



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Examen 3

PROFESOR

Gerardo Avilés Rosas

AYUDANTES

Gerardo Uriel Soto Miranda
Valeria Fernanda Manjarrez Angeles
Ricardo Badillo Macías
Jerónimo Almeida Rodríguez

ALUMNOS:

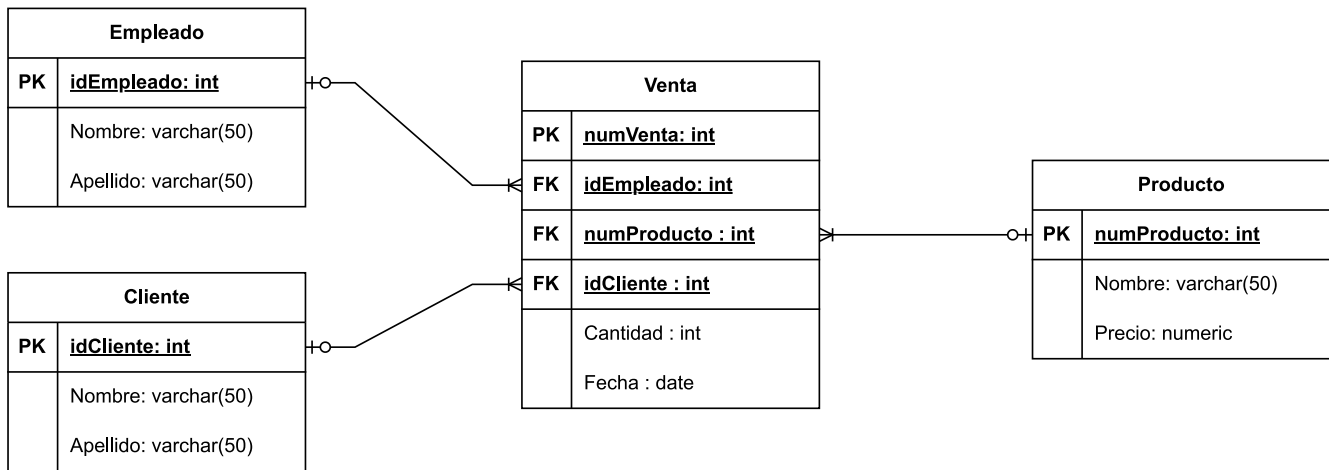
Castañon Maldonado Carlos Emilio
Navarro Santana Pablo César
Nepomuceno Escarcega Arizdelcy Lizbeth

ASIGNATURA

Fundamentos de Bases de Datos

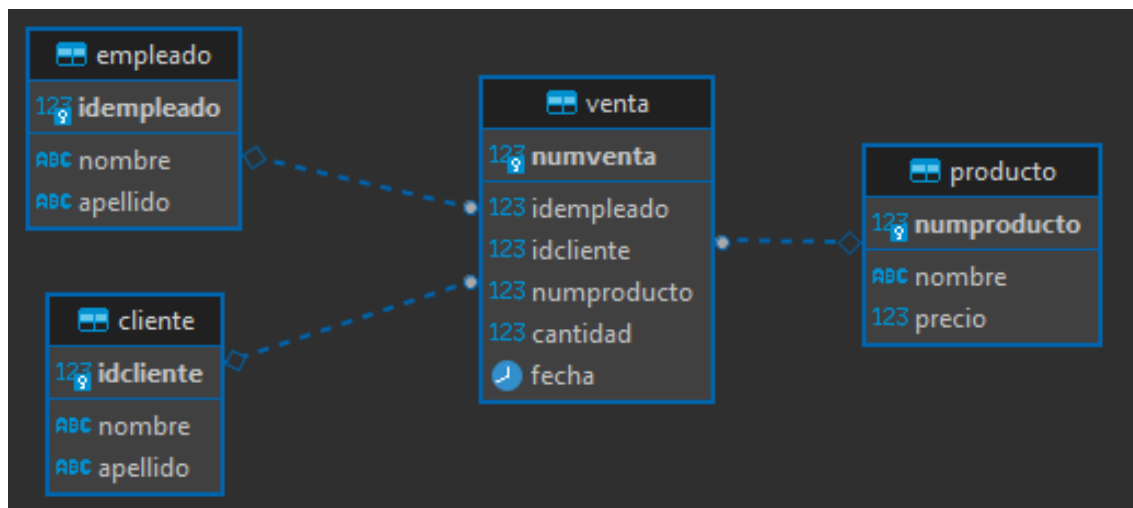
4 (45 puntos) Consultas a una base de datos

Se tiene el siguiente esquema de bases de datos acerca de **Empleados**, **Cientes**, **Productos** que se ofertan y **Ventas** que se realizan:



(6 puntos) Implementa el esquema anterior en PostgreSQL, utilizando para este proceso el archivo en BACKUP que acompaña a esta evaluación.

Nota: Las tablas que deberán quedar son: **Empleado**, **Producto**, **Cliente** y **Venta**.



Crea un archivo con **extensión SQL** y ahí guarda la **consulta** que da solución a cada una de las **solicitudes de información** que se listan a continuación.

(3 puntos por cada consulta):

- a) Obtener toda la información de los clientes que hayan realizado una compra con los vendedores 5 u 8 en 2015.

```
SELECT DISTINCT Cliente.*
FROM Venta
INNER JOIN Cliente ON Venta.idCliente = Cliente.idCliente
INNER JOIN Empleado ON Venta.idEmpleado = Empleado.idEmpleado
WHERE (Empleado.idEmpleado = 5 OR Empleado.idEmpleado = 8)
AND EXTRACT(YEAR FROM Venta.Fecha) = 2015;
```

cliente 1 ×				
SELECT DISTINCT Cliente.* FROM Venta INNER JOIN Cliente ON Venta.idCliente = Cliente.idCliente INNER JOIN Empleado ON Venta.idEmpleado = Empleado.idEmpleado WHERE (Empleado.idEmpleado = 5 OR Empleado.idEmpleado = 8) AND EXTRACT(YEAR FROM Venta.Fecha) = 2015;				
	idcliente	nombre	apellido	
1	905	Allison	Allen	
2	1.964	Austin	Jai	
3	4.446	Connor	Edwards	
4	4.942	Danielle	Sanders	
5	6.748	Evelyn	Perez	
6	7.700	Hailey	Coleman	
7	8.203	Isabel	Long	
8	9.765	Jésus	Munoz	
9	10.856	Kaitlyn	Harris	
10	12.300	Lauren	Cooper	

- b) Obtener toda la información de órdenes de venta que se hayan realizado en 2013 o 2014 durante los trimestres impares y cuya cantidad se encuentre entre 100 y 150 unidades.

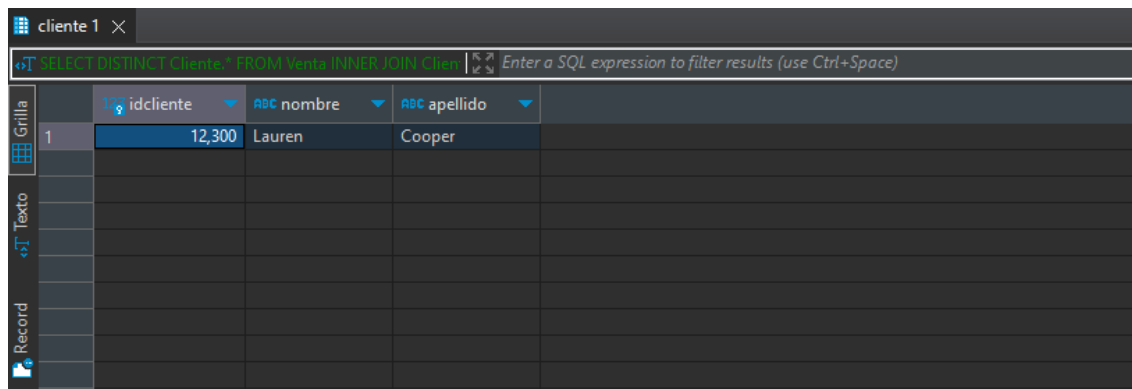
```
SELECT Venta.*, Producto.*, Empleado.*
FROM Venta
INNER JOIN Producto ON Venta.numProducto = Producto.numProducto
INNER JOIN Empleado ON Venta.idEmpleado = Empleado.idEmpleado
WHERE (EXTRACT(YEAR FROM Venta.Fecha) = 2013 OR EXTRACT(YEAR FROM Venta.Fecha) = 2014)
AND (EXTRACT(QUARTER FROM Venta.Fecha) IN (1, 3))
AND (Venta.Cantidad BETWEEN 100 AND 150);
```

venta(+) 1 ×									
SELECT Venta.*, Producto.*, Empleado.* FROM Venta INNER JOIN Producto ON Venta.numProducto = Producto.numProducto INNER JOIN Empleado ON Venta.idEmpleado = Empleado.idEmpleado WHERE (EXTRACT(YEAR FROM Venta.Fecha) = 2013 OR EXTRACT(YEAR FROM Venta.Fecha) = 2014) AND (EXTRACT(QUARTER FROM Venta.Fecha) IN (1, 3)) AND (Venta.Cantidad BETWEEN 100 AND 150);									
	numventa	idempleado	idcliente	numproducto	cantidad	fecha	numproducto	apellido	
1	274	15	10,700	107	107	2013-07-08	107	Thin-	
2	313	2	14,000	140	140	2014-02-12	140	Lock	
3	329	2	14,000	140	140	2013-09-22	140	Lock	
4	567	2	14,000	140	140	2014-02-08	140	Lock	
5	694	8	12,300	123	123	2013-09-19	123	Lock	
6	695	2	14,000	140	140	2014-01-10	140	Lock	
7	822	8	12,300	123	123	2013-09-08	123	Lock	
8	1,003	15	10,700	107	107	2013-07-25	107	Thin-	
9	1,067	15	10,700	107	107	2014-02-12	107	Thin-	

Comentario: Decidimos poner toda la información relacionada con la venta, incluyendo los datos de la tabla Producto y Empleado, esto debido a que nos pide "Toda" la información de las ordenes.

- c) Obtener toda la información de los clientes cuyo apellido contenga el patrón de cadena “co” (p.e. Collins) y que hayan solicitado productos con precio entre \$150 y \$180.

```
SELECT DISTINCT Cliente.*
FROM Venta
INNER JOIN Cliente ON Venta.idCliente = Cliente.idCliente
INNER JOIN Producto ON Venta.numProducto = Producto.numProducto
WHERE Cliente.Apellido ILIKE '%co%'
AND Producto.Precio BETWEEN 150 AND 180;
```

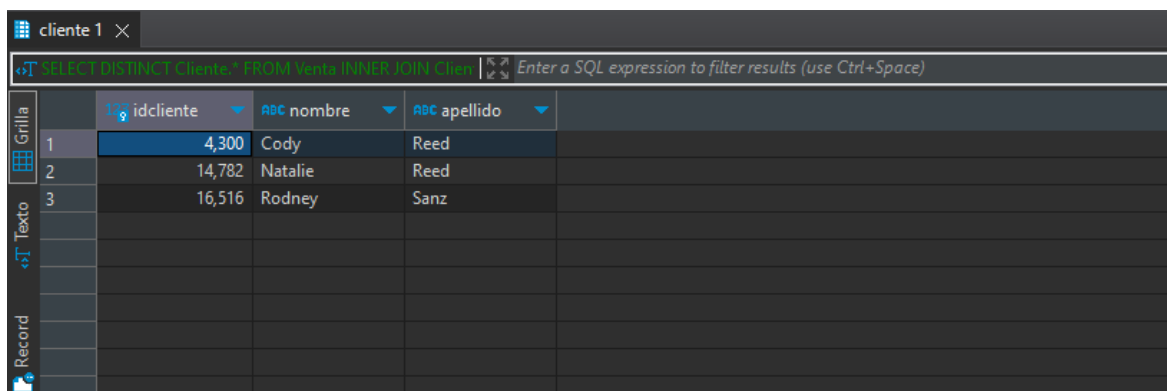


	idcliente	nombre	apellido
1	12,300	Lauren	Cooper

- d) Obtener toda la información de los clientes que tengan un apellido de exactamente cuatro caracteres, en donde la primera letra sea “C” o “R” o “S”, que hayan comprado productos que tengan el patrón de cadena “nut” y que hayan sido atendidos por empleados con identificadores pares.

Hint: apóyate en la función módulo, la cual devuelve el residuo de la división, $mod(dividendo, divisor)$.

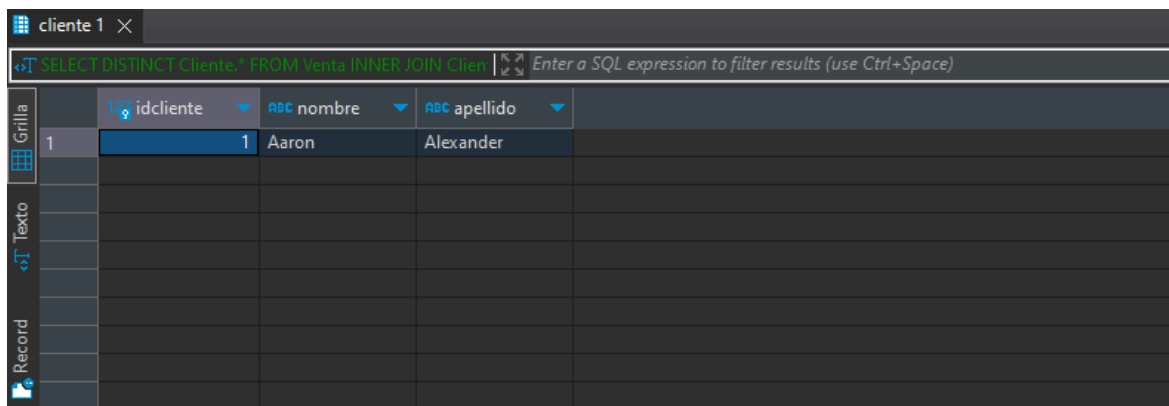
```
SELECT DISTINCT Cliente.*
FROM Venta
INNER JOIN Cliente ON Venta.idCliente = Cliente.idCliente
INNER JOIN Producto ON Venta.numProducto = Producto.numProducto
INNER JOIN Empleado ON Venta.idEmpleado = Empleado.idEmpleado
WHERE CHAR_LENGTH(Cliente.Apellido) = 4
AND LEFT(Cliente.Apellido, 1) IN ('C', 'R', 'S')
AND Producto.Nombre ILIKE '%nut%'
AND Empleado.idEmpleado % 2 = 0;
```



	idcliente	nombre	apellido
1	4,300	Cody	Reed
2	14,782	Natalie	Reed
3	16,516	Rodney	Sanz

- e) Obtener toda la información de los clientes que se llaman “Aaron” y que hayan realizado alguna compra durante los meses pares del año 2016.
Hint: apóyate en la función MOD, es posible que requieras realizar una conversión explícita en *date_part* así que puedes utilizar *CAST(valor as NuevoTipo)*.

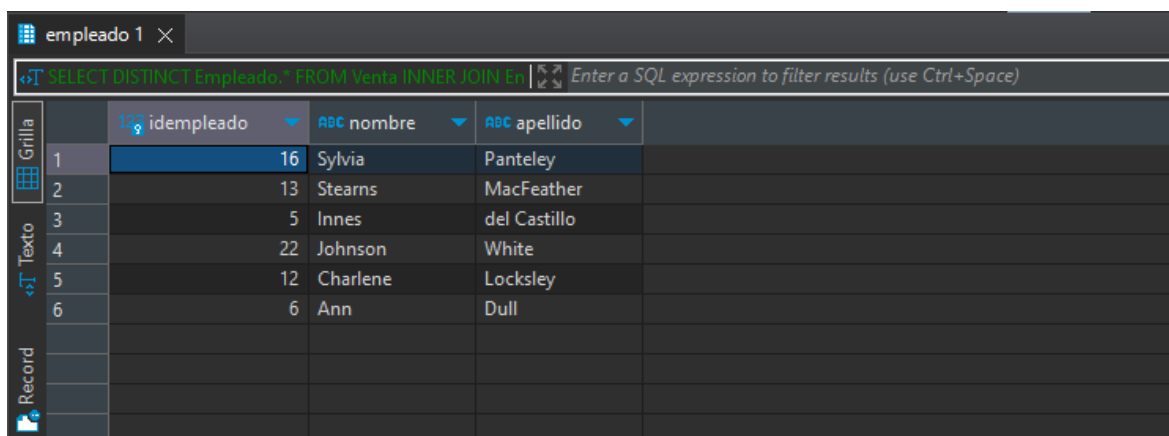
```
SELECT DISTINCT Cliente.*
FROM Venta
INNER JOIN Cliente ON Venta.idCliente = Cliente.idCliente
WHERE Cliente.Nombre = 'Aaron'
      AND EXTRACT(MONTH FROM CAST(Venta.Fecha AS DATE)) % 2 = 0
      AND EXTRACT(YEAR FROM CAST(Venta.Fecha AS DATE)) = 2016;
```



	idcliente	nombre	apellido
1	1	Aaron	Alexander

- f) Información de los empleados que hayan vendido productos con números entre 300 y 400, cuya cantidad solicitada esté por arriba de 800 unidades.

```
SELECT DISTINCT Empleado.*
FROM Venta
INNER JOIN Empleado ON Venta.idEmpleado = Empleado.idEmpleado
WHERE Venta.numProducto BETWEEN 300 AND 400
      AND Venta.Cantidad > 800;
```

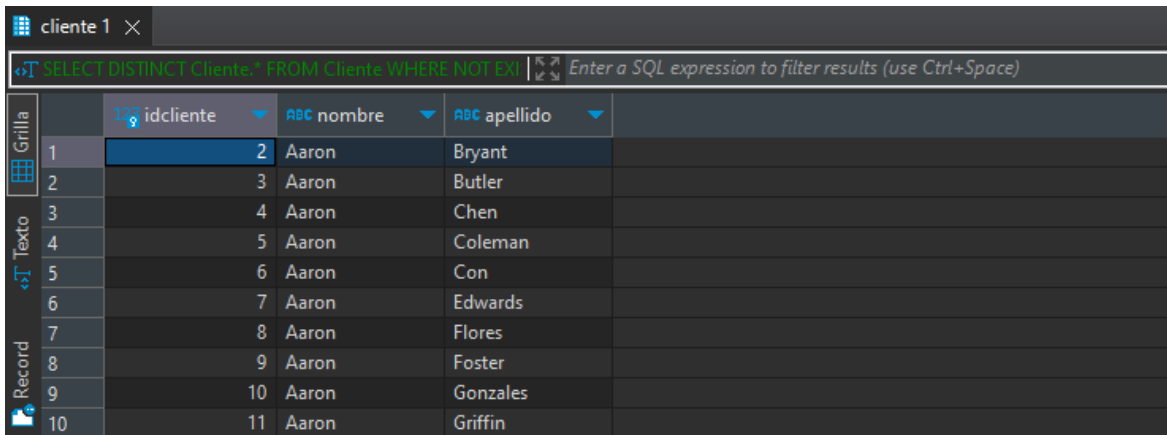


	idempleado	nombre	apellido
1	16	Sylvia	Panteley
2	13	Stearns	MacFeather
3	5	Innes	del Castillo
4	22	Johnson	White
5	12	Charlene	Locksley
6	6	Ann	Dull

Comentario: Nosotros entendimos la parte de “cuya cantidad solicitada esté por arriba de 800 unidades” como, que en una venta, el empleado vendió alrededor de 800 unidades de ese producto.

- g) Obtener toda la información de los clientes que no han efectuado ninguna compra en la tienda.

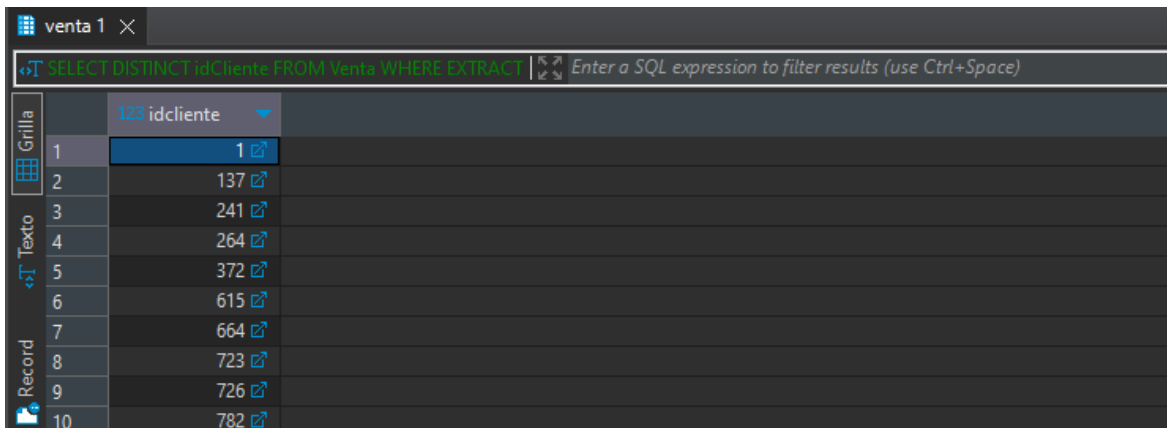
```
SELECT DISTINCT Cliente.*
FROM Cliente
WHERE NOT EXISTS (
    SELECT 1
    FROM Venta
    WHERE Venta.idCliente = Cliente.idCliente
);
```



	idcliente	nombre	apellido
1	2	Aaron	Bryant
2	3	Aaron	Butler
3	4	Aaron	Chen
4	5	Aaron	Coleman
5	6	Aaron	Con
6	7	Aaron	Edwards
7	8	Aaron	Flores
8	9	Aaron	Foster
9	10	Aaron	Gonzales
10	11	Aaron	Griffin

- h) Obtener el identificador de los clientes que hayan comprado algún producto en 2013, pero que no hayan adquirido el producto con identificador 500.

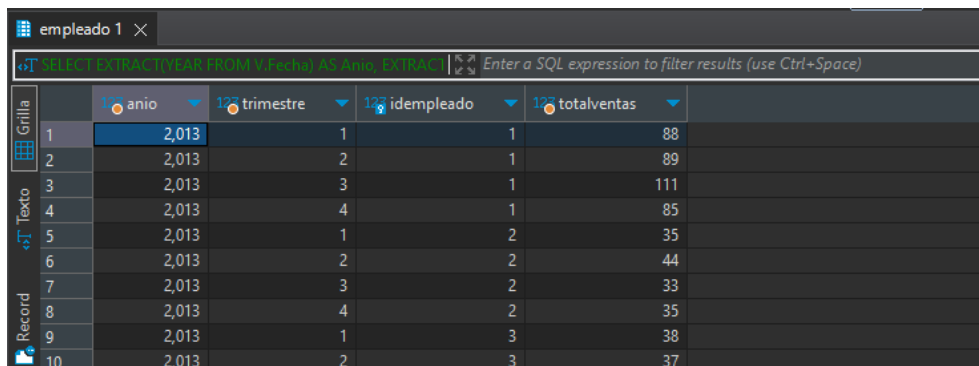
```
SELECT DISTINCT idCliente
FROM Venta
WHERE EXTRACT(YEAR FROM Fecha) = 2013
AND idCliente NOT IN (
    SELECT idCliente
    FROM Venta
    WHERE numProducto = 500
);
```



	idcliente
1	1
2	137
3	241
4	264
5	372
6	615
7	664
8	723
9	726
10	782

- i) Información del total de ventas hechas por cada vendedor en cada año y trimestre.
Mostrar la información ordenada por año, en el caso del vendedor mostrar solo su ID.

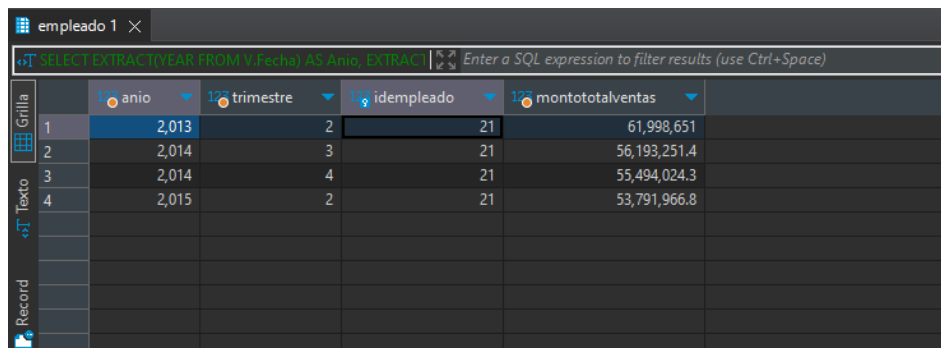
```
SELECT
    EXTRACT(YEAR FROM V.Fecha) AS Anio,
    EXTRACT(QUARTER FROM V.Fecha) AS Trimestre,
    E.idEmpleado,
    COUNT(*) AS TotalVentas
FROM Venta V
INNER JOIN Empleado E ON V.idEmpleado = E.idEmpleado
GROUP BY Anio, Trimestre, E.idEmpleado
ORDER BY Anio, E.idEmpleado, Trimestre;
```



	anio	trimestre	idempleado	totalventas
1	2,013	1	1	88
2	2,013	2	1	89
3	2,013	3	1	111
4	2,013	4	1	85
5	2,013	1	2	35
6	2,013	2	2	44
7	2,013	3	2	33
8	2,013	4	2	35
9	2,013	1	3	38
10	2,013	2	3	37

- j) Misma información que el inciso anterior, pero solo interesan las ventas que estén por arriba de \$50,000,000.00.

```
SELECT
    EXTRACT(YEAR FROM V.Fecha) AS Anio,
    EXTRACT(QUARTER FROM V.Fecha) AS Trimestre,
    E.idEmpleado,
    SUM(V.Cantidad * P.Precio) AS MontoTotalVentas
FROM Venta V
INNER JOIN Empleado E ON V.idEmpleado = E.idEmpleado
INNER JOIN Producto P ON V.numProducto = P.numProducto
GROUP BY Anio, Trimestre, E.idEmpleado
HAVING SUM(V.Cantidad * P.Precio) > 50000000.00
ORDER BY Anio, E.idEmpleado, Trimestre;
```



	anio	trimestre	idempleado	montototalventas
1	2,013	2	21	61,998,651
2	2,014	3	21	56,193,251.4
3	2,014	4	21	55,494,024.3
4	2,015	2	21	53,791,966.8



- k) Mostrar la información de la mayor compra, la menor compra, la compra promedio y el conteo de compras realizadas en la tienda por año y trimestre.
Mostrar la información ordenada.

```
SELECT
  EXTRACT(YEAR FROM Fecha) AS Año,
  EXTRACT(QUARTER FROM Fecha) AS Trimestre,
  MAX(Cantidad) AS Mayor_Compra,
  MIN(Cantidad) AS Menor_Compra,
  AVG(Cantidad) AS Compra_Promedio,
  COUNT(*) AS Conteo_Compras
FROM Venta
GROUP BY Año, Trimestre
ORDER BY Año, Trimestre;
```

Resultados 1							
Enter a SQL expression to filter results (use Ctrl+Space)							
	123 año	123 trimestre	123 mayor_compra	123 menor_compra	123 compra_promedio	123 conteo_compras	
1	2,013	1	997	11	479.0769230769	845	
2	2,013	2	997	11	500.3984018265	876	
3	2,013	3	983	11	490.9468208092	865	
4	2,013	4	997	11	477.2081395349	860	
5	2,014	1	997	11	494.0401891253	846	
6	2,014	2	983	11	498.2335483871	775	
7	2,014	3	983	11	500.9166666667	828	
8	2,014	4	983	11	493.8438202247	890	
9	2,015	1	997	11	482.7682038835	824	
10	2,015	2	997	11	502.4939759036	913	

- l) Mostrar por cada cliente y número de compra, la cantidad de productos que se solicitaron en la compra y el monto total pagado.

```
SELECT
  Cliente.idCliente,
  Venta.numVenta,
  SUM(Venta.Cantidad) AS CantidadTotal,
  SUM(Producto.Precio * Venta.Cantidad) AS MontoTotalPagado
FROM Venta
INNER JOIN Cliente ON Venta.idCliente = Cliente.idCliente
INNER JOIN Producto ON Venta.numProducto = Producto.numProducto
GROUP BY Cliente.idCliente, Venta.numVenta;
```

cliente(+) 1					
Enter a SQL expression to filter results (use Ctrl+Space)					
	123 idcliente	123 numventa	123 cantidadtotal	123 montototalpagado	
1	15,182	6,427	547	39,821.6	
2	13,845	8,053	903	109,678.38	
3	3,374	11,441	280	944,997.2	
4	10,366	11,998	250	337,400	
5	2,425	12,226	840	292,958.4	
6	11,264	7,864	903	109,678.38	
7	18,723	2,126	780	2,651,992.2	
8	16,881	8,933	827	70,762.255	
9	14,664	5,966	937	23,415.63	
10	2,302	10,328	437	10,920.63	



- m) Incrementa el precio de los productos en un 15 %.

```
update producto
set precio = (precio * 0.15) + precio
returning *;
```

producto 1 ×					
select * from producto Enter a SQL expression to filter results (use Ctrl+Space)					
Grilla		123 numproducto	ABC nombre	123 precio	
	1	79	LL Hub	736	
Texto	2	1	Adjustable Race	1,84	
	3	2	Bearing Ball	0,92	
	4	3	BB Ball Bearing	2,76	
	5	4	Headset Ball Bearings	0	
	6	5	Blade	218,04	
	7	6	LL Crankarm	437,46	
	8	7	ML Crankarm	219,42	
	9	8	HL Crankarm	366,85	
	10	9	Chainring Bolts	0	
	11	10	Chainring Nut	221,49	
Record	12	11	Chainring	222,18	
	13	12	Crown Race	594,32	
	14	13	Chain Stays	74,52	