

Desafío - Entendiendo cómo se comportan nuestros clientes - Hans Contreras Robledo

Requerimientos:

1. Cargar el respaldo de la base de datos unidad2.sql.

Primero se crea la data base unidad2 en la Shell de SQL

```
Username [postgres]:
Contraseña para usuario postgres:
psql (14.2)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Dígame «help» para obtener ayuda.

postgres=# CREATE DATABASE unidad2;
CREATE DATABASE
postgres=# \c unidad2
Ahora está conectado a la base de datos «unidad2» con el usuario «postgres».
```

Luego cargamos por CMD el respaldo unidad2.sql a la data base creada en la SQL Shell

```
Microsoft Windows [Versión 10.0.19043.1021]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Administrator>cd desktop

C:\Users\Administrator\Desktop>psql -U postgres unidad2 < unidad2.sql
Contraseña para usuario postgres:
SET
SET
SET
SET
SET
set_config
-----
(1 fila)
SET
SET
SET
SET
SET
CREATE EXTENSION
COMMENT
SET
SET
CREATE TABLE
ALTER TABLE
```

Ahora volvemos a SQL Shell y comprobamos que haya cargado correctamente la información del respaldo

```
postgres=# CREATE DATABASE unidad2;
CREATE DATABASE
postgres=# \c unidad2
Ahora está conectado a la base de datos «unidad2» con el usuario «postgres».
unidad2=# \d

Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
public | cliente | tabla | postgres
public | cliente_id_seq | secuencia | postgres
public | compra | tabla | postgres
public | compra_id_seq | secuencia | postgres
public | detalle_compra | tabla | postgres
public | detalle_compra_id_seq | secuencia | postgres
public | producto | tabla | postgres
public | producto_id_seq | secuencia | postgres
(8 filas)
```

2. El cliente usuario01 ha realizado la siguiente compra:

- producto: producto9.
- cantidad: 5.
- fecha: fecha del sistema.

Mediante el uso de transacciones, realiza las consultas correspondientes para este requerimiento y luego consulta la tabla producto para validar si fue efectivamente descontado en el stock.

```
unidad2=# BEGIN TRANSACTION;
BEGIN
unidad2=# UPDATE producto SET stock = stock - 5 WHERE descripcion='producto9';
UPDATE 1
unidad2=# INSERT INTO compra (cliente_id, fecha) VALUES ((SELECT id FROM cliente WHERE nombre='usuario01'), CURRENT
DATE);
INSERT 0 1
unidad2=# INSERT INTO detalle_compra (producto_id, compra_id, cantidad)
VALUES ((SELECT id FROM producto WHERE descripcion='producto9'), (SELECT MAX(id) FROM compra), 5);
INSERT 0 1
unidad2=# COMMIT;
COMMIT

unidad2=# select * from producto where id=9;
 id | descripcion | stock | precio
-----+-----+-----+-----
  9 | producto9   |     3 |  4219
(1 fila)

unidad2=#
```

3. El cliente usuario02 ha realizado la siguiente compra:

- producto: producto1, producto 2, producto 8.
- cantidad: 3 de cada producto.
- fecha: fecha del sistema.

Mediante el uso de transacciones, realiza las consultas correspondientes para este requerimiento y luego consulta la tabla producto para validar que, si alguno de ellos se queda sin stock, no se realice la compra.

-Ingreso producto1

```
unidad2=# BEGIN TRANSACTION;
BEGIN
unidad2=# INSERT INTO compra (cliente_id, fecha)VALUES ((SELECT id FROM cliente WHERE nombre='usuario02'), CURRENT_DATE
);
INSERT 0 1
unidad2=# INSERT INTO detalle_compra (producto_id, compra_id, cantidad)
VALUES ((SELECT id FROM producto WHERE descripcion='producto1'), (SELECT MAX(id) FROM compra), 3);
INSERT 0 1
unidad2=# UPDATE producto SET stock = stock - 3 WHERE descripcion='producto1';
UPDATE 1
unidad2=# SAVEPOINT check_01;
SAVEPOINT
unidad2=# SELECT * FROM producto WHERE id=1;
 id | descripcion | stock | precio
-----+-----+-----+-----
  1 | producto1   |     3 |  9107
(1 fila)

unidad2=# COMMIT;
COMMIT
unidad2=#
```

-Ingreso producto2

```
unidad2=#
unidad2=# BEGIN TRANSACTION;
BEGIN
unidad2=# INSERT INTO compra (cliente_id, fecha)VALUES ((SELECT id FROM cliente WHERE nombre='usuario02'), CURRENT_DATE
);
INSERT 0 1
unidad2=# INSERT INTO detalle_compra (producto_id, compra_id, cantidad)
unidad2=# VALUES ((SELECT id FROM producto WHERE descripcion='producto2'), (SELECT MAX(id) FROM compra), 3);
INSERT 0 1
unidad2=# UPDATE producto SET stock = stock - 3 WHERE descripcion='producto2';
UPDATE 1
unidad2=# SAVEPOINT check_02;
SAVEPOINT
unidad2=# SELECT * FROM producto WHERE id=2;
 id | descripcion | stock | precio
-----+-----+-----+-----
  2 | producto2   |     2 |   1760
(1 fila)

unidad2=# COMMIT;
COMMIT
unidad2=#
```

-Ingreso producto8

```
unidad2=# BEGIN TRANSACTION;
BEGIN
unidad2=# INSERT INTO compra (cliente_id, fecha)VALUES ((SELECT id FROM cliente WHERE nombre='usuario02'), CURRENT_DATE
);
INSERT 0 1
unidad2=# INSERT INTO detalle_compra (producto_id, compra_id, cantidad)
unidad2=# VALUES ((SELECT id FROM producto WHERE descripcion='producto8'), (SELECT MAX(id) FROM compra), 3);
INSERT 0 1
unidad2=# UPDATE producto SET stock = stock - 3 WHERE descripcion='producto8';
ERROR:  el nuevo registro para la relación «producto» viola la restricción «check» «stock_valido»
DETALLE:  La fila que falla contiene (8, producto8, -3, 8923).
unidad2=# SELECT * FROM producto WHERE id=8;
ERROR:  transacción abortada, las órdenes serán ignoradas hasta el fin de bloque de transacción
unidad2=# ROLLBACK TO check_02;
ERROR:  no existe el «savepoint» «check_02»
unidad2=# COMMIT;
ROLLBACK
unidad2=#
```

Al ingresar el producto8 se genera un error ya que el stock a descontar para el producto8 es 3, pero el stock disponible es 0. Por lo que vuelve al estado antes del ingreso mediante rollback.

4. Realizar las siguientes consultas:

- Deshabilitar el AUTOCOMMIT
- Insertar un nuevo cliente

```
unidad2=#
unidad2=# \set AUTOCOMMIT off
unidad2=#
unidad2=# SAVEPOINT check_03;
SAVEPOINT
unidad2=# INSERT INTO cliente (nombre, email) VALUES ('usuario11','usuario11@gmail.com');
INSERT 0 1
unidad2=#
unidad2=#
```

- Confirmar que fue agregado en la tabla cliente

```
unidad2=#
unidad2=# SELECT * from cliente WHERE nombre='usuario11';
 id | nombre | email
-----+-----+-----
 13 | usuario11 | usuario11@gmail.com
(1 fila)

unidad2=#
```

- d. Realizar un ROLLBACK
- e. Confirmar que se restauró la información, sin considerar la inserción del punto b

```
unidad2=*# ROLLBACK TO check_03;
ROLLBACK
unidad2=*# SELECT * from cliente;
id | nombre | email
-----
2 | usuario02 | usuario02@yahoo.com
3 | usuario03 | usuario03@hotmail.com
4 | usuario04 | usuario04@hotmail.com
5 | usuario05 | usuario05@yahoo.com
6 | usuario06 | usuario06@hotmail.com
7 | usuario07 | usuario07@yahoo.com
8 | usuario08 | usuario08@yahoo.com
9 | usuario09 | usuario09@hotmail.com
10 | usuario010 | usuario010@hotmail.com
1 | usuario01 | usuario01@gmail.com
(10 filas)
```

- f. Habilitar de nuevo el AUTOCOMMIT

```
unidad2=*# ROLLBACK TO check_03;
ROLLBACK
unidad2=*# SELECT * from cliente;
id | nombre | email
-----
2 | usuario02 | usuario02@yahoo.com
3 | usuario03 | usuario03@hotmail.com
4 | usuario04 | usuario04@hotmail.com
5 | usuario05 | usuario05@yahoo.com
6 | usuario06 | usuario06@hotmail.com
7 | usuario07 | usuario07@yahoo.com
8 | usuario08 | usuario08@yahoo.com
9 | usuario09 | usuario09@hotmail.com
10 | usuario010 | usuario010@hotmail.com
1 | usuario01 | usuario01@gmail.com
(10 filas)

unidad2=*# \set AUTOCOMMIT on
unidad2=*#
```