**PASOS COMO CREAR UNA PROPUESTA DE ALUMNO “CARRITO DE COMPRAS”**

**Objetivo:**

Agregar funcionalidades para manejar un carrito de compras temporal para los clientes, facilitando la simulación de compras antes de generar un pedido formal.

* **PROPUESTA Nª 01**

CREAR PROPUESTA DE ALUMNO (TABLA SHOPPING\_CART)

* Crear tabla shopping\_cart

Esta tabla almacenará los productos que un cliente tiene en su carrito, con cantidades y precios actuales.

**CREATE** **TABLE** shopping\_cart (

cart\_id SERIAL **PRIMARY** **KEY**,

customer\_id **VARCHAR**(5) **REFERENCES** customers(customer\_id),

product\_id **INT** **REFERENCES** products(product\_id),

quantity **INT** **CHECK** (quantity > 0),

added\_at **TIMESTAMP** **DEFAULT** **NOW**(),

**UNIQUE** (customer\_id, product\_id)

);

* cart\_id: identificador único del ítem en el carrito
* customer\_id: referencia al cliente
* product\_id: referencia al producto
* quantity: cantidad del producto en el carrito
* added\_at: fecha y hora en que se agregó

**CREAR UNA VISTA (CARRITO\_CLIENTE\_TOTAL)**

* Crear vista carrito\_cliente\_total

Vista para mostrar el resumen total del carrito por cliente, con nombre de producto y total por producto.

**CREATE** **OR** **REPLACE** VIEW carrito\_cliente\_total **AS**

**SELECT**

sc.customer\_id,

c.company\_name,

sc.product\_id,

p.product\_name,

sc.quantity,

p.unit\_price,

(sc.quantity \* p.unit\_price) **AS** total\_por\_producto

**FROM**

shopping\_cart sc

**JOIN** customers c **ON** sc.customer\_id = c.customer\_id

**JOIN** products p **ON** sc.product\_id = p.product\_id

**ORDER** **BY**

sc.customer\_id, p.product\_name;

**CREAR UNA FUNCION (VACIAR\_CARRITO)**

* Crear función vaciar\_carrito

Función para eliminar todos los productos del carrito de un cliente específico.

**CREATE** **OR** **REPLACE** **FUNCTION** vaciar\_carrito(p\_customer\_id **VARCHAR**)

**RETURNS** VOID **AS** $$

**BEGIN**

**DELETE** **FROM** shopping\_cart **WHERE** customer\_id = p\_customer\_id;

**END**;

$$ **LANGUAGE** plpgsql;

**INGRESAR PRODUCTOS PARA DIFERENTES CLIENTES**

* Cliente 1: ALFKI

**INSERT** **INTO** shopping\_cart (customer\_id, product\_id, quantity)

**VALUES**

('ALFKI', 1, 2), -- Producto 1, cantidad 2

('ALFKI', 2, 1); -- Producto 2, cantidad 1

* Cliente 2: ANATR

**INSERT** **INTO** shopping\_cart (customer\_id, product\_id, quantity)

**VALUES**

('ANATR', 3, 4), -- Producto 3, cantidad 4

('ANATR', 4, 2); -- Producto 4, cantidad 2

* Cliente 3: BONAP

**INSERT** **INTO** shopping\_cart (customer\_id, product\_id, quantity)

**VALUES**

('BONAP', 5, 3);

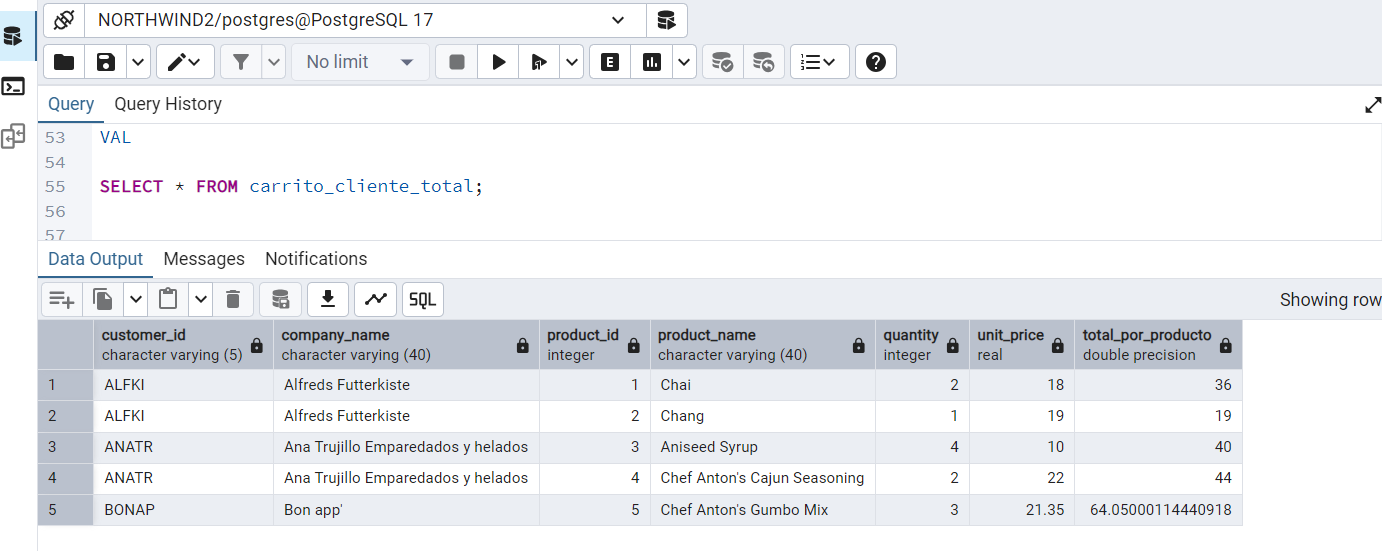
Asegúrate de que los valores de customer\_id y product\_id existen en tus tablas customers y products.

**SELECT** customer\_id **FROM** customers;

**SELECT** product\_id, product\_name **FROM** products **LIMIT** 10;

**VER EL CONTENIDO DEL CARRITO INGRESADO COMPLETO**

* **SELECT** \* **FROM** carrito\_cliente\_total;



VISUALIZAR INGRESO DE CARRITO POR CLIENTE

* Carrito del cliente ALFKI

**SELECT** \* **FROM** carrito\_cliente\_total **WHERE** customer\_id = 'ALFKI';

* Carrito del cliente ANATR

**SELECT** \* **FROM** carrito\_cliente\_total **WHERE** customer\_id = 'ANATR';

* Carrito del cliente BONAP

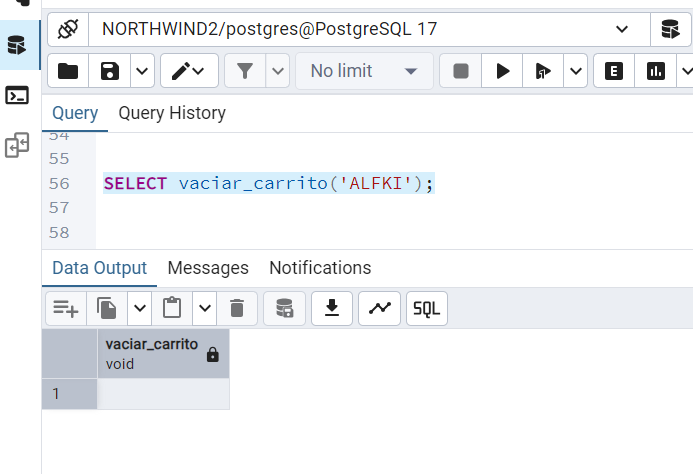
**SELECT** \* **FROM** carrito\_cliente\_total **WHERE** customer\_id = 'BONAP';

* USO DE LA FUNCIÓN

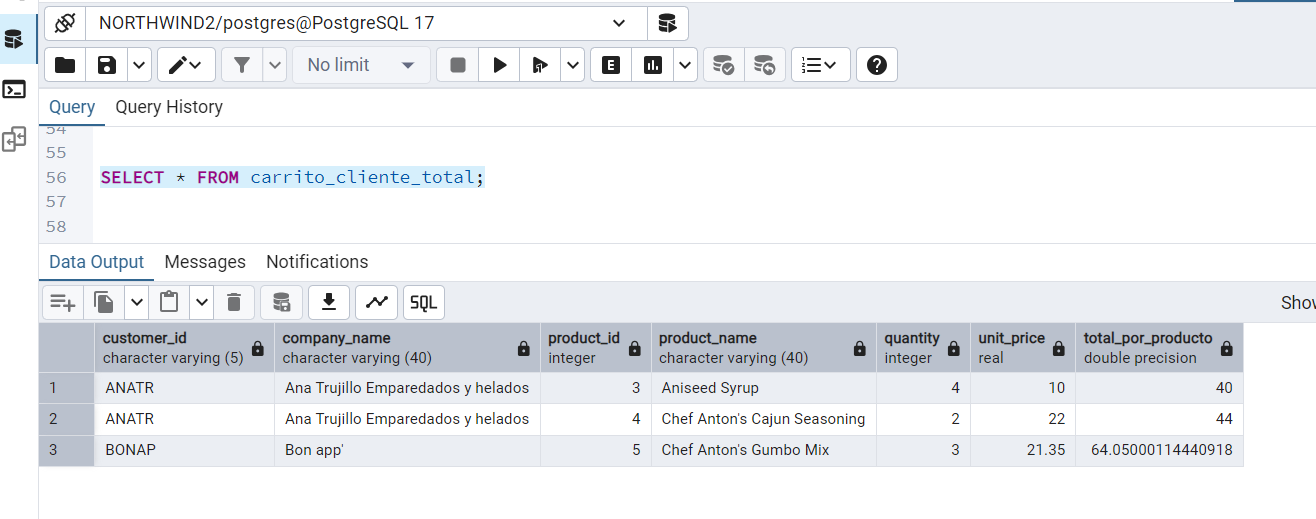
Función en PL/pgSQL que elimina todos los productos del carrito de un cliente específico. Representa la acción de "cancelar compra" o "vaciar carrito".

Para vaciar el carrito del cliente con ID 'ALFKI':

**SELECT** vaciar\_carrito('ALFKI');

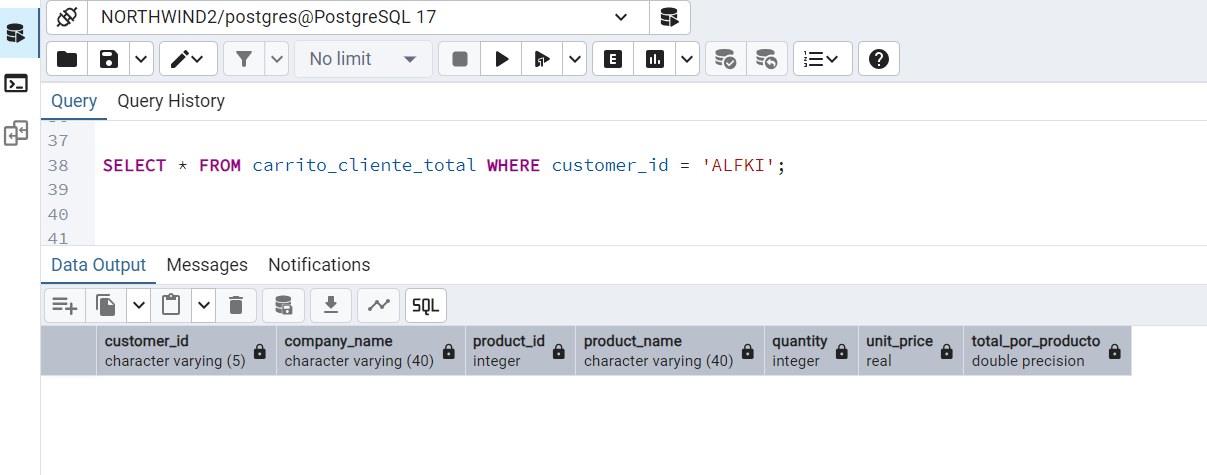


**VERIFICANDO LA ELIMINACION DEL CARRITO DEL CLIENTE ALFKI**



NOTA: También se puede usar para comprobar que el carrito quedó vacío

**SELECT** \* **FROM** carrito\_cliente\_total **WHERE** customer\_id = 'ALFKI';



PROPUESTA Nª 02

**MODIFICACIÓN DE LA TABLA PRODUCTS**

**Añadir columna JSONB para características dinámicas**

**ALTER** **TABLE** products

**ADD** **COLUMN** caracteristicas\_json JSONB;

**Insertar datos JSON con categoria y subcategoria**

**UPDATE** products

**SET** caracteristicas\_json = '{

"categoria": "Electrónica",

"subcategoria": "Audio",

"especificaciones": {

"potencia": "20W",

"color": "Negro"

}

}'

**WHERE** product\_id = 1;

**UPDATE** products

**SET** caracteristicas\_json = '{

"categoria": "Hogar",

"subcategoria": "Cocina",

"especificaciones": {

"material": "Acero inoxidable",

"garantía": "2 años"

}

}'

**WHERE** product\_id = 2;

**UPDATE** products

**SET** caracteristicas\_json = '{

"categoria": "Electrónica",

"subcategoria": "Video",

"especificaciones": {

"resolucion": "4K",

"pantalla": "LED"

}

}'

**WHERE** product\_id = 3;

**Consultas sobre datos JSON**

1. Productos por categoría

**SELECT** product\_id, product\_name, caracteristicas\_json

**FROM** products

**WHERE** caracteristicas\_json->>'categoria' = 'Electrónica';

1. Productos por subcategoría

**SELECT** product\_id, product\_name, caracteristicas\_json

**FROM** products

**WHERE** caracteristicas\_json->>'subcategoria' = 'Audio';

CONTROL DE STOCK INTELIGENTE

-- Crear vista para productos con stock bajo (< 10 unidades)

**CREATE** **OR** **REPLACE** VIEW productos\_stock\_bajo **AS**

**SELECT** product\_id, product\_name, units\_in\_stock

**FROM** products

**WHERE** units\_in\_stock < 10;

Consultar la vista

**SELECT** \* **FROM** productos\_stock\_bajo;

**PROPUESTA DEL ALUMNO: Añadir etiquetas a productos (tipo ARRAY**)

Añadir columna etiquetas (arreglo de texto)

**ALTER** **TABLE** products

**ADD** **COLUMN** etiquetas **TEXT**[];

Asignar etiquetas a productos

**UPDATE** products

**SET** etiquetas = **ARRAY**['orgánico', 'nuevo']

**WHERE** product\_id = 1;

**UPDATE** products

**SET** etiquetas = **ARRAY**['importado']

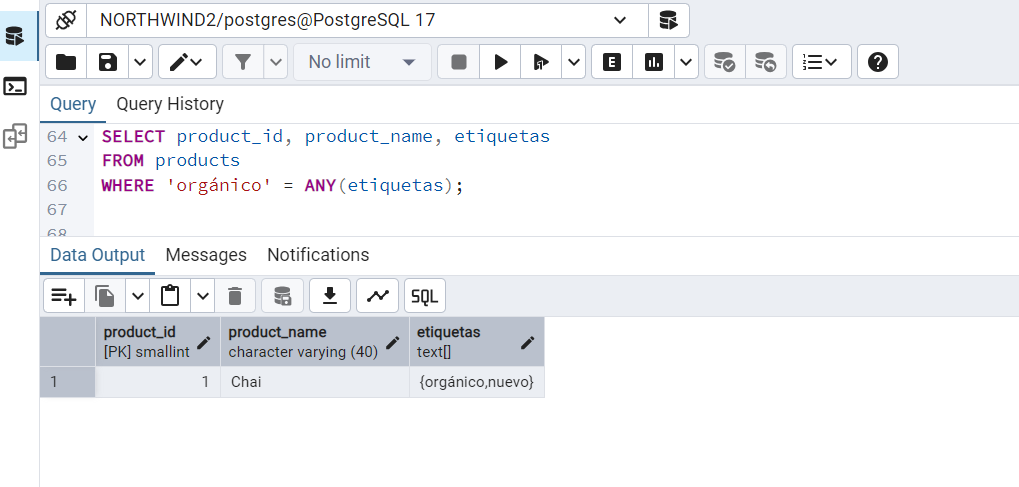
**WHERE** product\_id = 2;

Consulta: productos con una etiqueta específica

**SELECT** product\_id, product\_name, etiquetas

**FROM** products

**WHERE** 'orgánico' = **ANY**(etiquetas);



Este script realiza lo siguiente:

* Añade una columna JSONB para características flexibles.
* Inserta ejemplos con categoría y subcategoría.
* Crea y consulta una vista para productos con stock bajo.
* Implementa una propuesta con etiquetas (arrays de texto) para enriquecer la información de productos.