unit) total,
       r.cliente_num, r.nombre, r.apellido, r.totalCompraReferido*1.1 AS totalComision
FROM clientes c1 JOIN facturas f1 ON c1.cliente_num = f1.cliente_num
                JOIN facturas_det d1 ON f1.factura_num = d1.factura_num
LEFT JOIN (SELECT c2.cliente_num, c2.nombre, c2.apellido,
                  SUM(d2.precio_unit * d2.cantidad) AS totalCompraReferido,
                  c2.cliente_ref
            FROM clientes c2 JOIN facturas f2 ON c2.cliente_num=f2.cliente_num
                JOIN facturas_det d2 ON d2.factura_num=f2.factura_num
                WHERE d2.producto_cod < 1010
                GROUP BY c2.cliente_num, c2.nombre, c2.apellido,
                        c2.cliente_ref) AS R
ON r.cliente_ref=c1.cliente_num
GROUP BY c1.cliente_num, c1.nombre, c1.apellido, R.cliente_num,
         r.nombre, r.apellido, r.TotalCompraReferido
ORDER BY c1.cliente_num, r.cliente_num

```

```

14.
SELECT c.provincia_cod provincia, c.cliente_num CodigoDeCliente,
       c.nombre, c.apellido,
       SUM(d.cantidad*d.precio_unit)/COUNT(DISTINCT d.factura_num)
PromedioPorCliente,
       SUM(d.cantidad*d.precio_unit) TotalPorCliente,
       TotalPorProvincia
FROM clientes c
JOIN facturas o ON c.cliente_num = o.cliente_num
JOIN facturas_det d ON o.factura_num = d.factura_num
JOIN (SELECT top 3 c1.provincia_cod Provincia,

```

```

        SUM(d.cantidad*d.precio_unit) TotalPorProvincia
    FROM clientes c1
        JOIN facturas f ON c1.cliente_num = f.cliente_num
        JOIN facturas_det d ON f.factura_num = d.factura_num
    WHERE d.cantidad*d.precio_unit > 1000
        GROUP BY c1.provincia_cod ORDER BY 2 DESC) tc
        ON tc.provincia = c.provincia_cod
WHERE d.cantidad*d.precio_unit > 1000
GROUP BY c.provincia_cod, c.cliente_num, c.nombre, c.apellido, TotalPorProvincia
ORDER BY TotalPorProvincia DESC, TotalPorCliente DESC

```

Otra solución utilizando WITH

```

WITH clientesTot AS (
    SELECT c.provincia_cod, c.cliente_num, c.nombre, c.apellido,
        SUM(fd.precio_unit * fd.cantidad) /
            COUNT(distinct f.factura_num) AS promedio_compra_cliente,
        SUM(fd.precio_unit * fd.cantidad) AS total_comprado_cliente
    FROM clientes c JOIN facturas f ON c.cliente_num = f.cliente_num
        JOIN facturas_det fd ON f.factura_num = fd.factura_num
    WHERE fd.precio_unit * fd.cantidad > 1000 -- Filtrar por ítems con gasto mayor a 1000
    GROUP BY c.provincia_cod, c.cliente_num, c.nombre, c.apellido),
provinciasTot AS (
    SELECT top 3 c.provincia_cod,
        SUM(fd.precio_unit * fd.cantidad) totalProvincia
    FROM clientes c JOIN facturas f ON c.cliente_num = f.cliente_num
        JOIN facturas_det fd ON f.factura_num = fd.factura_num
    WHERE fd.precio_unit * fd.cantidad > 1000 -- Filtrar por ítems con gasto mayor a 1000
    GROUP BY c.provincia_cod
    order by totalProvincia desc
)
SELECT p.provincia_cod,
    c.cliente_num,
    c.nombre,
    c.apellido,
    c.promedio_compra_cliente,
    c.total_comprado_cliente,
    p.totalProvincia
FROM clientesTot c JOIN provinciasTot p ON c.provincia_cod = p.provincia_cod
ORDER BY p.totalProvincia DESC,
    c.total_comprado_cliente DESC;

```

15.

```

select f.fabricante_cod, fabricante_nom, p.producto_cod,
    sum(d.precio_unit*cantidad), 'Mas Vendido'
    from fabricantes f join productos p on f.fabricante_cod = p.fabricante_cod
        join facturas_det d on d.producto_cod = p.producto_cod
group by f.fabricante_cod, fabricante_nom, p.producto_cod
having sum(d.precio_unit*cantidad) = (select top 1 sum(d2.precio_unit*d2.cantidad) total
    from fabricantes f2 join productos p2 on f2.fabricante_cod =
p2.fabricante_cod
        join facturas_det d2 on d2.producto_cod = p2.producto_cod
        where f2.fabricante_cod = f.fabricante_cod
        group by f2.fabricante_cod, p2.producto_cod
    order by 1 desc)
UNION

```

```
select f.fabricante_cod, fabricante_nom, p.producto_cod,  
       sum(d.precio_unit*cantidad), 'Menos Vendido'  
   from fabricantes f join productos p on f.fabricante_cod = p.fabricante_cod  
                        join facturas_det d on d.producto_cod = p.producto_cod  
 group by f.fabricante_cod, fabricante_nom, p.producto_cod  
        having sum(d.precio_unit*cantidad) = (  
            select top 1 sum(d2.precio_unit*d2.cantidad) total  
              from fabricantes f2 join productos p2 on f2.fabricante_cod =  
p2.fabricante_cod  
                join facturas_det d2 on d2.producto_cod = p2.producto_cod  
               where f2.fabricante_cod = f.fabricante_cod  
             group by f2.fabricante_cod, p2.producto_cod  
             order by 1)  
 order by 1 desc, 5
```