



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

Laboratorio de Bases de Datos

Practica 10

Número de Cuenta: 422023645

Profesor: Cinthia Rodríguez Maya

Alumno: Hernández Diaz Sebastian

Semestre: 2024-2

ACTIVIDAD 1

Tomando en cuenta la base de datos vista en Laboratorio similar a Amazon

1. Crea la tabla statusCompras, esta tabla tendrá el catálogo de los status de las compras, los cuales pueden ser: Pagado y en proceso de envío, Entregado, Perdido y reembolsado, Cancelado por el cliente, Cancelado por el vendedor. El status actualmente es una columna en la tabla de compra, haz que sea un id de tipo llave foránea a ese catálogo. Puedes agregar los status de forma aleatoria.

Creación de la tabla statusCompras y la agregación de los status.

```
baseG1_422023645⇒ create table amazon.statusCompras(id serial CONSTRAINT pk_idstatus PRIMARY KEY, descripcion VARCHAR(255) not null);
CREATE TABLE
baseG1_422023645⇒ INSERT INTO amazon.statusCompras (descripcion) VALUES
('Pagado y en proceso de envío'),
('Entregado'),
('Perdido y reembolsado'),
('Cancelado por el cliente'),
('Cancelado por el vendedor');
INSERT 0 5
baseG1_422023645⇒ SELECT * FROM amazon.statusCompras;
id |      descripcion
---+-----
 1 | Pagado y en proceso de envío
 2 | Entregado
 3 | Perdido y reembolsado
 4 | Cancelado por el cliente
 5 | Cancelado por el vendedor
(5 rows)
```

Se cambia el tipo de dato en la columna status de amazon.compra para aceptar la llave foránea.

```
baseG1_422023645⇒ ALTER TABLE amazon.compra
ALTER COLUMN status TYPE INTEGER USING status::INTEGER;
ALTER TABLE
```

```
baseG1_422023645⇒ ALTER TABLE amazon.compra
ALTER COLUMN status SET NOT NULL;
ALTER TABLE
```

Se altera la tabla para que acepte las llaves foráneas desde la tabla de statusCompas.

```
baseG1_422023645⇒ ALTER TABLE amazon.compra ADD CONSTRAINT  
fk_statusCompras FOREIGN KEY (status)  
REFERENCES amazon.statusCompras(id)  
on delete cascade;
```

Se actualizan los datos en status:

```
baseG1_422023645⇒ UPDATE amazon.compra SET status = 2 WHERE id=1;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 1 WHERE id=2;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 5 WHERE id=3;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 3 WHERE id=4;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 5 WHERE id=5;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 4 WHERE id=6;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 2 WHERE id=7;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 3 WHERE id=8;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 1 WHERE id=9;  
UPDATE 1  
baseG1_422023645⇒ UPDATE amazon.compra SET status = 2 WHERE id=10;  
UPDATE 1
```

2. Responde las siguientes preguntas:

a. ¿Cuál es la cantidad de pedidos por cada status?

```
baseG1_422023645⇒ SELECT statusCompras.descripcion, COUNT(amazon.compra.status) AS cantidad_pedidos  
FROM amazon.statusCompras  
LEFT JOIN amazon.compra ON statusCompras.id = amazon.compra.status  
GROUP BY statusCompras.descripcion;
```

descripcion	cantidad_pedidos
Entregado	3
Perdido y reembolsado	2
Pagado y en proceso de envío	2
Cancelado por el vendedor	2
Cancelado por el cliente	1

(5 rows)

b. ¿Cuánto dinero ha gastado Carlos de la Facultad de Ingeniería en compras?

Primeramente, se muestran todas las compras de Carlos.

```
baseG1_422023645⇒ SELECT * FROM amazon.compra JOIN amazon.vendedorproducto
ON amazon.compra.idproducto=amazon.vendedorproducto.id
WHERE amazon.compra.idcliente LIKE 'carlos%';
```

id	idcliente	idproducto	fechacompra	status	id	idproducto	idvendedor	precio	unidades
1	carlos@fi.unam.mx	3	2024-01-10	2	3	HAJSAJ	GHAH55	23000	7
2	carlos@fi.unam.mx	3	2024-01-05	1	3	HAJSAJ	GHAH55	23000	7
3	carlos@fi.unam.mx	8	2024-02-05	5	8	QIIUU	JDSK5353GG	16000	2
4	carlos@fi.unam.mx	10	2024-04-08	3	10	JJJ999JJ	UWYUE77	39.5	23
5	carlos@fi.unam.mx	1	2024-04-01	5	1	YUEYWU	GHAH55	7000	4
6	carlos@fi.unam.mx	9	2023-11-22	4	9	JJJ999JJ	JDSK5353GG	39.56	45

(6 rows)

Despues todas estas se suman pero como podemos ver con anterioridad hay distintos status en las compras.

```
baseG1_422023645⇒ SELECT SUM(vendedorproducto.precio) AS total_gastado
FROM amazon.compra
JOIN amazon.cliente ON amazon.compra.idcliente = amazon.cliente.correo
JOIN amazon.vendedorproducto ON amazon.compra.idproducto = amazon.vendedorproducto.id
WHERE amazon.cliente.nombre = 'Carlos';
```

total_gastado
69079.06

(1 row)

Por lo que si solo queremos los productos que ya recibió el status debe de ser 2 ya que es el id del status de entregado, obteniendo así que ha gastado 23000.

```
baseG1_422023645⇒ SELECT SUM(vendedorproducto.precio) AS total_gastado
FROM amazon.compra
JOIN amazon.cliente ON amazon.compra.idcliente = amazon.cliente.correo
JOIN amazon.vendedorproducto ON amazon.compra.idproducto = amazon.vendedorproducto.id
WHERE amazon.cliente.nombre = 'Carlos' AND amazon.compra.status=2;
```

total_gastado
23000

(1 row)

Por otro lado, si se quiere tomar en cuenta también los que están en proceso de envío daría como resultado 46000, ya que los demás o fueron cancelados o perdidos y reembolsados.

```
baseG1_422023645⇒ SELECT SUM(vendedorproducto.precio) AS total_gastado
FROM amazon.compra
JOIN amazon.cliente ON amazon.compra.idcliente = amazon.cliente.correo
JOIN amazon.vendedorproducto ON amazon.compra.idproducto = amazon.vendedorproducto.id
WHERE amazon.cliente.nombre = 'Carlos' AND (amazon.compra.status=2 OR amazon.compra.status=1);
```

total_gastado
46000

(1 row)

c. Agrega 10 compras para Hugo, los productos que tú desees en el intervalo de tiempo de 2020 a 2024

```
baseG1_422023645⇒ INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','3','2020-01-10',1);
INSERT 0 1
baseG1_422023645⇒ INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','2','2022-01-5',2);

INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','8','2024-02-5',2);
INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','10','2021-04-08',3);
INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','9','2020-04-01',4);
INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','2','2020-11-22',1);
INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','1','2024-02-22',4);
INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','6','2022-02-22',5);
INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','7','2023-01-19',2);
INSERT INTO amazon.compra(idcliente, idproducto, fechacompra, status) VALUES ('mapamundi@hotmail.com','6','2021-01-19',2);
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

d. Imprime un reporte de la cantidad de compras por año, ordenadas del año más reciente al más antiguo

Primero obtenemos todas las compras dado que son pocas podemos contarlas manualmente para verificar el resultado.

```
baseG1_422023645⇒ SELECT * FROM amazon.compra;
id | idcliente | idproducto | fechacompra | status
---+---+---+---+---
1 | carlos@fi.unam.mx | 3 | 2024-01-10 | 2
2 | carlos@fi.unam.mx | 3 | 2024-01-05 | 1
3 | carlos@fi.unam.mx | 8 | 2024-02-05 | 5
4 | carlos@fi.unam.mx | 10 | 2024-04-08 | 3
5 | carlos@fi.unam.mx | 1 | 2024-04-01 | 5
6 | carlos@fi.unam.mx | 9 | 2023-11-22 | 4
7 | delfines@gmail.com | 10 | 2024-02-22 | 2
8 | delfines@gmail.com | 1 | 2024-02-22 | 3
9 | delfines@gmail.com | 4 | 2024-01-19 | 1
10 | delfines@gmail.com | 8 | 2022-01-19 | 2
11 | mapamundi@hotmail.com | 3 | 2020-01-10 | 1
12 | mapamundi@hotmail.com | 2 | 2022-01-05 | 2
13 | mapamundi@hotmail.com | 8 | 2024-02-05 | 2
14 | mapamundi@hotmail.com | 10 | 2021-04-08 | 3
15 | mapamundi@hotmail.com | 9 | 2020-04-01 | 4
16 | mapamundi@hotmail.com | 2 | 2020-11-22 | 1
17 | mapamundi@hotmail.com | 1 | 2024-02-22 | 4
18 | mapamundi@hotmail.com | 6 | 2022-02-22 | 5
19 | mapamundi@hotmail.com | 7 | 2023-01-19 | 2
20 | mapamundi@hotmail.com | 6 | 2021-01-19 | 2
(20 rows)
```

Después observamos el resultado de la consulta y al comparar podemos ver que es correcto.

```
baseG1_422023645⇒ SELECT EXTRACT(YEAR FROM fechacompra) AS ano, COUNT(*) AS cantidad_compras
FROM amazon.compra
GROUP BY ano
ORDER BY ano DESC;
ano | cantidad_compras
-----+-----
2024 | 10
2023 | 2
2022 | 3
2021 | 2
2020 | 3
(5 rows)
```

e. ¿Qué vendedor tiene a la venta más productos?

Primero podemos ver a todos los vendedores y la cantidad de productos que tienen.

```
baseG1_422023645⇒ SELECT amazon.vendedor.nombre, COUNT(*) AS cantidad_productos
FROM amazon.vendedor
JOIN amazon.vendedorproducto ON amazon.vendedor.rfc = amazon.vendedorproducto.idvendedor
GROUP BY amazon.vendedor.nombre
ORDER BY cantidad_productos DESC;
nombre | cantidad_productos
-----+-----
El baúl de cachivaches Store | 5
Sony tienda original | 3
Lupita independiente | 2
(3 rows)
```

Para después seleccionar al de mayor cantidad.

```
baseG1_422023645⇒ SELECT amazon.vendedor.nombre, COUNT(*) AS cantidad_productos
FROM amazon.vendedor
JOIN amazon.vendedorproducto ON amazon.vendedor.rfc = amazon.vendedorproducto.idvendedor
GROUP BY amazon.vendedor.nombre
ORDER BY cantidad_productos DESC
LIMIT 1;
nombre | cantidad_productos
-----+-----
El baúl de cachivaches Store | 5
(1 row)
```

f. Agrega el producto Sofá Laurita al inventario de todos los vendedores, pero con diferentes precios para cada uno, mediante otra query responde ¿Quién lo da más caro?

Lo primero a hacer es crear el producto y agregarlo a la lista de vendedorproducto.

```
baseG1_422023645⇒ INSERT INTO amazon.producto(sku, nombre, idtipo, descripcion)
VALUES ('ABCDE', 'Sofá Laurita', 3, 'Sofa de piel marca LAURITA');
INSERT 0 1
baseG1_422023645⇒ INSERT INTO amazon.vendedorproducto(idproducto, idvendedor, precio, unidades) VALUES ('ABCDE', 'GHAH55', 6500, 2);
INSERT 0 1
baseG1_422023645⇒ INSERT INTO amazon.vendedorproducto(idproducto, idvendedor, precio, unidades) VALUES ('ABCDE', 'JDSK5353GG', 6000, 6);
INSERT 0 1
baseG1_422023645⇒ INSERT INTO amazon.vendedorproducto(idproducto, idvendedor, precio, unidades) VALUES ('ABCDE', 'UWYUE77', 7500, 10);
INSERT 0 1
```

De esta forma podemos realizar la búsqueda y observamos que el resultado es correcto.

```
baseG1_422023645⇒ SELECT nombre, MAX(precio) AS precio_mas_caro
FROM amazon.vendedor
JOIN amazon.vendedorproducto ON amazon.vendedor.rfc = amazon.vendedorproducto.idvendedor
WHERE amazon.vendedorproducto.idproducto = 'ABCDE'
GROUP BY nombre
ORDER BY precio_mas_caro DESC
LIMIT 1;
  nombre | precio_mas_caro
-----+-----
Lupita independiente |          7500
(1 row)
```

Para confirmar obtenemos los precios por cada vendedor.

```
baseG1_422023645⇒ SELECT nombre, MAX(precio) AS precio_mas_caro
FROM amazon.vendedor
JOIN amazon.vendedorproducto ON amazon.vendedor.rfc = amazon.vendedorproducto.idvendedor
WHERE amazon.vendedorproducto.idproducto = 'ABCDE'
GROUP BY nombre
ORDER BY precio_mas_caro DESC;
  nombre | precio_mas_caro
-----+-----
Lupita independiente |          7500
Sony tienda original |          6500
El baúl de cachivaches Store |          6000
(3 rows)
```

g. ¿Cuánto dinero perdió cada tienda por artículos perdidos?

Comenzaremos por buscar todas las compras en las cuales se reembolso por tener artículos perdidos, recordemos que el status para estas ocasiones es 3.

```
baseG1_422023645⇒ SELECT * FROM amazon.compra JOIN amazon.vendedorproducto
ON amazon.compra.idproducto=amazon.vendedorproducto.id WHERE status=3;
```

id	idcliente	idproducto	fechacompra	status	id	idproducto	idvendedor	precio	unidades
8	delfines@gmail.com	1	2024-02-22	3	1	YUEYWU	GHAH55	7000	4
14	mapamundi@hotmail.com	10	2021-04-08	3	10	JJJ999JJ	UWYUE77	39.5	23
4	carlos@fi.unam.mx	10	2024-04-08	3	10	JJJ999JJ	UWYUE77	39.5	23

(3 rows)

En este caso podemos ver que un vendedor perdió el mismo producto 2 veces por lo cual se sumara apareciendo finalmente lo siguiente:

```
baseG1_422023645⇒ SELECT vendedor.nombre, SUM(vendedorproducto.precio) AS dinero_perdido
FROM amazon.vendedor
JOIN amazon.vendedorproducto ON amazon.vendedor.rfc = amazon.vendedorproducto.idvendedor
JOIN amazon.compra ON amazon.vendedorproducto.id = amazon.compra.idproducto
WHERE amazon.compra.status = 3
GROUP BY vendedor.nombre;
```

nombre	dinero_perdido
Lupita independiente	79
Sony tienda original	7000

(2 rows)

h. ¿Melissa hizo alguna compra en 2023?

Esto se comprueba con la sentencia EXISTS la cual devolverá f en caso de que no encuentre nada o t en caso de encontrar alguna coincidencia.

```
baseG1_422023645⇒ SELECT EXISTS (
  SELECT 1
  FROM amazon.compra
  JOIN amazon.cliente ON amazon.compra.idcliente = amazon.cliente.correo
  WHERE amazon.cliente.nombre = 'Melissa' AND EXTRACT(YEAR FROM amazon.compra.fechacompra) = 2023
) AS hizo_compra;
```

hizo_compra
f

(1 row)

En este caso nos mostro una f lo que significa que Melissa no hizo una compra en 2023, para confirmar esto listamos las compras de Melissa para poder observar la fecha.


```
base61_422023645⇒ SELECT idcliente, fechacompra, status, nombre, apellidopaterno, apellidomaterno
FROM amazon.compra JOIN amazon.cliente ON amazon.compra.idcliente = amazon.cliente.correo
WHERE amazon.cliente.nombre = 'Melissa';
```

idcliente	fechacompra	status	nombre	apellidopaterno	apellidomaterno
delfines@gmail.com	2024-02-22	2	Melissa	de la Cruz	Sullivan
delfines@gmail.com	2024-02-22	3	Melissa	de la Cruz	Sullivan
delfines@gmail.com	2024-01-19	1	Melissa	de la Cruz	Sullivan
delfines@gmail.com	2022-01-19	2	Melissa	de la Cruz	Sullivan

(4 rows)

Podemos ver que el resultado es correcto puesto que Melissa no realizo compras en 2023.