

## Bases de datos II

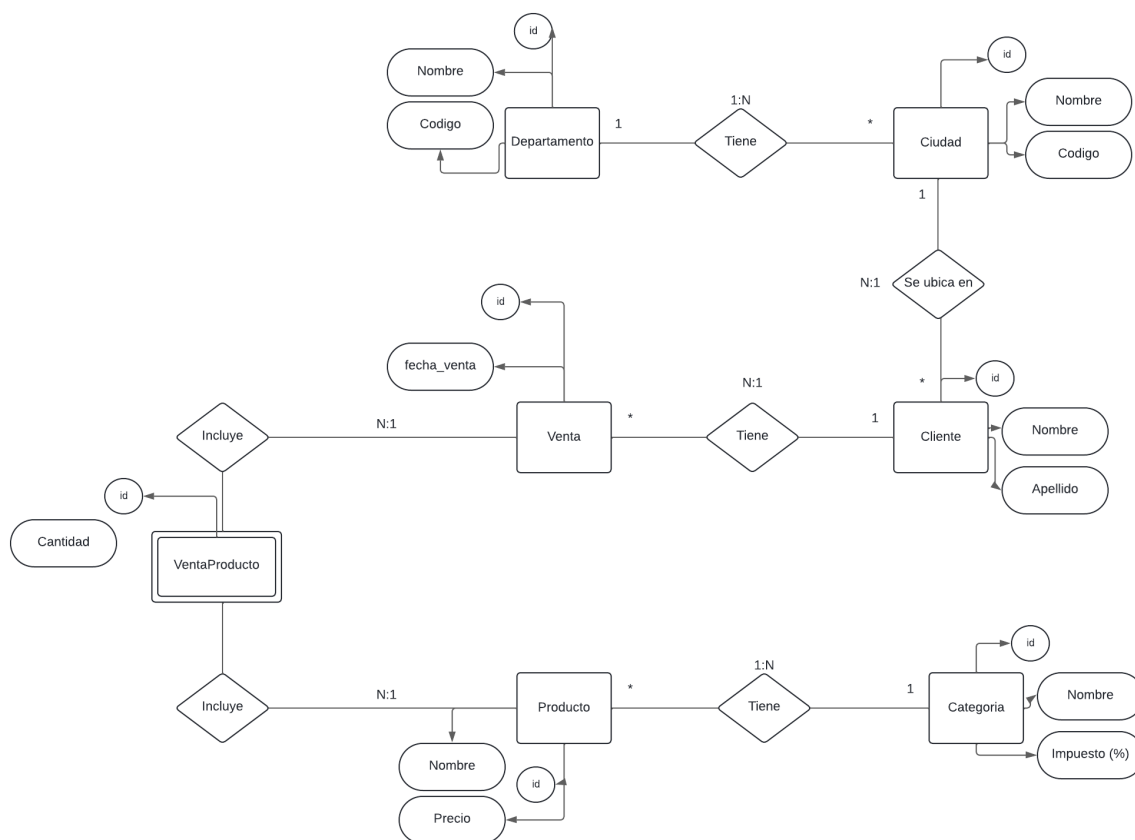
**Objetivo:** Resolver una serie de ejercicios utilizando SQL para trabajar con un modelo de base de datos de ventas. Las consultas deben reflejar la capacidad para manejar datos, realizar agregaciones, y aplicar funciones de agrupación y filtrado.

**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.

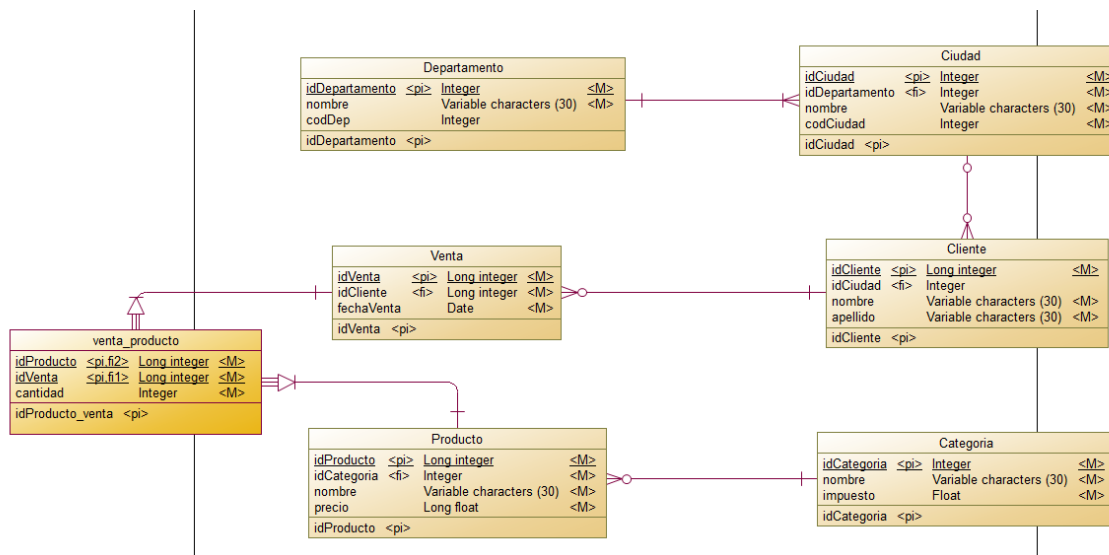
## Solución:

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

### - Modelo entidad- relación:



- **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
);
```

ciudad(+) 1 ×

select ciu.nombre ciudad,p.nombre producto,vp.cai Enter a

	ABC ciudad	ABC producto	123 cantidad
1	Medellín	Laptop	1
2	Medellín	Smartphone	1
3	Medellín	Cerveza	1
4	Medellín	Smartphone	1
5	Barranquilla	Arroz	1
6	Barranquilla	Arroz	1
7	Barranquilla	Smartphone	1
8	Barranquilla	Camiseta	1
9	Cartagena	Smartphone	1
10	Cartagena	Smartphone	1
11	Tunja	Camiseta	1
12	Tunja	Cerveza	1
13	Tunja	Camiseta	1
14	Tunja	Arroz	1
15	Bogotá	Smartphone	1

## 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
)
```

with product_por_mes as( select vp.idproducto prod				
	mes	producto	cantidad_vendida	
1	2024-01	Smartphone	1	
2	2024-02	Arroz	1	
3	2024-02	Cerveza	1	
4	2024-03	Smartphone	1	
5	2024-03	Arroz	1	
6	2024-04	Camiseta	1	
7	2024-05	Arroz	1	
8	2024-05	Camiseta	1	
9	2024-06	Cerveza	2	
10	2024-07	Smartphone	1	
11	2024-07	Camiseta	1	
12	2024-08	Laptop	1	
13	2024-08	Smartphone	1	
14	2024-08	Smartphone	1	
15	2024-09	Cerveza	2	
16	2024-10	Smartphone	1	
17	2024-11	Pantalón	2	
18	2024-12	Cerveza	1	

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);
```

with compra\_cliente as ( select c.idcliente cliente,sui | Enter a SQ

	ABC nombre	ABC cliente	123 total_gastado
1	Bogotá	Andrés	1.243,2
2	Tunja	Luisa	15.147,9
3	Barranquilla	Ana	8.869,2
4	Medellín	Juan	15.275
5	Cartagena	Carlos	17.855,4

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;
```

Resultados 1 x

select sum((p.precio\*(cat.impuesto/100))\*vp.c

	123 total_impuestos
1	10.998,621

5. Listado de precios actuales:

Escribe una consulta SQL para listar los precios actuales de todos los productos. La salida debe incluir:

Bases de datos II

- Nombre del producto
- Precio

```
select p.nombre producto,p.precio precio
from producto p ;
```

producto 1 ×			
select p.nombre producto,p.precio precio from pro			
Grilla	producto	precio	
1	Smartphone	1.200	
2	Laptop	2.500	
3	Arroz	2,5	
4	Leche	1,2	
5	Camiseta	15	
6	Pantalón	25	
7	Cerveza	1,8	
8	Jugo de Naranja	2	
9	Aspirina	3	
10	Muñeca	20	

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
);

```

departamento(+) 1 ×

with cant\_por\_cat as ( select p.idcategoria cat ,d.idc

	departamento	categoria	cantidad
1	Bolívar	Ropa	18
2	Caquetá	Alimentos	12
3	Córdoba	Alimentos	17
4	Guainía	Electrónica	6
5	Risaralda	Bebidas	24

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 ;

```

SQL Query: `select p.nombre producto,p.precio precio_actual,((p.precio*1.23)) as proyeccion_precios`

	ABC producto	123 precio_actual	123 proyeccion_precios
1	Smartphone	1.200	1.227,6
2	Laptop	2.500	2.557,5
3	Arroz	2,5	2,5575
4	Leche	1,2	1,2276
5	Camiseta	15	15,345
6	Pantalón	25	25,575
7	Cerveza	1,8	1,8414
8	Jugo de Naranja	2	2,046
9	Aspirina	3	3,069
10	Muñeca	20	20,46

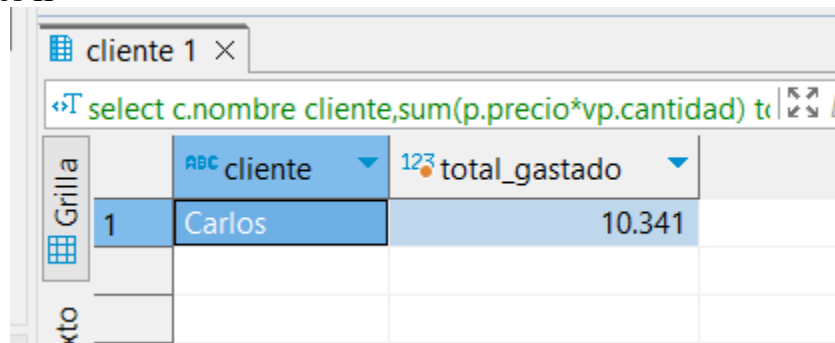
8. Cliente que más ha comprado en el último semestre \$\$\$:

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);
```





	cliente	total_gastado
1	Carlos	10.341

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);
```

with cantidad\_cat\_cliente as ( select c.idcliente clien

	Grilla	ABC categoria	ABC cliente	123 cantidad
1		Electrónica	Andrés	1
2		Alimentos	Luisa	5
3	Texto	Ropa	Luisa	6
4		Bebidas	Juan	1

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
);
```

Taller de repaso: Consultas SQL  
Brayan Alejandro Riveros Rodríguez  
20201020084  
Bases de datos II

departamento(+) 1 ×

with prod\_por\_dep as ( select d.iddepartamento de Enter a SQL expression to

	departamento	producto	monto_recaudado
1	Guainía	Cerveza	5,4
2	Risaralda	Cerveza	43,2
3	Caquetá	Cerveza	7,2
4	Córdoba	Arroz	10
5	Bolívar	Cerveza	1,8