Tu propuesta de modificar la estructura actual tiene sentido y puede ayudarte a organizar mejor los datos. Aquí está el razonamiento detrás y algunas sugerencias sobre cómo podrías implementar la nueva estructura:

**Propuesta de Tablas**

**1. Tabla productos:**

* Esta tabla debería contener información estática sobre los productos, como su nombre, descripción y cualquier otra información relevante que no cambie con frecuencia.
* Aquí no deberías incluir datos que cambian dinámicamente como el stock o los precios específicos de transacciones.
* Campos sugeridos:
  + id: Identificador único del producto.
  + nombre: Nombre del producto.
  + descripcion: (opcional) Descripción del producto.
  + categoria: (opcional) Para clasificar el tipo de producto.

sql

Copiar código

CREATE TABLE productos (

id SERIAL PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

descripcion TEXT,

categoria VARCHAR(50)

);

**2. Tabla stock:**

* Esta tabla sería responsable de mantener el inventario actualizado. Aquí es donde almacenarías el stock actual disponible para cada producto.
* También puedes registrar cuándo se actualiza el stock.
* Campos sugeridos:
  + id: Identificador único para cada registro.
  + producto\_id: Llave foránea que referencia la tabla productos.
  + cantidad: La cantidad disponible en stock.
  + fecha\_actualizacion: (opcional) Fecha de la última actualización del stock.

sql

Copiar código

CREATE TABLE stock (

id SERIAL PRIMARY KEY,

producto\_id INT REFERENCES productos(id),

cantidad INT NOT NULL,

fecha\_actualizacion TIMESTAMP DEFAULT NOW()

);

**3. Tabla precios:**

* Si los precios cambian regularmente (por ejemplo, por promociones o ajustes de mercado), podrías llevar los precios a una tabla separada para registrar su historial.
* Campos sugeridos:
  + id: Identificador único.
  + producto\_id: Llave foránea que referencia la tabla productos.
  + precio: Precio de venta actual.
  + fecha\_inicio: Fecha en que este precio comenzó a aplicarse.
  + fecha\_fin: (opcional) Fecha en que este precio dejó de estar vigente.

sql

Copiar código

CREATE TABLE precios (

id SERIAL PRIMARY KEY,

producto\_id INT REFERENCES productos(id),

precio DECIMAL(10, 2) NOT NULL,

fecha\_inicio TIMESTAMP DEFAULT NOW(),

fecha\_fin TIMESTAMP

);

**Ventajas del Cambio**

* **Mejor organización:** Separar los productos y el stock en tablas diferentes te permite manejar mejor la información estática y dinámica. Por ejemplo, los productos no deberían cambiar, pero el stock se actualiza constantemente.
* **Historial de precios y cambios de stock:** Si registras los precios en una tabla aparte y actualizas los movimientos de stock, puedes tener un mejor control de cómo han cambiado estos valores a lo largo del tiempo.
* **Escalabilidad:** Al tener esta estructura, puedes fácilmente expandir el sistema para soportar diferentes almacenes, manejar devoluciones, o aplicar descuentos en precios.

**Flujo General para Compras y Ventas**

1. **Ventas:**
   * El sistema verifica el stock disponible en la tabla stock antes de realizar la venta.
   * Si hay suficiente stock, el sistema actualiza la cantidad en la tabla stock después de completar la transacción.
   * La tabla de precios te ayudará a obtener el precio de venta actual.
2. **Compras:**
   * Al agregar stock, simplemente aumentas la cantidad disponible en la tabla stock correspondiente a ese producto.

**Conclusión**

Este enfoque modular mejora la gestión de tu inventario y precios, facilitando el seguimiento de las transacciones. Cambiar la tabla actual de productos a stock y mover la información estática de los productos a una tabla separada tiene mucho sentido si buscas claridad y flexibilidad en la gestión de inventarios y ventas.