

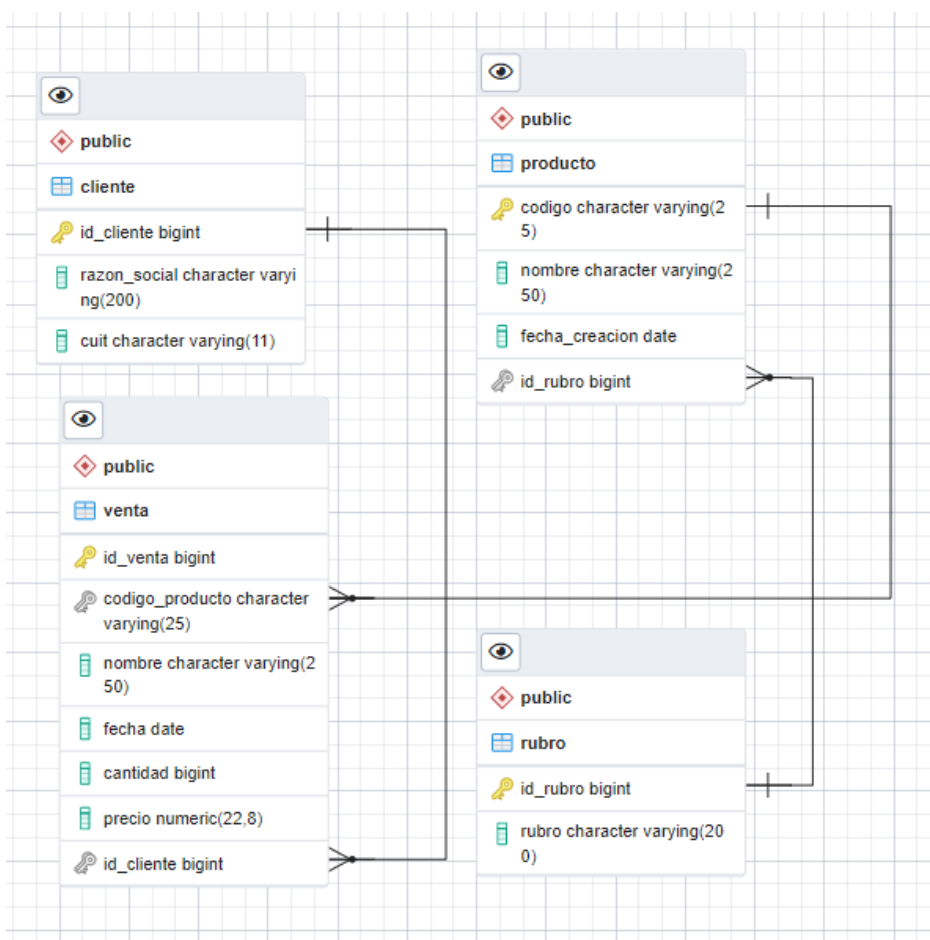
Parte 1 Prueba Técnica:

Consultas SQL

Tagliaferri Stefano Guillermo.

Realizar las consultas SQL necesarias para obtener los siguientes datos:

A modo de confirmación, se replicó la base de datos presentada en el enunciado en POSTGRESQL y se la cargo con algunos datos provisorios para realizar las consultas.



Cliente:

	id_cliente [PK] bigint	razon_social character varying (200)	cuit character varying (11)
1	1	Jose	11
2	2	Juan	22
3	3	Pedro	33
4	4	Carlos	44
5	5	Maria	55








Rubro:

	id_rubro [PK] bigint	rubro character varying (200)
1	1	libreria
2	2	bazar
3	3	cocina

Producto:

	codigo [PK] character varying (25)	nombre character varying (250)	fecha_creacion date	id_rubro bigint
1	1	plato	2021-12-20	2
2	2	tupper	2022-03-20	2
3	3	lapiz	2022-02-15	1
4	4	papel	2021-11-10	1
5	5	lapicera	2022-03-27	1
6	6	condimento	2022-03-28	3
7	7	cuaderno	2022-03-28	1

Venta:

Data Output		Explain	Messages	Notifications			
	 id_venta [PK] bigint	 codigo_producto character varying (25)	 nombre character varying (250)	 fecha date	 cantidad bigint	 precio numeric (22,8)	 id_cliente bigint
1	1	1	venta de plato	2022-03-25	10	200.00000000	1
2	2	1	venta de plato 2	2022-02-25	10	200.00000000	1
3	5	4	venta papel	2022-03-20	15	50.00000000	4
4	7	6	venta condimento	2021-12-20	15	150.00000000	5
5	3	2	venta tupper	2022-01-01	15	100.00000000	2
6	4	2	venta tupper 2	2022-01-01	15	100.00000000	2
7	6	2	venta tupper 3	2021-03-20	15	150.00000000	5

Enunciados:

- Todos los productos del rubro "libreria", creados hoy.

```
SELECT p.*  
FROM producto p JOIN rubro r ON p.id_rubro = r.id_rubro  
WHERE r.rubro = 'libreria' AND p.fecha_creacion = CURRENT_DATE
```

RESULTADO: (nótese que la fecha en la que se realizó la consulta fue el lunes 28/03):

	codigo [PK] character varying (25)	nombre character varying (250)	fecha_creacion date	id_rubro bigint
1	7	cuaderno	2022-03-28	1

- Monto total vendido por cliente (mostrar nombre del cliente y monto).

```
SELECT c.razon_social, SUM(v.precio) AS monto_total  
FROM cliente c JOIN venta v ON c.id_cliente = v.id_cliente  
GROUP BY c.id_cliente
```

RESULTADO:

	razon_social character varying (200)	monto_total numeric
1	Maria	300.00000000
2	Carlos	50.00000000
3	Juan	200.00000000
4	Jose	400.00000000

- Cantidad de ventas por producto.

// cantidad de ventas por producto. Contando las ventas.

```
SELECT p.nombre, COUNT(v.codigo_producto) as cantidad_de_ventas
FROM producto p JOIN venta v ON p.codigo = v.codigo_producto
GROUP BY p.codigo
```

RESULTADO:

	nombre character varying (250)	cantidad_de_ventas bigint
1	tupper	3
2	papel	1
3	condimento	1
4	plato	2

// cantidad de ventas por producto considerando la cantidad de productos en cada venta.

```
SELECT p.nombre, SUM(v.cantidad) as cantidad_de_ventas
FROM producto p JOIN venta v ON p.codigo = v.codigo_producto
GROUP BY p.codigo
```

RESULTADO:

	nombre character varying (250)	cantidad_de_ventas numeric
1	tupper	45
2	papel	15
3	condimento	15
4	plato	20

- Cantidad de productos comprados por cliente en el mes actual.

// comparo mes y año porque solo quiero mostrar las entradas del mes y el año actuales. No del mismo mes, pero de años anteriores. En caso de querer mostrar todas las entradas históricas para el mismo mes que el actual la segunda condición no iría.

```
SELECT c.razon_social, SUM(v.cantidad) as cantidad_productos_comprados
FROM cliente c JOIN venta v ON c.id_cliente = v.id_cliente
WHERE EXTRACT(MONTH from v.fecha) = EXTRACT(MONTH from CURRENT_DATE)
AND EXTRACT(YEAR from v.fecha) = EXTRACT(YEAR from CURRENT_DATE)
GROUP BY c.id_cliente
```

RESULTADO:

	razon_social character varying (200)	cantidad_productos_comprados numeric
1	Jose	10
2	Carlos	15

- Ventas que tienen al menos un producto del rubro "bazar".

// funciona bajo la condición de el atributo cantidad de la tabla venta no puede ser 0, Por lo tanto cada venta tiene al menos 1 producto.

```
SELECT v.*
FROM venta v join producto p ON v.codigo_producto = p.codigo
JOIN rubro r on r.id_rubro = p.id_rubro
WHERE r.rubro = 'bazar'
```

RESULTADO:

	id_venta [PK] bigint	codigo_producto character varying (25)	nombre character varying (250)	fecha date	cantidad bigint	precio numeric (22,8)	id_cliente bigint
1	1	1	venta de plato	2022-03-25	10	200.00000000	1
2	2	1	venta de plato 2	2022-02-25	10	200.00000000	1
3	3	2	venta tupper	2022-01-01	15	100.00000000	2
4	4	2	venta tupper 2	2022-01-01	15	100.00000000	2
5	6	2	venta tupper 3	2021-03-20	15	150.00000000	5

- Rubros que no tienen ventas en los ultimos 2 meses.

// revisa las entradas con fechas en los 2 últimos meses en formato días, por lo tanto se fija si hay ventas en los últimos ~(31*2) días.

// en primer lugar realizo una subconsulta que me devuelve todos los rubros que tengan ventas en los últimos 2 meses, luego en la consulta principal selecciono los rubros que no pertenecen a la primera tabla, por lo tanto son todos los que no tienen ventas en los últimos 2 meses.

```
SELECT r.*
FROM rubro r
WHERE r.id_rubro NOT IN (
    SELECT r2.id_rubro
    FROM venta v2 join producto p2 ON v2.codigo_producto = p2.codigo
    JOIN rubro r2 on r2.id_rubro = p2.id_rubro
    WHERE v2.fecha > CURRENT_DATE - INTERVAL '2 months'
    GROUP BY r2.id_rubro)
```

RESULTADO:

	id_rubro [PK] bigint	rubro character varying (200)
1	3	cocina

RESULTADO SUBCONSULTA:

	id_rubro [PK] bigint
1	1
2	2