```sql
-- Ejercicio - SQL Técnicas Avanzadas


-----------------------------------------------
-- Generar un listado de clientes e ítems que incluya
-- un subquery dentro del SELECT que muestre
-- las últimas fechas de compra de cada uno de ellos
-----------------------------------------------


-- Los primeros dos queries generan lo que se pide en el enunciado.
-- Sin necesidad de hacer un subquery (de todas formas lo dejé comentado)

select
    CL.apellido,
    CL.nombre,
    I.descripcion,
    MAX(IV.fecha) as ultimaCompra
from clientes CL
inner join ventasporcliente VC on CL.idcliente=VC.idcliente
inner join itemsporventa IV on VC.idventa = IV.idventa
inner join inventario I on IV.iditem = I.iditem
-- where IV.iditem in (select distinct  IV.iditem from itemsporventa IV)
group by 1, 2 , 3
order by 3, 4


select
    CL.apellido,
    CL.nombre,
    I.descripcion,
    MAX(IV.fecha) as ultimaCompra
from clientes CL
inner join ventasporcliente VC on CL.idcliente=VC.idcliente
inner join itemsporventa IV on VC.idventa = IV.idventa
inner join inventario I on IV.iditem = I.iditem
-- where VC.idcliente in (select distinct  VC.idcliente from ventasporcliente VC)
group by 1, 2 , 3
order by 3, 4

-- En este query, en cambio, si use un subquery para encontrar cual fue el último bien
-- que ha sido vendido.

select
    CL.apellido,
    CL.nombre,
    I.descripcion,
    IV.fecha as ultimaCompra
from clientes CL
inner join ventasporcliente VC on CL.idcliente=VC.idcliente
inner join itemsporventa IV on VC.idventa = IV.idventa
inner join inventario I on IV.iditem = I.iditem
where IV.fecha in (select max(IV.fecha) from itemsporventa IV)
order by 1

-- Este query genera la lista de la fecha en la cual cada ítem fue vendido por última vez
-- (sin la información del cliente)

select
```

```
    I.descripcion,
    MAX(IV.fecha) as ultimaCompra
from itemsporventa IV
inner join inventario I on IV.iditem = I.iditem
--where IV.fecha in (select distinct  max(IV.fecha) from itemsporventa IV GROUP BY
IV.iditem)
group by 1
order by 1,2

-- Este query genera la lista de la fecha en la cual los clientes realizaron su última
compra
-- (sin la información del ítem)

select
    CL.apellido,
    CL.nombre,
    MAX(VC.fecha) as ultimaCompra
from ventasporcliente VC
inner join clientes CL on CL.idcliente  = VC.idcliente
group by 1,2
order by 3


-------------------------------------------------------
-- Construir una vista que muestre:
-- Id Cliente,
-- # de ventas por cliente
-------------------------------------------------------

create or replace view v_clientexnumventas as
select
    idcliente,
    count(idcliente) as numVentas
from ventasporcliente VC
group by idcliente
order by 2 desc;


select * from v_clientexnumventas;

-------------------------------------------------------
-- Construir una vista que detalle:
-- Id Item,
-- Descripción del Item,
-- # de Compras (cantidad),
-- Id Cliente y
-- Nombre Cliente.
-------------------------------------------------------
create or replace view v_detalladaitempsxcliente as
select
    I.iditem,
    I.descripcion,
    count(I.iditem) as Cantidad,
    CL.idcliente,
    CL.apellido || ' ' || CL.nombre as NombreCliente
from clientes CL
inner join ventasporcliente VC on CL.idcliente=VC.idcliente
inner join itemsporventa IV on VC.idventa = IV.idventa
inner join inventario I on IV.iditem = I.iditem
```

```
group by (1,2,4,5)
order by 1;

select * from v_detalladaitempsxcliente;


-------------------------------------------------------
-- Construir una vista que muestre:
-- Fecha (aaaa-mm-dd),
-- Id Venta,
-- Id Cliente,
-- Total Venta
-------------------------------------------------------

create or replace view v_tablaVentasXCliente as
select * from ventasporcliente VC
order by 1;

select * from v_tablaVentasXCliente;

-------------------------------------------------------
-- Construir un Procedimiento Almacenado que añada un campo
-- ULT_FECHA_COMPRA,
-- ULT_ITEM_COMPRA
-- a la tabla maestro de clientes y que actualice la información cada vez que
-- es llamado. El procedimiento recibe como información la fecha y el id item
-------------------------------------------------------

select * from clientes c

ALTER TABLE clientes
ADD ULT_FECHA_COMPRA date,
ADD ULT_ITEM_COMPRA CHARACTER VARYING(20);


-- Ya que en ejercicios anteriores fue especificado que por cada orden de compra
-- se incluyeran varios items, en el momento en el cual añado la columna "ULT_ITEM_COMPRA"
-- a la tabla maestra cliente me va a arrojar más de un resultado por cliente.
-- Lo que representaría un problema ya que tendría llaves primarias duplicadas por cada
registro.
-- Por lo cual, como solución opté por incluir una columna (ULT_ID_COMPRA)
-- que contenga el ID de la última venta hecha por el cliente,en vez de ULT_ITEM_COMPRA.

ALter table clientes
rename column ULT_ITEM_COMPRA to ULT_COMPRA;

Alter table clientes
rename COLUMN ULT_COMPRA to ULT_ID_COMPRA;

select * from clientes c

-- Este sería el codigo a tener en cuenta
select
    VC.idventa,
    VC.fecha
from ventasporcliente VC
where VC.idcliente ='10' and vc.fecha in
        (select max(VC.fecha)
        from ventasporcliente VC
```

```
        where vc.idcliente ='10')

-- Creo dos funciones que saran utilizadas en el Procedimiento Almacenado:


create function SP_get_ult_id_venta(idcl CHARACTER VARYING(20))
returns varchar
language plpgsql
as
$$
declare
ult_id_venta CHARACTER VARYING(20);

begin
    select VC.idventa
    into ult_id_venta
    from ventasporcliente VC
    where VC.idcliente =idcl and vc.fecha in
        (select max(VC.fecha)
        from ventasporcliente VC
        where vc.idcliente =idcl);

    return ult_id_venta;
end;
$$

--select SP_get_ult_id_venta('1')

create function SP_get_ult_fecha_venta(idcl CHARACTER VARYING(20))
returns date
language plpgsql
as
$$
declare
ult_fecha_venta date;

begin
    select VC.fecha
    into ult_fecha_venta
    from ventasporcliente VC
    where VC.idcliente =idcl and vc.fecha in
        (select max(VC.fecha)
        from ventasporcliente VC
        where vc.idcliente =idcl);

    return ult_fecha_venta;
end;
$$

--select SP_get_ult_fecha_venta('1')

-- Creo los procedimientos almacenados

drop procedure SP_update_clientes;

create or replace procedure SP_update_clientes( id varchar)
language plpgsql
as $$
begin
```

```
    update clientes
    set ULT_ID_COMPRA = SP_get_ult_id_venta(id),
        ULT_FECHA_COMPRA = SP_get_ult_fecha_venta(id)
    where idcliente = id;

end;
$$;

call SP_update_clientes('1');
call SP_update_clientes('2');
call SP_update_clientes('3');
call SP_update_clientes('4');
call SP_update_clientes('5');
call SP_update_clientes('6');
call SP_update_clientes('7');
call SP_update_clientes('8');
call SP_update_clientes('9');
call SP_update_clientes('10');


SELECT * FROM clientes c


-------------------------------------------------------------
-- Construir un Procedimiento Almacenado que genere un ranking
-- de todos los ítems por Ventas y # Clientes
-------------------------------------------------------------


-- procedimiento Almacenado que genere un ranking
-- de todos los ítems por Ventas

create or replace procedure SP_ranking_items_by_ventas()
language plpgsql
as $$
begin
    create or replace view v_ranking_items_by_ventas as
    select
        i.descripcion,
        sum(cantidad) as total_cantidad,
        precio* sum(cantidad) as total_venta,
        RANK () OVER (  ORDER BY precio* sum(cantidad) DESC) rank_number_by_total_venta
        from itemsporventa iv
        inner join inventario i on i.iditem = iv.iditem
        group by 1, precio;

end;
$$

call sp_ranking_items_by_ventas();
select * from v_ranking_items_by_ventas;


-- procedimiento Almacenado que genere un ranking
-- de todos los ítems por # clientes (lo tomé como unidades vendidas)
create or replace procedure SP_ranking_items_by_cantidad()
language plpgsql
as $$
begin
```

```
    create or replace view v_ranking_items_by_cantidad as
    select
    i.descripcion,
    sum(cantidad) as total_cantidad,
    precio* sum(cantidad) as total_venta,
    RANK () OVER (  ORDER BY sum(cantidad) DESC) rank_number_by_cantidad_venta
    from itemsporventa iv
    inner join inventario i on i.iditem = iv.iditem
    group by 1, precio;
end;
$$

call sp_ranking_items_by_cantidad() ;
select * from v_ranking_items_by_cantidad;
```