

FYT A GRANEL

Objetivo: Este proyecto tiene como finalidad la automatización y gestión operativa del emprendimiento de venta de productos a granel. Se implementa un sistema de base de datos en SQL que permite administrar compras, ventas, stock, precios y costos históricos de manera eficiente.

Estructura de la Base de Datos 1. Productos

Esta tabla almacena los productos disponibles para la venta, junto con su unidad de medida, costo de compra, precio de venta calculado automáticamente con un 30% de margen y el stock disponible.

Nombre de Campo	Tipo de Dato
id_producto	INT AUTO_INCREMENT PRIMARY KEY
nombre	VARCHAR(255) NOT NULL
unidad_medida	VARCHAR(50) NOT NULL
costo_unitario	DECIMAL(10,2) NOT NULL
precio_venta	DECIMAL(10,2) GENERATED ALWAYS AS (costo_unitario * 1.30) STORED
stock_disponible	DECIMAL(10,2) NOT NULL DEFAULT 0
fecha_actualizacion	DATE NOT NULL

2. Compras

Registra las compras realizadas con la fecha y número de remito correspondiente.

Nombre de Campo	Tipo de Dato
id_compra	INT AUTO_INCREMENT PRIMARY KEY
fecha_compra	DATE NOT NULL
numero_remito	VARCHAR(50) NOT NULL

3. Detalle de Compras

Guarda el detalle de cada compra, identificando el producto, su cantidad y su costo unitario.

Nombre de Campo	Tipo de Dato
id_detalle	INT AUTO_INCREMENT PRIMARY KEY

id_compra	INT NOT NULL
nombre_proveedor	VARCHAR(255) NOT NULL
id_producto	INT
cantidad	DECIMAL(10,2) NOT NULL
costo_unitario	DECIMAL(10,2) NOT NULL
FOREIGN KEY (id_compra)	REFERENCES Compras(id_compra)
FOREIGN KEY (id_producto)	REFERENCES Productos(id_producto)

4. Equivalencias

Dado que los productos pueden aparecer con diferentes nombres en los remitos de los proveedores, esta tabla permite mapearlos a los nombres usados internamente en la base de datos.

Nombre de Campo	Tipo de Dato
id_equivalencia	INT AUTO_INCREMENT PRIMARY KEY
nombre_proveedor	VARCHAR(255) NOT NULL
id_producto	INT NOT NULL
unidad_proveedor	VARCHAR(50)
FOREIGN KEY (id_producto)	REFERENCES Productos(id_producto)

5. Ventas

Registra cada transacción de venta con su fecha y total de venta calculado automáticamente.

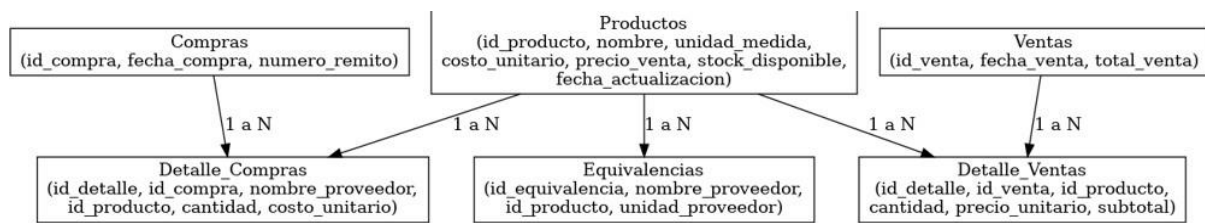
Nombre de Campo	Tipo de Dato
id_venta	INT AUTO_INCREMENT PRIMARY KEY
fecha_venta	DATE NOT NULL
total_venta	DECIMAL(10,2) DEFAULT 0

6. Detalle de Ventas

Almacena los productos vendidos en cada transacción junto con sus cantidades y precios unitarios.

Nombre de Campo	Tipo de Dato
id_detalle	INT AUTO_INCREMENT PRIMARY KEY
id_venta	INT NOT NULL
id_producto	INT NOT NULL
cantidad	DECIMAL(10,2) NOT NULL
precio_unitario	DECIMAL(10,2)
subtotal	DECIMAL(10,2)
FOREIGN KEY (id_venta)	REFERENCES Ventas(id_venta)
FOREIGN KEY (id_producto)	REFERENCES Productos(id_producto)

Diagrama ER



Triggers Implementados 1. before_insert_detalle_compras

- Antes de insertar un detalle de compra, busca el ID del producto en la tabla de equivalencias.
- Extrae la cantidad desde la descripción del proveedor.
- Completa automáticamente el id_producto y la cantidad en la compra.

2. after_insert_detalle_ventas

- Cada vez que se inserta un detalle de venta, se descuenta automáticamente la cantidad vendida del stock disponible en la tabla de productos.

3. actualizar_total_venta

- Luego de insertar un detalle de venta, recalcula y actualiza el total de la venta sumando los subtotales de los productos vendidos.

Vistas Implementadas 1. Vista_Costos_Historicos

- Permite visualizar los costos históricos de los productos junto con la fecha de compra y el número de remito correspondiente.

2. Vista_Ventas

- Muestra el total de cada venta calculado a partir de los subtotales de los productos vendidos.

Datos Precargados

Se han insertado registros de productos con sus costos iniciales, equivalencias con nombres de proveedores y algunas compras y ventas de prueba para validar la operatividad del sistema.

Consultas SQL Generadas

1. Ver Costos Históricos de los Productos

```
SELECT * FROM Vista_Costos_Historicos;
```

2. Obtener el Total de Ventas por Fecha

```
SELECT fecha_venta, SUM(total_venta) AS total_diario
```

```
FROM Ventas
```

```
GROUP BY fecha_venta
```

```
ORDER BY fecha_venta DESC;
```

3. Productos con Menor Stock Disponible

```
SELECT nombre, stock_disponible
```

```
FROM Productos
```

```
ORDER BY stock_disponible ASC
```

```
LIMIT 10;
```

4. Ventas Detalladas con Productos

```
SELECT v.id_venta, v.fecha_venta, p.nombre, dv.cantidad, dv.precio_unitario, dv.subtotal
```

```
FROM Detalle_Ventas dv
```

```
JOIN Ventas v ON dv.id_venta = v.id_venta
```

JOIN Productos p ON dv.id_producto = p.id_producto

ORDER BY v.fecha_venta DESC;

Beneficios del Sistema

- Automatización de costos y precios: El precio de venta se calcula automáticamente a partir del costo de compra.
- Actualización automática de stock: Los productos vendidos se descuentan del stock en tiempo real.
- Gestión de compras y remitos: Se facilita la identificación de productos comprados y su relación con los proveedores.
- Análisis de ventas y costos históricos: A través de las vistas, se pueden consultar los precios históricos de compra y el rendimiento de las ventas.

Este modelo de base de datos permitirá mejorar la eficiencia operativa del emprendimiento, reduciendo la carga de trabajo manual y asegurando un control preciso de los costos y el stock.