

Para manejar múltiples precios y descuentos en un modelo relacional, crea tablas separadas para Productos, Descuentos y Precios. La tabla de precios puede incluir el precio de referencia y las listas de precios (incluyendo segmentación por cliente o tipo de venta), mientras que la tabla de descuentos especifica el tipo de descuento (por cantidad, porcentaje, etc.) y los criterios de aplicación (mínimo, fecha, cliente). Usa tablas de relación o diseño de tablas intermedias si necesitas asociar varios descuentos a un mismo artículo o aplicar diferentes precios bajo criterios específicos.

Estructura de las tablas

Tabla: Fatura

RUC emisor

Denominación del documento

Numeración 15 dígitos

Número de autorización 10 dígitos (Otorgado por el SRI)

Fecha de autorización (dd/mm/aaaa)

RUC. / C.I. / Pasaporte del adquiriente

Nombre comercial (si consta el en RUC)

Razón social emisor

Dirección de la matriz / Dirección

establecimiento (cuando corresponda)

Intricación adquirente

Fecha de emisión

Fecha de caducidad

Forma de pago

Valor forma de pago

Número de Guía de Remisión (cuando corresponda)

Tabla: Línea de Factura

Descripción del bien o servicio

Cantidad

Precio unitario

Valor total (Cantidad * Precio Unitario)

Totalles

Base imponible 12%

Base imponible 0%

Descuento comercial

Valor subtotal (sin incluir impuestos)

Valor del IVA

Valor del ICE

Valor total

Tabla: Nota de Credito

Nombre comercial (si consta en el RUC) (Empresa)

Razón social emisor (Empresa)

Dirección de la matriz / Dirección (Empresa)

establecimiento (cuando corresponda) (Empresa)

RUC emisor (Empresa)

Denominación del documento

Numeración 15 dígitos

Número de autorización 10 dígitos (Otorgado por el SRI)

Fecha de autorización (dd/mm/aaaa)

Fecha de emisión

Tipo y número del comprobante que se modifica

Identificación del Adquiriente (Cliente)

RUC. / C.I. / Pasaporte del adquiriente (Cliente)

Descripción motivo de la modificacion

Fecha de caducidad (dd/mm/aaaa)

Tabla: Linea Nota Credito

Cantidad

Descripción del Item (Item)

Valor del Item Cobrado

Valor total (Cantidad * Precio Unitario)

Base imponible 12%

Base imponible 0%

Base imponible exento de IVA

Base imponible no objeto de IVA

Descuento comercial

Valor subtotal sin incluir impuestos

Valor del ICE

Valor del IVA

Valor total

Tabla Productos: Almacena la información básica de cada artículo.

id_producto (clave primaria)

nombre_producto

descripcion

precio_base (precio estándar)

Tabla Precios: Gestiona diferentes precios para los productos, permitiendo variaciones.

id_precio (clave primaria)

id_producto (clave foránea)

precio (el precio específico)

tipo_lista_precio (ej: "corporativo", "online", "mayorista")

fecha_inicio

fecha_fin

segmento_cliente (opcional, ej: "VIP", "Minorista")

Tabla TiposDescuento: Define los tipos de descuentos disponibles.

id_tipo_descuento (clave primaria)

nombre_tipo (ej: "Porcentaje", "Cantidad Fija", "2x1")

descripcion_tipo

Tabla Descuentos: Registra los descuentos aplicados, incluyendo sus condiciones.

id_descuento (clave primaria)

id_producto (clave foránea, opcional)

id_tipo_descuento (clave foránea)

valor (el porcentaje o cantidad del descuento)
cantidad_minima (ej: para descuentos por cantidad)
cliente_aplicable (opcional)
fecha_inicio_descuento
fecha_fin_descuento

Relaciones entre tablas

Productos a Precios

: Un producto puede tener varios precios (relación uno a muchos).

Productos a Descuentos

: Un producto puede tener varios descuentos, y un descuento puede aplicarse a varios productos (relación muchos a muchos, gestionada mediante la tabla Descuentos o una tabla intermedia si es más complejo).

TiposDescuento a Descuentos

: Un tipo de descuento se puede aplicar a varios descuentos (relación uno a muchos).

Consideraciones adicionales

Listas de precios

: Puedes crear diferentes listas de precios (por cliente, por temporada) que actúen como una forma de gestionar variaciones de precio.

Reglas de descuento

: Para descuentos más complejos, usa la tabla Descuentos para aplicar reglas de precio (como "3x2" o "comprar 1, llevar 2").

Vigencia

: Utiliza las fechas de inicio y fin en las tablas de Precios y Descuentos para gestionar promociones temporales y evitar que los descuentos expiren.

Tablas de enlace

: Si un descuento puede aplicarse a un grupo de productos en lugar de uno específico, crea una tabla intermedia (ej: DescuentoProducto) que relacione Descuentos y Productos.

A continuación, te presento un diseño optimizado de las tablas propuestas, integrando la estructura para manejar múltiples precios y descuentos en un modelo relacional, junto con las tablas de Factura, Línea de Factura, Nota de Crédito y Línea de Nota de Crédito. El diseño incluye las relaciones especificadas y consideraciones adicionales,

como tablas intermedias para manejar relaciones muchos a muchos y la gestión de fechas de vigencia.

Estructura de las tablas

Tabla: Productos

Almacena la información básica de los productos.

Campo	Tipo de dato	Descripción
id_producto	INT (PK)	Identificador único del producto
nombre_producto	VARCHAR(100)	Nombre del producto
descripcion	TEXT	Descripción del producto
precio_base	DECIMAL(10,2)	Precio estándar del producto

Tabla: Precios

Gestiona diferentes precios para un producto según el tipo de lista o segmento de cliente.

Campo	Tipo de dato	Descripción
id_precio	INT (PK)	Identificador único del precio
id_producto	INT (FK)	Referencia al producto
precio	DECIMAL(10,2)	Precio específico
tipo_lista_precio	VARCHAR(50)	Tipo de lista (ej: "corporativo", "mayorista")
fecha_inicio	DATE	Fecha de inicio de vigencia
fecha_fin	DATE	Fecha de fin de vigencia (opcional)
segmento_cliente	VARCHAR(50)	Segmento de cliente (ej: "VIP", "Minorista")

Tabla: TiposDescuento

Define los tipos de descuentos disponibles.

Campo	Tipo de dato	Descripción
id_tipo_descuento	INT (PK)	Identificador único del tipo de descuento

Campo	Tipo de dato	Descripción
nombre_tipo	VARCHAR(50)	Nombre del tipo (ej: "Porcentaje", "2x1")
descripcion_tipo	TEXT	Descripción del tipo de descuento

Tabla: Descuentos

Registra los descuentos aplicables con sus condiciones.

Campo	Tipo de dato	Descripción
id_descuento	INT (PK)	Identificador único del descuento
id_tipo_descuento	INT (FK)	Referencia al tipo de descuento
valor	DECIMAL(10,2)	Valor del descuento (porcentaje o cantidad)
cantidad_minima	INT	Cantidad mínima para aplicar el descuento
cliente_aplicable	VARCHAR(50)	Cliente o segmento al que aplica (opcional)
fecha_inicio_descuento	DATE	Fecha de inicio de vigencia
fecha_fin_descuento	DATE	Fecha de fin de vigencia (opcional)

Tabla: DescuentoProducto (Intermedia)

Tabla intermedia para relacionar descuentos con múltiples productos (muchos a muchos).

Campo	Tipo de dato	Descripción
id_descuento	INT (FK)	Referencia al descuento
id_producto	INT (FK)	Referencia al producto

Clave primaria compuesta: (id_descuento, id_producto)

Tabla: Factura

Almacena la información de las facturas emitidas.

Campo	Tipo de dato	Descripción
id_factura	INT (PK)	Identificador único de la factura

Campo	Tipo de dato	Descripción
ruc_emisor	VARCHAR(13)	RUC del emisor
denominacion_documento	VARCHAR(50)	Tipo de documento (ej: "Factura")
numeracion	VARCHAR(15)	Numeración de 15 dígitos
numero_autorizacion	VARCHAR(10)	Número de autorización del SRI
fecha_autorizacion	DATE	Fecha de autorización (dd/mm/aaaa)
identificacion_adquiriente	VARCHAR(13)	RUC, C.I. o Pasaporte del adquiriente
nombre_comercial	VARCHAR(100)	Nombre comercial del emisor (si aplica)
razon_social_emisor	VARCHAR(100)	Razón social del emisor
direccion_matriz	VARCHAR(200)	Dirección de la matriz del emisor
direccion_establecimiento	VARCHAR(200)	Dirección del establecimiento (si aplica)
fecha_emision	DATE	Fecha de emisión de la factura
fecha_caducidad	DATE	Fecha de caducidad
forma_pago	VARCHAR(50)	Forma de pago (ej: "Efectivo", "Transferencia")
valor_forma_pago	DECIMAL(10,2)	Valor asociado a la forma de pago
numero_guia_remision	VARCHAR(15)	Número de guía de remisión (si aplica)

Tabla: Línea de Factura

Registra los detalles de los ítems en una factura.

Campo	Tipo de dato	Descripción
id_linea_factura	INT (PK)	Identificador único de la línea
id_factura	INT (FK)	Referencia a la factura
id_producto	INT (FK)	Referencia al producto
descripcion	TEXT	Descripción del bien o servicio
cantidad	INT	Cantidad vendida

Campo	Tipo de dato	Descripción
precio_unitario	DECIMAL(10,2)	Precio unitario aplicado
valor_total	DECIMAL(10,2)	Cantidad * Precio unitario
id_descuento	INT (FK, opcional)	Descuento aplicado (si aplica)

Tabla: Totales Factura

Almacena los totales calculados de la factura.

Campo	Tipo de dato	Descripción
id_factura	INT (FK, PK)	Referencia a la factura
base_imponible_12	DECIMAL(10,2)	Base imponible con IVA 12%
base_imponible_0	DECIMAL(10,2)	Base imponible con IVA 0%
descuento_comercial	DECIMAL(10,2)	Descuento comercial aplicado
valor_subtotal	DECIMAL(10,2)	Subtotal sin impuestos
valor_iva	DECIMAL(10,2)	Valor del IVA
valor_ice	DECIMAL(10,2)	Valor del ICE (si aplica)
valor_total	DECIMAL(10,2)	Total de la factura

Tabla: Nota de Crédito

Registra las notas de crédito emitidas.

Campo	Tipo de dato	Descripción
id/nota_credito	INT (PK)	Identificador único de la nota de crédito
ruc_emisor	VARCHAR(13)	RUC del emisor
denominacion_documento	VARCHAR(50)	Tipo de documento (ej: "Nota de Crédito")
numeracion	VARCHAR(15)	Numeración de 15 dígitos
numero_autorizacion	VARCHAR(10)	Número de autorización del SRI
fecha_autorizacion	DATE	Fecha de autorización (dd/mm/aaaa)

Campo	Tipo de dato	Descripción
fecha_emision	DATE	Fecha de emisión
fecha_caducidad	DATE	Fecha de caducidad
id_factura_modificada	INT (FK)	Factura que se modifica
identificacion_adquiriente	VARCHAR(13)	RUC, C.I. o Pasaporte del adquiriente
nombre_comercial	VARCHAR(100)	Nombre comercial del emisor (si aplica)
razon_social_emisor	VARCHAR(100)	Razón social del emisor
direccion_matriz	VARCHAR(200)	Dirección de la matriz del emisor
direccion_establecimiento	VARCHAR(200)	Dirección del establecimiento (si aplica)
motivo_modificacion	TEXT	Descripción del motivo de la nota de crédito

Tabla: Línea de Nota de Crédito

Registra los detalles de los ítems en una nota de crédito.

Campo	Tipo de dato	Descripción
id_linea_nota_credito	INT (PK)	Identificador único de la línea
id_nota_credito	INT (FK)	Referencia a la nota de crédito
id_producto	INT (FK)	Referencia al producto
cantidad	INT	Cantidad devuelta
descripcion	TEXT	Descripción del ítem
valor_item_cobrado	DECIMAL(10,2)	Valor cobrado originalmente
valor_total	DECIMAL(10,2)	Cantidad * Valor unitario

Tabla: Totales Nota de Crédito

Almacena los totales calculados de la nota de crédito.

Campo	Tipo de dato	Descripción
id_nota_credito	INT (FK, PK)	Referencia a la nota de crédito

Campo	Tipo de dato	Descripción
base_imponible_12	DECIMAL(10,2)	Base imponible con IVA 12%
base_imponible_0	DECIMAL(10,2)	Base imponible con IVA 0%
base_imponible_exenta_iva	DECIMAL(10,2)	Base imponible exenta de IVA
base_imponible_no_objeto_iva	DECIMAL(10,2)	Base imponible no objeto de IVA
descuento_comercial	DECIMAL(10,2)	Descuento comercial aplicado
valor_subtotal	DECIMAL(10,2)	Subtotal sin impuestos
valor_iva	DECIMAL(10,2)	Valor del IVA
valor_ice	DECIMAL(10,2)	Valor del ICE (si aplica)
valor_total	DECIMAL(10,2)	Total de la nota de crédito

Relaciones entre tablas

1. Productos a Precios:

- Relación **uno a muchos**: Un producto (`id_producto`) puede tener múltiples precios en la tabla `Precios` (`id_producto` como clave foránea).

2. Productos a Descuentos:

- Relación **muchos a muchos**: Un producto puede tener múltiples descuentos, y un descuento puede aplicarse a múltiples productos. Esto se gestiona mediante la tabla intermedia `DescuentoProducto`.

3. TiposDescuento a Descuentos:

- Relación **uno a muchos**: Un tipo de descuento (`id_tipo_descuento`) puede estar asociado a múltiples descuentos en la tabla `Descuentos` (`id_tipo_descuento` como clave foránea).

4. Factura a Línea de Factura:

- Relación **uno a muchos**: Una factura (`id_factura`) puede tener múltiples líneas en la tabla `Línea de Factura` (`id_factura` como clave foránea).

5. Factura a Totales Factura:

- Relación **uno a uno**: Cada factura tiene un único registro de totales (`id_factura` como clave foránea y primaria).

6. Nota de Crédito a Línea de Nota de Crédito:

- Relación **uno a muchos**: Una nota de crédito (`id_nota_credito`) puede tener múltiples líneas en la tabla `Línea de Nota de Crédito` (`id_nota_credito` como clave foránea).

7. Nota de Crédito a Totales Nota de Crédito:

- Relación **uno a uno**: Cada nota de crédito tiene un único registro de totales (`id_nota_credito` como clave foránea y primaria).

8. Nota de Crédito a Factura:

- Relación **muchos a uno**: Una nota de crédito (`id_nota_credito`) está asociada a una factura específica (`id_factura_modificada` como clave foránea).

9. Línea de Factura a Productos:

- Relación **muchos a uno**: Cada línea de factura (`id_linea_factura`) está asociada a un producto (`id_producto` como clave foránea).

10. Línea de Factura a Descuentos:

- Relación **muchos a uno** (opcional): Una línea de factura puede estar asociada a un descuento específico (`id_descuento` como clave foránea).

11. Línea de Nota de Crédito a Productos:

- Relación **muchos a uno**: Cada línea de nota de crédito (`id_linea_nota_credito`) está asociada a un producto (`id_producto` como clave foránea).

Consideraciones adicionales

1. Listas de precios:

- La tabla `Precios` permite gestionar diferentes listas de precios mediante el campo `tipo_lista_precio` y `segmento_cliente`. Por ejemplo, un producto puede tener un precio "corporativo" de \$10 y un precio "minorista" de \$12.

2. Reglas de descuento:

- La tabla `Descuentos` soporta reglas complejas como "3x2" o "descuento por cantidad" mediante los campos `id_tipo_descuento` y `cantidad_minima`. Por

Por ejemplo, un descuento "2x1" se define en **TiposDescuento** y se aplica en **Descuentos** con un **valor** específico.

3. Vigencia:

- Los campos **fecha_inicio** y **fecha_fin** en **Precios** y **fecha_inicio_descuento** y **fecha_fin_descuento** en **Descuentos** permiten gestionar promociones temporales. Por ejemplo, un descuento del 20% puede aplicarse solo del 01/10/2025 al 31/10/2025.

4. Tablas de enlace:

- La tabla **DescuentoProducto** permite asociar un descuento a múltiples productos, ideal para promociones generales como "20% en todos los productos de una categoría".

5. Facturación y notas de crédito:

- Las tablas **Factura**, **Línea de Factura**, **Nota de Crédito** y **Línea de Nota de Crédito** cumplen con los requisitos del SRI para la emisión de comprobantes electrónicos, incluyendo información fiscal como RUC, número de autorización, y bases imponibles.
- La tabla **Totales Factura** y **Totales Nota de Crédito** consolidan los cálculos de impuestos (IVA, ICE) y descuentos comerciales.

6. Integridad referencial:

- Se deben definir claves foráneas con restricciones **ON DELETE** y **ON UPDATE** (por ejemplo, **CASCADE** o **RESTRICT**) para garantizar la consistencia de los datos.

7. Escalabilidad:

- El diseño permite añadir más campos según necesidades específicas, como categorías de productos, impuestos adicionales, o condiciones de pago más complejas.

Ejemplo de consultas SQL

1. Obtener el precio actual de un producto para un segmento específico:

```
sql
```

```
SELECT p.nombre_producto, pr.precio  
FROM Productos p
```

```
JOIN Precios pr ON p.id_producto = pr.id_producto
WHERE pr.segmento_cliente = 'VIP'
AND CURDATE() BETWEEN pr.fecha_inicio AND pr.fecha_fin;
```

2. Aplicar un descuento a una factura:

sql

```
SELECT lf.id_factura, lf.id_producto, lf.cantidad, lf.precio_unitario,
       (lf.cantidad * lf.precio_unitario) * (1 - d.valor / 100) AS valor_descuento
FROM Linea_Factura lf
JOIN Descuentos d ON lf.id_descuento = d.id_descuento
WHERE CURDATE() BETWEEN d.fecha_inicio_descuento AND d.fecha_fin_descuento;
```

3. Listar notas de crédito asociadas a una factura:

sql

```
SELECT nc.id_nota_credito, nc.fecha_emision, nc.motivo_modificacion
FROM Nota_Credito nc
WHERE nc.id_factura_modificada = 123;
```

Este diseño es robusto, flexible y cumple con los requisitos fiscales y de negocio descritos. Si necesitas un esquema visual (diagrama ER), un ejemplo de datos o una implementación específica en un motor de base de datos (como MySQL o PostgreSQL), házmelo saber.

puedes crear un modelo relacional para una empresa comercializadora de partes y piezas de automotores, teniendo en cuenta las estructuras que se mencionan y segmentando por línea, categoría, teniendo en cuenta que la empresa tiene sucursales y maneja varias bodegas en su inventario,

💡 Razonar por 1m 1s

Modelo Relacional para Empresa Comercializadora de Partes y Piezas de Automotores

A continuación, presento un diseño de modelo relacional extendido para una empresa que comercializa partes y piezas de automotores. Este modelo integra las estructuras previamente mencionadas (Productos, Precios, Descuentos, Facturas, Notas de Crédito, etc.) y añade segmentación por **línea** (ej: línea de frenos, motor, suspensión) y **categoría** (ej: accesorios, repuestos originales, genéricos). Además, incorpora la

gestión de **sucursales** (branches) y **bodegas** (warehouses) para el control de inventario, permitiendo rastrear stock por ubicación.

El diseño es flexible para manejar múltiples precios y descuentos (como en el modelo anterior), facturación fiscal (cumpliendo con requisitos como los del SRI en Ecuador), y operaciones en múltiples sucursales/bodegas. Utilizo un enfoque relacional estándar, con claves primarias (PK), foráneas (FK) y tablas intermedias para relaciones complejas.

Estructura de las Tablas

Tabla: Líneas

Segmenta los productos por línea de negocio o tipo principal (ej: "Frenos", "Motor").

Campo	Tipo de dato	Descripción
id_linea	INT (PK)	Identificador único de la línea
nombre_linea	VARCHAR(100)	Nombre de la línea
descripcion	TEXT	Descripción de la línea

Tabla: Categorías

Segmenta los productos por categoría (ej: "Repuestos Originales", "Accesorios").

Campo	Tipo de dato	Descripción
id_categoria	INT (PK)	Identificador único de la categoría
nombre_categoria	VARCHAR(100)	Nombre de la categoría
descripcion	TEXT	Descripción de la categoría

Tabla: Sucursales

Registra las sucursales de la empresa.

Campo	Tipo de dato	Descripción
id_sucursal	INT (PK)	Identificador único de la sucursal
nombre_sucursal	VARCHAR(100)	Nombre de la sucursal
direccion	VARCHAR(200)	Dirección de la sucursal

Campo	Tipo de dato	Descripción
telefono	VARCHAR(20)	Teléfono de contacto
ruc_sucursal	VARCHAR(13)	RUC asociado (si aplica por sucursal)

Tabla: Bodegas

Gestiona las bodegas (warehouses) asociadas a sucursales. Una sucursal puede tener múltiples bodegas.

Campo	Tipo de dato	Descripción
id_bodega	INT (PK)	Identificador único de la bodega
id_sucursal	INT (FK)	Referencia a la sucursal
nombre_bodega	VARCHAR(100)	Nombre de la bodega
direccion	VARCHAR(200)	Dirección de la bodega
capacidad_maxima	INT	Capacidad máxima (opcional, en unidades)

Tabla: Productos

Almacena la información básica de los productos, ahora segmentados por línea y categoría.

Campo	Tipo de dato	Descripción
id_producto	INT (PK)	Identificador único del producto
id_linea	INT (FK)	Referencia a la línea
id_categoria	INT (FK)	Referencia a la categoría
nombre_producto	VARCHAR(100)	Nombre del producto
descripcion	TEXT	Descripción del producto
precio_base	DECIMAL(10,2)	Precio estándar del producto
codigo_barras	VARCHAR(50)	Código de barras (opcional)
marca	VARCHAR(50)	Marca del producto (ej: Bosch, Toyota)

Tabla: Inventario

Gestiona el stock de productos por bodega, permitiendo control multi-bodega.

Campo	Tipo de dato	Descripción
id_inventario	INT (PK)	Identificador único del registro
id_producto	INT (FK)	Referencia al producto
id_bodega	INT (FK)	Referencia a la bodega
cantidad_stock	INT	Cantidad disponible en stock
cantidad_minima	INT	Stock mínimo para alerta
fecha_actualizacion	DATE	Fecha de última actualización

Tabla: Precios

Gestiona diferentes precios para productos, como en el modelo anterior, pero ahora puede vincularse a sucursales si los precios varían por ubicación.

Campo	Tipo de dato	Descripción
id_precio	INT (PK)	Identificador único del precio
id_producto	INT (FK)	Referencia al producto
id_sucursal	INT (FK, opcional)	Referencia a sucursal (si precios varían)
precio	DECIMAL(10,2)	Precio específico
tipo_lista_precio	VARCHAR(50)	Tipo de lista (ej: "mayorista", "minorista")
fecha_inicio	DATE	Fecha de inicio de vigencia
fecha_fin	DATE	Fecha de fin de vigencia (opcional)
segmento_cliente	VARCHAR(50)	Segmento de cliente (ej: "VIP")

Tabla: TiposDescuento

Define los tipos de descuentos, como en el modelo anterior.

Campo	Tipo de dato	Descripción
id_tipo_descuento	INT (PK)	Identificador único del tipo de descuento

Campo	Tipo de dato	Descripción
nombre_tipo	VARCHAR(50)	Nombre del tipo (ej: "Porcentaje", "2x1")
descripcion_tipo	TEXT	Descripción del tipo de descuento

Tabla: Descuentos

Registra descuentos, ahora con opción de aplicar por sucursal o bodega.

Campo	Tipo de dato	Descripción
id_descuento	INT (PK)	Identificador único del descuento
id_tipo_descuento	INT (FK)	Referencia al tipo de descuento
valor	DECIMAL(10,2)	Valor del descuento (porcentaje o cantidad)
cantidad_minima	INT	Cantidad mínima para aplicar
cliente_aplicable	VARCHAR(50)	Cliente o segmento (opcional)
id_sucursal	INT (FK, opcional)	Referencia a sucursal (si aplica por ubicación)
fecha_inicio_descuento	DATE	Fecha de inicio de vigencia
fecha_fin_descuento	DATE	Fecha de fin de vigencia (opcional)

Tabla: DescuentoProducto (Intermedia)

Relaciona descuentos con productos (muchos a muchos), como en el modelo anterior.

Campo	Tipo de dato	Descripción
id_descuento	INT (FK)	Referencia al descuento
id_producto	INT (FK)	Referencia al producto

Clave primaria compuesta: (id_descuento, id_producto)

Tabla: Factura

Almacena facturas, ahora asociadas a sucursales y bodegas (para rastrear origen del inventario).

Campo	Tipo de dato	Descripción
id_factura	INT (PK)	Identificador único de la factura
id_sucursal	INT (FK)	Referencia a la sucursal emisora
id_bodega	INT (FK, opcional)	Referencia a la bodega de despacho
ruc_emisor	VARCHAR(13)	RUC del emisor
denominacion_documento	VARCHAR(50)	Tipo de documento (ej: "Factura")
numeracion	VARCHAR(15)	Numeración de 15 dígitos
numero_autorizacion	VARCHAR(10)	Número de autorización del SRI
fecha_autorizacion	DATE	Fecha de autorización
identificacion_adquiriente	VARCHAR(13)	RUC/C.I./Pasaporte del adquiriente
nombre_comercial	VARCHAR(100)	Nombre comercial del emisor
razon_social_emisor	VARCHAR(100)	Razón social del emisor
direccion_matriz	VARCHAR(200)	Dirección de la matriz
direccion_establecimiento	VARCHAR(200)	Dirección del establecimiento
fecha_emision	DATE	Fecha de emisión
fecha_caducidad	DATE	Fecha de caducidad
forma_pago	VARCHAR(50)	Forma de pago
valor_forma_pago	DECIMAL(10,2)	Valor de la forma de pago
numero_guia_remision	VARCHAR(15)	Número de guía de remisión (opcional)

Tabla: Línea de Factura

Detalles de ítems en facturas, con referencia a inventario.

Campo	Tipo de dato	Descripción
id_linea_factura	INT (PK)	Identificador único de la línea
id_factura	INT (FK)	Referencia a la factura
id_producto	INT (FK)	Referencia al producto

Campo	Tipo de dato	Descripción
id_inventario	INT (FK, opcional)	Referencia al stock en bodega
descripcion	TEXT	Descripción del bien o servicio
cantidad	INT	Cantidad vendida
precio_unitario	DECIMAL(10,2)	Precio unitario aplicado
valor_total	DECIMAL(10,2)	Cantidad * Precio unitario
id_descuento	INT (FK, opcional)	Descuento aplicado

Tabla: Totales Factura

Totales de facturas, como en el modelo anterior.

Campo	Tipo de dato	Descripción
id_factura	INT (FK, PK)	Referencia a la factura
base_imponible_12	DECIMAL(10,2)	Base imponible con IVA 12%
base_imponible_0	DECIMAL(10,2)	Base imponible con IVA 0%
descuento_comercial	DECIMAL(10,2)	Descuento comercial
valor_subtotal	DECIMAL(10,2)	Subtotal sin impuestos
valor_iva	DECIMAL(10,2)	Valor del IVA
valor_ice	DECIMAL(10,2)	Valor del ICE
valor_total	DECIMAL(10,2)	Total de la factura

Tabla: Nota de Crédito

Notas de crédito, asociadas a sucursales.

Campo	Tipo de dato	Descripción
id/nota_credito	INT (PK)	Identificador único de la nota de crédito
id_sucursal	INT (FK)	Referencia a la sucursal emisora
ruc_emisor	VARCHAR(13)	RUC del emisor

Campo	Tipo de dato	Descripción
denominacion_documento	VARCHAR(50)	Tipo de documento
numeracion	VARCHAR(15)	Numeración de 15 dígitos
numero_autorizacion	VARCHAR(10)	Número de autorización del SRI
fecha_autorizacion	DATE	Fecha de autorización
fecha_emision	DATE	Fecha de emisión
fecha_caducidad	DATE	Fecha de caducidad
id_factura_modificada	INT (FK)	Factura que se modifica
identificacion_adquiriente	VARCHAR(13)	RUC/C.I./Pasaporte del adquiriente
nombre_comercial	VARCHAR(100)	Nombre comercial del emisor
razon_social_emisor	VARCHAR(100)	Razón social del emisor
direccion_matriz	VARCHAR(200)	Dirección de la matriz
direccion_establecimiento	VARCHAR(200)	Dirección del establecimiento
motivo_modificacion	TEXT	Motivo de la modificación

Tabla: Línea de Nota de Crédito

Detalles de ítems en notas de crédito.

Campo	Tipo de dato	Descripción
id_linea_nota_credito	INT (PK)	Identificador único de la línea
id_nota_credito	INT (FK)	Referencia a la nota de crédito
id_producto	INT (FK)	Referencia al producto
cantidad	INT	Cantidad devuelta
descripcion	TEXT	Descripción del ítem
valor_item_cobrado	DECIMAL(10,2)	Valor cobrado originalmente
valor_total	DECIMAL(10,2)	Cantidad * Valor unitario

Tabla: Totales Nota de Crédito

Totales de notas de crédito, como en el modelo anterior.

Campo	Tipo de dato	Descripción
id_nota_credito	INT (FK, PK)	Referencia a la nota de crédito
base_imponible_12	DECIMAL(10,2)	Base imponible con IVA 12%
base_imponible_0	DECIMAL(10,2)	Base imponible con IVA 0%
base_imponible_exenta_iva	DECIMAL(10,2)	Base imponible exenta de IVA
base_imponible_no_objeto_iva	DECIMAL(10,2)	Base imponible no objeto de IVA
descuento_comercial	DECIMAL(10,2)	Descuento comercial
valor_subtotal	DECIMAL(10,2)	Subtotal sin impuestos
valor_iva	DECIMAL(10,2)	Valor del IVA
valor_ice	DECIMAL(10,2)	Valor del ICE
valor_total	DECIMAL(10,2)	Total de la nota de crédito

Relaciones entre Tablas

- Líneas a Productos:** Uno a muchos (una línea tiene muchos productos).
- Categorías a Productos:** Uno a muchos (una categoría tiene muchos productos).
- Sucursales a Bodegas:** Uno a muchos (una sucursal tiene muchas bodegas).
- Productos a Inventory:** Uno a muchos (un producto tiene stock en muchas bodegas).
- Bodegas a Inventory:** Uno a muchos (una bodega almacena muchos productos).
- Productos a Precios:** Uno a muchos.
- Sucursales a Precios:** Uno a muchos (opcional, para precios por sucursal).
- TiposDescuento a Descuentos:** Uno a muchos.
- Productos a Descuentos:** Muchos a muchos (vía DescuentoProducto).
- Sucursales a Descuentos:** Uno a muchos (opcional).
- Sucursales a Factura:** Uno a muchos.

12. **Bodegas a Factura:** Uno a muchos (opcional).
 13. **Factura a Línea de Factura:** Uno a muchos.
 14. **Productos a Línea de Factura:** Muchos a uno.
 15. **Inventario a Línea de Factura:** Muchos a uno (opcional).
 16. **Descuentos a Línea de Factura:** Muchos a uno (opcional).
 17. **Factura a Totales Factura:** Uno a uno.
 18. **Sucursales a Nota de Crédito:** Uno a muchos.
 19. **Nota de Crédito a Factura:** Muchos a uno.
 20. **Nota de Crédito a Línea de Nota de Crédito:** Uno a muchos.
 21. **Productos a Línea de Nota de Crédito:** Muchos a uno.
 22. **Nota de Crédito a Totales Nota de Crédito:** Uno a uno.
-

Consideraciones Adicionales

1. **Segmentación por Línea y Categoría:** Permite consultas como "Productos de la línea 'Motor' en categoría 'Repuestos Originales'". Útil para reportes y filtros en el sistema.
2. **Gestión de Sucursales y Bodegas:** El inventario es multi-ubicación, lo que evita stock negativo al despachar desde bodegas específicas. Puedes agregar triggers para actualizar stock automáticamente al facturar.
3. **Inventario:** Soporta movimientos como entradas/salidas (puedes extender con una tabla de "MovimientosInventario" si necesitas historial).
4. **Precios y Descuentos:** Pueden variar por sucursal (ej: precios más altos en sucursales remotas). Las fechas de vigencia gestionan promociones temporales.
5. **Facturación y Cumplimiento Fiscal:** Mantiene los campos requeridos por el SRI, y asocia documentos a sucursales para auditorías.
6. **Escalabilidad:** Añade tablas como "Clientes" o "Proveedores" si es necesario. Usa índices en campos como id_producto para optimizar consultas.
7. **Integridad:** Define restricciones FK con ON DELETE RESTRICT (para evitar borrados accidentales) y triggers para cálculos automáticos (ej: actualizar valor_total).
8. **Ejemplos de Consultas:**

- Stock por bodega: `SELECT p.nombre_producto, i.cantidad_stock FROM Productos p JOIN Inventario i ON p.id_producto = i.id_producto WHERE i.id_bodega = 1;`
- Facturas por sucursal: `SELECT * FROM Factura WHERE id_sucursal = 2 AND fecha_emision >= '2025-01-01';`
- Productos con descuento en una línea: `SELECT p.nombre_producto, d.valor FROM Productos p JOIN DescuentoProducto dp ON p.id_producto = dp.id_producto JOIN Descuentos d ON dp.id_descuento = d.id_descuento WHERE p.id_linea = 3;`

Este modelo es completo y adaptable. Si necesitas un diagrama ER, scripts SQL para creación de tablas o extensiones (ej: módulos de compras), indícame.

Incluye scripts SQL creación

💡 Razonar por 47s

sql

```
-- Crear tabla Líneas
CREATE TABLE Lineas (
    id_linea INT AUTO_INCREMENT PRIMARY KEY,
    nombre_linea VARCHAR(100) NOT NULL,
    descripcion TEXT
);

-- Crear tabla Categorías
CREATE TABLE Categorias (
    id_categoria INT AUTO_INCREMENT PRIMARY KEY,
    nombre_categoria VARCHAR(100) NOT NULL,
    descripcion TEXT
);

-- Crear tabla Sucursales
CREATE TABLE Sucursales (
    id_sucursal INT AUTO_INCREMENT PRIMARY KEY,
    nombre_sucursal VARCHAR(100) NOT NULL,
    direccion VARCHAR(200),
    telefono VARCHAR(20),
    ruc_sucursal VARCHAR(13)
);

-- Crear tabla Bodegas
CREATE TABLE Bodegas (
    id_bodega INT AUTO_INCREMENT PRIMARY KEY
)
```

```
    id_sucursal INT NOT NULL,
    nombre_bodega VARCHAR(100) NOT NULL,
    direccion VARCHAR(200),
    capacidad_maxima INT,
    FOREIGN KEY (id_sucursal) REFERENCES Sucursales(id_sucursal) ON DEL
);

-- Crear tabla Productos
CREATE TABLE Productos (
    id_producto INT AUTO_INCREMENT PRIMARY KEY,
    id_linea INT NOT NULL,
    id_categoria INT NOT NULL,
    nombre_producto VARCHAR(100) NOT NULL,
    descripcion TEXT,
    precio_base DECIMAL(10,2) NOT NULL,
    codigo_barra VARCHAR(50),
    marca VARCHAR(50),
    FOREIGN KEY (id_linea) REFERENCES Lineas(id_linea) ON DELETE RESTRI
    FOREIGN KEY (id_categoria) REFERENCES Categorias(id_categoria) ON D
);

-- Crear tabla inventario
CREATE TABLE inventario (
    id_inventario INT AUTO_INCREMENT PRIMARY KEY,
    id_producto INT NOT NULL,
    id_bodega INT NOT NULL,
    cantidad_stock INT NOT NULL DEFAULT 0,
    cantidad_minima INT DEFAULT 0,
    fecha_actualizacion DATE,
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELE
    FOREIGN KEY (id_bodega) REFERENCES Bodegas(id_bodega) ON DELETE CAS
);

-- Crear tabla Precios
CREATE TABLE Precios (
    id_precio INT AUTO_INCREMENT PRIMARY KEY,
    id_producto INT NOT NULL,
    id_sucursal INT,
    precio DECIMAL(10,2) NOT NULL,
    tipo_lista_precio VARCHAR(50),
    fecha_inicio DATE,
    fecha_fin DATE,
    segmento_cliente VARCHAR(50),
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELE
    FOREIGN KEY (id_sucursal) REFERENCES Sucursales(id_sucursal) ON DEL
);

-- Crear tabla TiposDescuento
```

```
CREATE TABLE TiposDescuento (
    id_tipo_descuento INT AUTO_INCREMENT PRIMARY KEY,
    nombre_tipo VARCHAR(50) NOT NULL,
    descripcion_tipo TEXT
);

-- Crear tabla Descuentos
CREATE TABLE Descuentos (
    id_descuento INT AUTO_INCREMENT PRIMARY KEY,
    id_tipo_descuento INT NOT NULL,
    valor DECIMAL(10,2) NOT NULL,
    cantidad_minima INT,
    cliente_aplicable VARCHAR(50),
    id_sucursal INT,
    fecha_inicio_descuento DATE,
    fecha_fin_descuento DATE,
    FOREIGN KEY (id_tipo_descuento) REFERENCES TiposDescuento(id_tipo_d)
    FOREIGN KEY (id_sucursal) REFERENCES Sucursales(id_sucursal) ON DEL
);

-- Crear tabla DescuentoProducto (Intermedia)
CREATE TABLE DescuentoProducto (
    id_descuento INT NOT NULL,
    id_producto INT NOT NULL,
    PRIMARY KEY (id_descuento, id_producto),
    FOREIGN KEY (id_descuento) REFERENCES Descuentos(id_descuento) ON D
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELE
);

-- Crear tabla Factura
CREATE TABLE Factura (
    id_factura INT AUTO_INCREMENT PRIMARY KEY,
    id_sucursal INT NOT NULL,
    id_bodega INT,
    ruc_emisor VARCHAR(13) NOT NULL,
    denominacion_documento VARCHAR(50) NOT NULL,
    numeracion VARCHAR(15) NOT NULL,
    numero_autorizacion VARCHAR(10) NOT NULL,
    fecha_autorizacion DATE NOT NULL,
    identificacion_adquiriente VARCHAR(13) NOT NULL,
    nombre_comercial VARCHAR(100),
    razon_social_emisor VARCHAR(100) NOT NULL,
    direccion_matriz VARCHAR(200) NOT NULL,
    direccion_establecimiento VARCHAR(200),
    fecha_emision DATE NOT NULL,
    fecha_caducidad DATE,
    forma_pago VARCHAR(50) NOT NULL,
    valor_forma_pago DECIMAL(10,2) NOT NULL,
    numero_guia_remision VARCHAR(15)
```

```
FOREIGN KEY (id_sucursal) REFERENCES Sucursales(id_sucursal) ON DELETE C
FOREIGN KEY (id_bodega) REFERENCES Bodegas(id_bodega) ON DELETE SET
);

-- Crear tabla Línea de Factura
CREATE TABLE LineaFactura (
    id_linea_factura INT AUTO_INCREMENT PRIMARY KEY,
    id_factura INT NOT NULL,
    id_producto INT NOT NULL,
    id_inventario INT,
    descripcion TEXT NOT NULL,
    cantidad INT NOT NULL,
    precio_unitario DECIMAL(10,2) NOT NULL,
    valor_total DECIMAL(10,2) NOT NULL,
    id_descuento INT,
    FOREIGN KEY (id_factura) REFERENCES Factura(id_factura) ON DELETE C
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELE
    FOREIGN KEY (id_inventario) REFERENCES inventario(id_inventario) ON
    FOREIGN KEY (id_descuento) REFERENCES Descuentos(id_descuento) ON D
);

-- Crear tabla Totales Factura
CREATE TABLE TotalesFactura (
    id_factura INT PRIMARY KEY,
    base_imponible_12 DECIMAL(10,2) DEFAULT 0,
    base_imponible_0 DECIMAL(10,2) DEFAULT 0,
    descuento_comercial DECIMAL(10,2) DEFAULT 0,
    valor_subtotal DECIMAL(10,2) NOT NULL,
    valor_iva DECIMAL(10,2) DEFAULT 0,
    valor_ice DECIMAL(10,2) DEFAULT 0,
    valor_total DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (id_factura) REFERENCES Factura(id_factura) ON DELETE C
);

-- Crear tabla Nota de Crédito
CREATE TABLE NotaCredito (
    id_nota_credito INT AUTO_INCREMENT PRIMARY KEY,
    id_sucursal INT NOT NULL,
    ruc_emisor VARCHAR(13) NOT NULL,
    denominacion_documento VARCHAR(50) NOT NULL,
    numeracion VARCHAR(15) NOT NULL,
    numero_autorizacion VARCHAR(10) NOT NULL,
    fecha_autorizacion DATE NOT NULL,
    fecha_emision DATE NOT NULL,
    fecha_caducidad DATE,
    id_factura_modificada INT NOT NULL,
    identificacion_adquiriente VARCHAR(13) NOT NULL,
    nombre_comercial VARCHAR(100),
```

```

    razon_social_emisor VARCHAR(100) NOT NULL,
    direccion_matriz VARCHAR(200) NOT NULL,
    direccion_establecimiento VARCHAR(200),
    motivo_modificacion TEXT NOT NULL,
    FOREIGN KEY (id_sucursal) REFERENCES Sucursales(id_sucursal) ON DEL
    FOREIGN KEY (id_factura_modificada) REFERENCES Factura(id_factura)
);

```

-- Crear tabla Línea de Nota de Crédito

```

CREATE TABLE LineaNotaCredito (
    id_linea_nota_credito INT AUTO_INCREMENT PRIMARY KEY,
    id_nota_credito INT NOT NULL,
    id_producto INT NOT NULL,
    cantidad INT NOT NULL,
    descripcion TEXT NOT NULL,
    valor_item_cobrado DECIMAL(10,2) NOT NULL,
    valor_total DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (id_nota_credito) REFERENCES NotaCredito(id_nota_credito),
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELETE
);

```

-- Crear tabla Totales Nota de Crédito

```

CREATE TABLE TotalesNotaCredito (
    id_nota_credito INT PRIMARY KEY,
    base_imponible_12 DECIMAL(10,2) DEFAULT 0,
    base_imponible_0 DECIMAL(10,2) DEFAULT 0,
    base_imponible_exenta_iva DECIMAL(10,2) DEFAULT 0,
    base_imponible_no_objeto_iva DECIMAL(10,2) DEFAULT 0,
    descuento_comercial DECIMAL(10,2) DEFAULT 0,
    valor_subtotal DECIMAL(10,2) NOT NULL,
    valor_iva DECIMAL(10,2) DEFAULT 0,
    valor_ice DECIMAL(10,2) DEFAULT 0,
    valor_total DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (id_nota_credito) REFERENCES NotaCredito(id_nota_credito)
);

```

puedo obtener un reporte de inventario con las compras y ventas de los productos que tenian los productos en el momento de cada transaccion,

Para generar un **reporte de inventario** que incluya las **compras** y **ventas** de productos, junto con el **stock** disponible en el momento de cada transacción, es necesario extender el modelo relacional con una tabla de **Movimientos de Inventario** para registrar las transacciones de entrada (compras) y salida (ventas). Luego, se puede crear una consulta SQL que combine esta información con los datos de inventario,

productos, facturas y bodegas, mostrando el estado del stock en el momento de cada transacción.

Paso 1: Extender el modelo con la tabla MovimientosInventario

Primero, agregamos una tabla para registrar los movimientos de inventario (entradas y salidas), vinculada a productos, bodegas, facturas (para ventas) y, opcionalmente, a documentos de compra (como órdenes de compra).

Tabla: MovimientosInventario

Registra los movimientos de inventario (compras y ventas).

Campo	Tipo de dato	Descripción
id_movimiento	INT (PK)	Identificador único del movimiento
id_producto	INT (FK)	Referencia al producto
id_bodega	INT (FK)	Referencia a la bodega
id_factura	INT (FK, opcional)	Referencia a factura (si es venta)
id_orden_compra	INT (FK, opcional)	Referencia a orden de compra (si es compra)
tipo_movimiento	ENUM('ENTRADA', 'SALIDA')	Tipo de movimiento (compra o venta)
cantidad	INT	Cantidad movida
fecha_movimiento	DATETIME	Fecha y hora del movimiento
stock_post_movimiento	INT	Stock en la bodega tras el movimiento

SQL para crear la tabla MovimientosInventario:

sql

```
CREATE TABLE MovimientosInventario (
    id_movimiento INT AUTO_INCREMENT PRIMARY KEY,
    id_producto INT NOT NULL,
    id_bodega INT NOT NULL,
    id_factura INT,
    id_orden_compra INT,
    tipo_movimiento ENUM('ENTRADA', 'SALIDA') NOT NULL,
    cantidad INT NOT NULL,
    fecha_movimiento DATETIME NOT NULL,
```

```

stock_post_movimiento INT NOT NULL,
FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELETE RE
FOREIGN KEY (id_bODEga) REFERENCES Bodegas(id_bodega) ON DELETE RE
FOREIGN KEY (id_factura) REFERENCES Factura(id_factura) ON DELETE S
FOREIGN KEY (id_orden_compra) REFERENCES OrdenesCompra(id_orden_com
);

```

Nota: La tabla `OrdenesCompra` no estaba en el modelo original, pero se asume que existe para registrar compras. Si no la tienes, puedes omitir el campo

`id_orden_compra` o crear una tabla básica como:

sql

```

CREATE TABLE OrdenesCompra (
    id_orden_compra INT AUTO_INCREMENT PRIMARY KEY,
    numero_orden VARCHAR(50) NOT NULL,
    fecha_compra DATE NOT NULL,
    proveedor VARCHAR(100)
);

```

Paso 2: Lógica del reporte

El reporte debe:

1. Mostrar los movimientos de inventario (compras y ventas) por producto y bodega.
2. Incluir el `stock_post_movimiento` para reflejar el estado del inventario tras cada transacción.
3. Mostrar detalles del producto (nombre, línea, categoría), bodega, sucursal, y tipo de transacción (entrada o salida).
4. Opcionalmente, incluir información de la factura (para ventas) o la orden de compra (para entradas).
5. Ordenar los resultados por fecha de movimiento para ver la evolución del stock.

Paso 3: Consulta SQL para el reporte

La siguiente consulta SQL genera el reporte de inventario con compras y ventas, incluyendo el stock en el momento de cada transacción:

sql

```

SELECT
    m.id_movimiento,
    m.fecha_movimiento

```

```

    m.id_movimiento,
    p.nombre_producto,
    l.nombre_linea,
    c.nombre_categoria,
    b.nombre_bodega,
    s.nombre_sucursal,
    m.tipo_movimiento,
    m.cantidad,
    m.stock_post_movimiento,
    COALESCE(f.numeracion, 'N/A') AS numero_factura,
    COALESCE(oc.numero_orden, 'N/A') AS numero_orden_compra
FROM
    MovimientosInventario m
    JOIN Productos p ON m.id_producto = p.id_producto
    JOIN Lineas l ON p.id_linea = l.id_linea
    JOIN Categorias c ON p.id_categoria = c.id_categoria
    JOIN Bodegas b ON m.id_bodega = b.id_bodega
    JOIN Sucursales s ON b.id_sucursal = s.id_sucursal
    LEFT JOIN Factura f ON m.id_factura = f.id_factura
    LEFT JOIN OrdenesCompra oc ON m.id_orden_compra = oc.id_orden_compr
ORDER BY
    m.fecha_movimiento ASC, p.nombre_producto, b.nombre_bodega;

```

Explicación de la consulta

- **Tablas involucradas:**
 - **MovimientosInventario**: Proporciona los movimientos (entradas/salidas) con cantidades y stock tras cada transacción.
 - **Productos , Lineas , Categorias**: Detalles del producto, su línea y categoría.
 - **Bodegas , Sucursales**: Ubicación del inventario.
 - **Factura** (LEFT JOIN): Vincula ventas con facturas (opcional, ya que las entradas no tienen factura).
 - **OrdenesCompra** (LEFT JOIN): Vincula compras con órdenes de compra (opcional, para entradas).
- **Campos seleccionados:**
 - **id_movimiento**: Identificador del movimiento.
 - **fecha_movimiento**: Fecha y hora de la transacción.
 - **nombre_producto , nombre_linea , nombre_categoria**: Información del producto.

- `nombre_bodega`, `nombre_sucursal`: Ubicación del movimiento.
- `tipo_movimiento`: Indica si es entrada (compra) o salida (venta).
- `cantidad`: Cantidad movida.
- `stock_post_movimiento`: Stock en la bodega tras el movimiento.
- `numero_factura`, `numero_orden_compra`: Documentos asociados (si aplican).
- **Ordenamiento:** Por fecha de movimiento, producto y bodega para un seguimiento cronológico claro.

Paso 4: Trigger para actualizar stock

Para garantizar que el campo `stock_post_movimiento` sea consistente, puedes usar un **trigger** que actualice el stock en `Inventario` y registre el valor en `MovimientosInventario` tras cada transacción.

Ejemplo de trigger para insertar en MovimientosInventario:

sql

```

DELIMITER //
CREATE TRIGGER after_movimiento_insert
AFTER INSERT ON MovimientosInventario
FOR EACH ROW
BEGIN
    DECLARE nuevo_stock INT;

    -- Calcular el nuevo stock en la tabla Inventario
    IF NEW.tipo_movimiento = 'ENTRADA' THEN
        SET nuevo_stock = (
            SELECT cantidad_stock
            FROM Inventario
            WHERE id_producto = NEW.id_producto AND id_bodega = NEW.id_
        ) + NEW.cantidad;
    ELSE
        SET nuevo_stock = (
            SELECT cantidad_stock
            FROM Inventario
            WHERE id_producto = NEW.id_producto AND id_bodega = NEW.id_
        ) - NEW.cantidad;
    END IF;

    -- Actualizar stock en la tabla Inventario
    UPDATE Inventario
    SET cantidad_stock = nuevo_stock,
        fecha_actualizacion = CURDATE()

```

```

WHERE id_producto = NEW.id_producto AND id_bodega = NEW.id_bodega;

-- Actualizar stock_post_movimiento en el movimiento recién inserta
UPDATE MovimientosInventario
SET stock_post_movimiento = nuevo_stock
WHERE id_movimiento = NEW.id_movimiento;
END //
DELIMITER ;

```

Paso 5: Ejemplo de datos y resultado esperado

Supongamos los siguientes datos:

- Producto: "Pastillas de freno" (id_producto=1, linea="Frenos", categoría="Repuestos Originales").
- Bodega: "Bodega Central" (id_bodega=1, sucursal="Sucursal Quito").
- Movimientos:
 - 2025-09-01: Compra de 100 unidades (orden de compra #001), stock inicial 0 → stock final 100.
 - 2025-09-15: Venta de 20 unidades (factura #F001), stock final 80.
 - 2025-09-20: Compra de 50 unidades (orden de compra #002), stock final 130.

Salida del reporte (ejemplo):

id_movimiento	fecha_movimiento	nombre_producto	nombre_linea	nombre_categoria
1	2025-09-01 10:00:00	Pastillas de freno	Frenos	Repuestos Originales
2	2025-09-15 14:30:00	Pastillas de freno	Frenos	Repuestos Originales
3	2025-09-20 09:00:00	Pastillas de freno	Frenos	Repuestos Originales

Consideraciones adicionales

1. **Integridad del stock:** El trigger asegura que el stock en **Inventario** y **MovimientosInventario** sea consistente. Puedes agregar validaciones para evitar stock negativo.

2. **Filtros en el reporte:** Modifica la consulta para filtrar por fecha, producto, bodega o sucursal según necesidades (ej: `WHERE m.fecha_movimiento BETWEEN '2025-09-01' AND '2025-09-30'`).
3. **Escalabilidad:** Si necesitas más detalles (como precios en el momento de la venta), une con la tabla `Precios` usando `fecha_movimiento` para obtener el precio vigente.
4. **Auditoría:** La tabla `MovimientosInventario` actúa como historial, permitiendo rastrear cambios en el inventario.

Si necesitas ajustar el reporte (por ejemplo, agregar precios o filtrar por línea/categoría), implementar más triggers, o generar un ejemplo con datos de prueba, házmelo saber.

Como manejaría los precios del inventario teniendo en cuenta que los productos pueden adquirirse a diferente precio en cada compra, que modelo sería recomendable utilizar FIFO o LIFO y como se implementaría en el modelo

💡 Razonar por 1m 58s

Manejo de Precios en Inventario con Costos Variables por Compra

Cuando los productos se adquieren a precios diferentes en cada compra, es esencial implementar un método de valoración de inventarios para determinar el **costo de los bienes vendidos (COGS)** de manera precisa y consistente. Esto afecta directamente los estados financieros (como el balance general y el estado de resultados), los impuestos y la rentabilidad reportada. En un modelo relacional como el que hemos diseñado para la empresa de partes automotrices, el manejo de estos precios variables requiere rastrear los costos asociados a cada entrada (compra) y aplicarlos correctamente en las salidas (ventas).

Los métodos comunes para valorar inventarios en escenarios de costos variables son:

- **FIFO (First In, First Out o PEPS - Primero en Entrar, Primero en Salir):** Asume que los productos más antiguos (con su costo original) se venden primero. Esto refleja el flujo físico real de muchos inventarios (especialmente no perecederos como partes automotrices) y resulta en un costo de ventas más bajo durante períodos de inflación, ya que usa costos históricos más bajos.
- **LIFO (Last In, First Out o UEPS - Último en Entrar, Primero en Salir):** Asume que los productos más recientes se venden primero. Esto puede reducir la renta

imponible en inflación (usando costos más altos para ventas), pero tiene desventajas como distorsionar el valor del inventario restante.

- **Costo Promedio Ponderado:** Calcula un costo promedio de todas las unidades en stock y lo aplica a las salidas. Es simple, pero no rastrea costos individuales por lote.

Recomendación: ¿FIFO o LIFO?

En el contexto de Ecuador, donde la empresa opera bajo regulaciones del SRI (Servicio de Rentas Internas) y las Normas Internacionales de Información Financiera (NIIF o IFRS, adoptadas desde 2010), **recomiendo el método FIFO**. Aquí las razones clave:

- **Cumplimiento normativo:** La NIC 2 (Inventarios) de las NIIF prohíbe explícitamente el uso de LIFO para reportes financieros, ya que no refleja el flujo real de costos y puede subvalorar el inventario restante. Los métodos permitidos son FIFO, costo promedio ponderado e identificación específica (para ítems únicos). LIFO solo se menciona en contextos generales o históricos, pero no es aceptado bajo NIIF. Para fines fiscales con el SRI, los reportes deben alinearse con las NIIF, por lo que LIFO podría generar discrepancias en auditorías o declaraciones de impuestos (como el Formulario 101 para renta). contiflico.com
- **Ventajas prácticas para partes automotores:** FIFO es lógico para productos duraderos, ya que evita obsolescencia (vende lo viejo primero) y proporciona un balance más realista. En períodos de inflación (común en Ecuador), FIFO reporta mayores utilidades, lo que puede ser beneficioso para inversionistas, aunque aumenta la carga fiscal ligeramente comparado con LIFO.
- **Desventajas de LIFO:** Aunque podría reducir impuestos en inflación al usar costos altos para ventas, su prohibición bajo NIIF lo hace inviable para empresas que preparan estados financieros estandarizados. Además, distorsiona el valor del inventario (lo deja con costos antiguos bajos), lo que no es ideal para toma de decisiones.

Si los costos son muy volátiles o el inventario es homogéneo, considera el **costo promedio ponderado** como alternativa simple a FIFO, ya que también está permitido y evita la complejidad de rastrear lotes individuales.

Implementación en el Modelo Relacional

Para implementar FIFO (recomendado), extendemos el modelo con una tabla de **Lotes** que rastrea cada compra como un "lote" independiente con su costo unitario. Esto permite despachar ventas del lote más antiguo disponible. Usamos la tabla

MovimientosInventario (ya existente) para registrar entradas/salidas, y agregamos lógica (via triggers o procedimientos almacenados) para calcular costos automáticamente.

1. Extensión de Tablas

Agregamos la tabla **Lotes** para rastrear costos por compra:

Campo	Tipo de dato	Descripción
id_lote	INT (PK)	Identificador único del lote
id_producto	INT (FK)	Referencia al producto
id_bodega	INT (FK)	Referencia a la bodega
id_orden_compra	INT (FK, opcional)	Referencia a la orden de compra
fecha_compra	DATE	Fecha de la compra
cantidad_inicial	INT	Cantidad comprada inicialmente
cantidad_restante	INT	Cantidad disponible en el lote
costo_unitario	DECIMAL(10,2)	Costo por unidad en esta compra

SQL para crear la tabla Lotes:

sql

```
CREATE TABLE Lotes (
    id_lote INT AUTO_INCREMENT PRIMARY KEY,
    id_producto INT NOT NULL,
    id_bodega INT NOT NULL,
    id_orden_compra INT,
    fecha_compra DATE NOT NULL,
    cantidad_inicial INT NOT NULL,
    cantidad_restante INT NOT NULL,
    costo_unitario DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELETE RESTRICT,
    FOREIGN KEY (id_bodega) REFERENCES Bodegas(id_bodega) ON DELETE RESTRICT,
    FOREIGN KEY (id_orden_compra) REFERENCES OrdenesCompra(id_orden_compra)
);
```

Modificamos **MovimientosInventario** para incluir el lote y el costo aplicado:

sql

```
ALTER TABLE MovimientosInventario
ADD COLUMN id_lote INT,
ADD COLUMN costo_unitario_aplicado DECIMAL(10,2),
ADD FOREIGN KEY (id_lote) REFERENCES Lotes(id_lote) ON DELETE SET NULL;
```

2. Proceso de Implementación de FIFO

- **En Compras (Entradas):**
 - Al registrar una compra en `OrdenesCompra`, inserta un nuevo registro en `Lotes` con la cantidad comprada, costo unitario y fecha.
 - Inserta un movimiento de 'ENTRADA' en `MovimientosInventario`, actualizando el stock total en `Inventario`.
 - El costo promedio en `Productos` o `Inventario` puede actualizarse opcionalmente, pero para FIFO no es necesario (solo se usa para reportes generales).
- **En Ventas (Salidas):**
 - Al facturar (insertar en `LíneaFactura`), identifica el lote más antiguo (menor `fecha_compra`) con `cantidad_restante > 0` para el producto y bodega.
 - Reduce `cantidad_restante` en ese lote por la cantidad vendida.
 - Registra el movimiento de 'SALIDA' en `MovimientosInventario` con el `id_lote` y `costo_unitario_aplicado` del lote seleccionado.
 - Si un lote se agota, pasa al siguiente más antiguo (puede requerir un loop en un procedimiento).

Ejemplo de Procedimiento Almacenado para Registrar una Salida (Venta) con FIFO:

sql

```
DELIMITER //
CREATE PROCEDURE RegistrarSalidaFIFO (
    IN p_id_producto INT,
    IN p_id_bodega INT,
    IN p_cantidad INT,
    IN p_id_factura INT
)
BEGIN
    DECLARE v_id_lote INT;
    DECLARE v_costo_unitario DECIMAL(10,2);
    DECLARE v_cantidad_restante INT;
    DECIMAL v_cantidad_pendiente TNT DFFAIJI T n cantidad:
```

```

WHILE v_cantidad_pendiente > 0 DO
    -- Encontrar el lote más antiguo con stock
    SELECT id_lote, costo_unitario, cantidad_restante
    INTO v_id_lote, v_costo_unitario, v_cantidad_restante
    FROM Lotes
    WHERE id_producto = p_id_producto AND id_bodega = p_id_bodega A
    ORDER BY fecha_compra ASC
    LIMIT 1;

    IF v_id_lote IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Stock insuficiente';
    END IF;

    -- Calcular cantidad a despachar de este lote
    SET @despacho = LEAST(v_cantidad_restante, v_cantidad_pendiente);

    -- Actualizar lote
    UPDATE Lotes SET cantidad_restante = cantidad_restante - @despacho;

    -- Registrar movimiento
    INSERT INTO MovimientosInventario (id_producto, id_bodega, id_factura, tipo_movimiento)
    VALUES (p_id_producto, p_id_bodega, p_id_factura, 'SALIDA', @despacho);

    -- Actualizar stock total
    UPDATE inventario SET cantidad_stock = cantidad_stock - @despacho;

    SET v_cantidad_pendiente = v_cantidad_pendiente - @despacho;
END WHILE;
END //;
DELIMITER ;

```

3. Reporte de Costos y Valoración

Para generar reportes (como el valor del inventario o COGS), consulta los lotes restantes:

sql

```

SELECT p.nombre_producto, SUM(l.cantidad_restante * l.costo_unitario) AS COGS
FROM Lotes l
JOIN Productos p ON l.id_producto = p.id_producto
WHERE l.cantidad_restante > 0
GROUP BY p.id_producto;

```

4. Implementación de LIFO (No Recomendada)

Si por alguna razón se opta por LIFO (a pesar de no ser permitido bajo NIIF), la implementación es similar: Cambia el `ORDER BY fecha_compra ASC` a `DESC` en el procedimiento para seleccionar el lote más reciente primero. Sin embargo, esto requeriría ajustes contables separados para fines fiscales vs. financieros, lo que complica el sistema.

Esta extensión mantiene la integridad del modelo relacional, asegura trazabilidad para auditorías del SRI y permite cálculos precisos de rentabilidad. Si necesitas scripts para triggers en compras o ejemplos con datos, indícame.

Más ejemplos con datos de prueba

Sin respuesta.



No se puede confirmar la autenticidad del usuario.

Por favor, inicia sesión o actualiza la página

