**Objetivo:** Resolver una serie de ejercicios utilizando SQL para trabajar con un modelo de base de datos de ventas. Las consultas deben reflejar la capacidad para manejar datos, realizar agregaciones, y aplicar funciones de agrupación y filtrado.

**Instrucciones:** Resuelve los siguientes ejercicios basándote en un modelo de base de datos que incluye información sobre productos, clientes, ventas, categorías, departamentos y ciudades. Escribe consultas SQL para responder a cada pregunta. Asegúrate de probar tus consultas y verificar los resultados.

**Solución:**

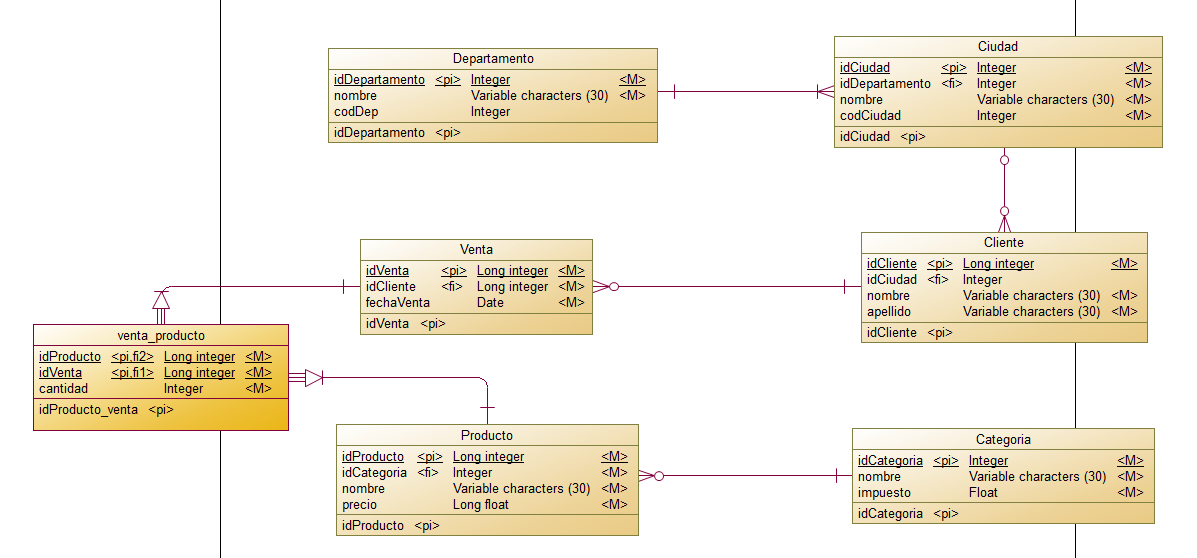
Primero se realiza el modelado del esquema de base de datos a utilizar

* **Modelo entidad- relación:**

**Diagrama

Descripción generada automáticamente**

* **Modelo relacional:**

****

**Ejercicios**

1. Producto menos vendido por ciudad (cantidad):

Escribe una consulta SQL para encontrar el producto que ha sido menos vendido

(en términos de cantidad) en cada ciudad. La salida debe incluir:

* Nombre de la ciudad
* Nombre del producto
* Cantidad vendida

select ciu.nombre ciudad,p.nombre producto,vp.cantidad cantidad

from cliente c,ciudad ciu,venta v,venta\_producto vp,producto p

where c.idciudad = ciu.idciudad

and v.idcliente = c.idcliente

and v.idventa = vp.idventa

and vp.idproducto = p.idproducto

and (ciu.idciudad,vp.cantidad) in (select ciu.idciudad ciudad ,min(vp.cantidad)

from cliente c,ciudad ciu,venta v,venta\_producto vp,producto p

where c.idciudad = ciu.idciudad

and v.idcliente = c.idcliente

and v.idventa = vp.idventa

and vp.idproducto = p.idproducto

group by ciu.idciudad

);

Tabla

Descripción generada automáticamente

2. Producto menos vendido por mes (cantidad):

Escribe una consulta SQL para determinar el producto que ha sido menos vendido

(en términos de cantidad) cada mes. La salida debe incluir:

* Mes (en formato YYYY-MM)
* Nombre del producto
* Cantidad vendida

with product\_por\_mes as(

    select vp.idproducto producto,v.fechaventa fecha,sum(vp.cantidad) cantidad

    from venta v,venta\_producto vp,producto p

    where v.idventa = vp.idventa

    and p.idproducto = vp.idproducto

    group by vp.idproducto,v.fechaventa

    order by v.fechaventa,vp.idproducto

)

Tabla

Descripción generada automáticamente

3. Cliente que más ha comprado $$$ en cada ciudad:

Escribe una consulta SQL para encontrar el cliente que ha gastado más dinero en

cada ciudad. La salida debe incluir:

* Nombre de la ciudad
* Nombre del cliente
* Total gastado

with compra\_cliente as (

    select c.idcliente cliente,sum(p.precio\*vp.cantidad) total

    from cliente c, ciudad ci,venta v,venta\_producto vp, producto p

    where c.idciudad = ci.idciudad

    and c.idcliente = v.idcliente

    and v.idventa = vp.idventa

    and vp.idproducto = p.idproducto

    group by c.idcliente

)

select ciu.nombre ,c.nombre cliente,compra\_cliente.total total\_gastado

from compra\_cliente,cliente c,ciudad ciu

where compra\_cliente.cliente = c.idcliente

and ciu.idciudad = c.idciudad

and(compra\_cliente.cliente,compra\_cliente.total)

in

(compra\_cliente);

Tabla

Descripción generada automáticamente

4. Valor total recaudado por concepto de impuestos:

Escribe una consulta SQL para calcular el valor total recaudado por concepto de

impuestos. La salida debe mostrar:

* Total de impuestos

select sum((p.precio\*(cat.impuesto/100))\*vp.cantidad) total\_impuestos

from venta v,venta\_producto vp,producto p,categoria cat

where v.idventa = vp.idventa

and p.idproducto = vp.idproducto

and p.idcategoria = cat.idcategoria;

Tabla

Descripción generada automáticamente

5. Listado de precios actuales:

Escribe una consulta SQL para listar los precios actuales de todos los productos. La

salida debe incluir:

* Nombre del producto
* Precio

s|elect p.nombre producto,p.precio precio

from producto p ;

Tabla

Descripción generada automáticamente

6. Categoría más vendida en cada departamento (cantidad):

Escribe una consulta SQL para determinar cuál es la categoría más vendida (en

términos de cantidad) en cada departamento. La salida debe incluir:

* Nombre del departamento
* Nombre de la categoría
* Cantidad vendida
* with cant\_por\_cat as (
* select p.idcategoria cat ,d.iddepartamento dep,sum(vp.cantidad) cantidad
* from departamento d,ciudad ci,cliente c,venta v,venta\_producto vp,producto p ,categoria cat
* where d.iddepartamento = ci.iddepartamento
* and ci.idciudad = c.idciudad
* and c.idcliente = v.idcliente
* and v.idventa = vp.idventa
* and vp.idproducto = p.idproducto

    and p.idcategoria = cat.idcategoria

    group by p.idproducto,d.iddepartamento

    order by dep,p.idcategoria

)

select dep.nombre departamento,cat.nombre categoria, cant\_por\_cat.cantidad

from cant\_por\_cat,categoria cat,departamento dep

where cant\_por\_cat.cat = cat.idcategoria

and cant\_por\_cat.dep = dep.iddepartamento

and (cant\_por\_cat.dep,cant\_por\_cat.cantidad) in (

select cant\_por\_cat.dep,max(cant\_por\_cat.cantidad)

from cant\_por\_cat

group by cant\_por\_cat.dep

);

Tabla

Descripción generada automáticamente

7. Proyección de nuevos precios subiendo un 2.3% del precio actual de todos los

productos:

Escribe una consulta SQL para calcular los nuevos precios de todos los productos,

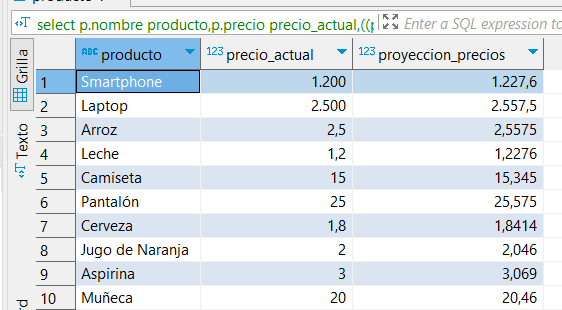
aplicando un incremento del 2.3% al precio actual. La salida debe incluir:

* Nombre del producto
* Precio actual
* Nuevo precio

select p.nombre producto,p.precio precio\_actual,((p.precio\*(2.3/100))+p.precio) proyeccion\_precios

from producto p,categoria c

where p.idcategoria = c.idcategoria ;



8. Cliente que más ha comprado en el último semestre $$$:

Escribe una consulta SQL para encontrar el cliente que más ha gastado en el último

semestre. La salida debe incluir:

* Nombre del cliente
* Total gastado

select c.nombre cliente,sum(p.precio\*vp.cantidad) total\_gastado

    from cliente c,venta v,venta\_producto vp,producto p

    where c.idcliente = v.idcliente

    and v.idventa = vp.idventa

    and vp.idproducto = p.idproducto

    and extract(month from v.fechaventa)>6

    group by c.idcliente,c.nombre

    having sum(p.precio\*vp.cantidad) = (select max(total\_semestre.total)

from (

    select c.nombre cliente,sum(p.precio\*vp.cantidad) total

    from cliente c,venta v,venta\_producto vp,producto p

    where c.idcliente = v.idcliente

    and v.idventa = vp.idventa

    and vp.idproducto = p.idproducto

    and extract(month from v.fechaventa)>6

    group by c.idcliente,c.nombre

)as total\_semestre);

Tabla

Descripción generada automáticamente

9. Cliente que menos ha comprado por cada categoría (cantidad):

Escribe una consulta SQL para determinar el cliente que menos ha comprado (en

términos de cantidad) por cada categoría. La salida debe incluir:

* Nombre de la categoría
* Nombre del cliente
* Cantidad comprada

with cantidad\_cat\_cliente as (

    select c.idcliente cliente,cat.idcategoria categoria,sum(vp.cantidad) cantidad

    from cliente c join venta v on c.idcliente = v.idcliente

    join venta\_producto vp on v.idventa = vp.idventa

    join producto p on vp.idproducto = p.idproducto

    join categoria cat on p.idcategoria = cat.idcategoria

    group by c.idcliente ,cat.idcategoria

    order by cat.idcategoria,c.idcliente

)

select cat.nombre categoria,c.nombrecliente,cantidad\_cat\_cliente.cantidad

from cantidad\_cat\_cliente,cliente c,categoria cat

where cantidad\_cat\_cliente.cliente = c.idcliente and cantidad\_cat\_cliente.categoria = cat.idcategoria

and (cantidad\_cat\_cliente.categoria,cantidad\_cat\_cliente.cantidad)

in

(select cantidad\_cat\_cliente.categoria categoria,min(cantidad\_cat\_cliente.cantidad) min\_cant

from cantidad\_cat\_cliente

group by cantidad\_cat\_cliente.categoria);

Tabla

Descripción generada automáticamente

10. Producto que menos dinero ha recaudado en cada uno de los departamentos:

Escribe una consulta SQL para identificar el producto que menos dinero ha

recaudado en cada departamento. La salida debe incluir:

* Nombre del departamento
* Nombre del producto
* Monto recaudado

with prod\_por\_dep as (

    select d.iddepartamento dep,p.idproducto prod,sum(vp.cantidad\*p.precio) cantidad

    from departamento d join ciudad ci on d.iddepartamento = ci.iddepartamento

    join cliente c on c.idciudad = ci.idciudad join venta v on v.idcliente = c.idcliente

    join venta\_producto vp on vp.idventa = v.idventa join producto p on p.idproducto = vp.idproducto

    group by d.iddepartamento,p.idproducto)

select d.nombre departamento, p.nombre producto, prod\_por\_dep.cantidad monto\_recaudado

from prod\_por\_dep join departamento d on prod\_por\_dep.dep=d.iddepartamento

join producto p on prod\_por\_dep.prod = p.idproducto

where(prod\_por\_dep.dep,prod\_por\_dep.cantidad)

in

(

    select prod\_por\_dep.dep,min(prod\_por\_dep.cantidad)

    from prod\_por\_dep

    group by prod\_por\_dep.dep

);

