

Resumen del modelo de datos

Base de datos simple para gestionar clientes, productos y pedidos. Contiene cuatro tablas relacionadas: Cliente, Producto, Pedido y DetallePedido. El diseño guarda información de clientes y productos, registra pedidos por cliente y los ítems (detalles) de cada pedido con sus cantidades y precios unitarios.

Esquema y propósito de cada tabla

Cliente

- **Propósito:** almacenar información de los clientes que realizan pedidos.
- **Columnas:**
 - **cliente_id** INT PRIMARY KEY — identificador único del cliente.
 - **nombre** VARCHAR(100) NOT NULL — nombre del cliente.
 - **direccion** VARCHAR(200) — dirección postal.
 - **telefono** VARCHAR(20) — teléfono de contacto.
 - **correo** VARCHAR(100) — correo electrónico.

Producto

- **Propósito:** catálogo de productos disponibles para vender.
- **Columnas:**
 - **producto_id** INT PRIMARY KEY — identificador único del producto.
 - **nombre** VARCHAR(100) NOT NULL — nombre del producto.
 - **descripcion** TEXT — descripción detallada.
 - **precio** DECIMAL(10,2) NOT NULL — precio unitario actual del producto.

Pedido

- **Propósito:** registra pedidos realizados por clientes.
- **Columnas:**
 - **pedido_id** INT PRIMARY KEY — identificador único del pedido.
 - **fecha** DATE NOT NULL — fecha del pedido.
 - **cliente_id** INT NOT NULL — referencia al cliente que hizo el pedido.

- **estado** VARCHAR(20) — estado del pedido (ejemplo: pendiente, enviado, cancelado).
- **Restricciones:**
 - **FOREIGN KEY (cliente_id)** REFERENCES Cliente(cliente_id) — integridad referencial hacia Cliente.

DetallePedido

- **Propósito:** líneas de cada pedido; relaciona pedidos con productos y cantidades.
- **Columnas:**
 - **detalle_id** INT PRIMARY KEY — identificador único de la línea.
 - **pedido_id** INT NOT NULL — referencia al pedido.
 - **producto_id** INT NOT NULL — referencia al producto.
 - **cantidad** INT NOT NULL — unidades pedidas.
 - **precio_unitario** DECIMAL(10,2) NOT NULL — precio unitario aplicado en el pedido.
- **Restricciones:**
 - **FOREIGN KEY (pedido_id)** REFERENCES Pedido(pedido_id).
 - **FOREIGN KEY (producto_id)** REFERENCES Producto(producto_id).

Reglas de integridad y comportamiento esperado

- Las claves primarias garantizan unicidad por fila en cada tabla.
- Las claves externas impiden que exista un pedido sin cliente y una línea sin pedido o producto.
- El **precio_unitario** en DetallePedido captura el precio al momento del pedido, protegiendo el histórico frente a cambios futuros en Producto.precio..
- No hay acciones ON DELETE o ON UPDATE especificadas; por defecto la base puede bloquear o impedir borrados que rompan la integridad. Se recomienda definir políticas explícitas (CASCADE, SET NULL, RESTRICT) según la lógica de negocio.

Normalización y observaciones de diseño

- El modelo está en 3NF razonable:

- No hay columnas repetidas ni grupos multivaluados.
- Las dependencias funcionales principales se respetan: atributos dependen de la clave primaria de su propia tabla.
- Observaciones y mejoras posibles:
 - Usar secuencias o AUTO_INCREMENT para las claves primarias para evitar manejo manual de IDs.
 - Añadir un índice sobre Pedido(cliente_id) y DetallePedido(pedido_id) para consultas frecuentes.
 - Considerar una tabla EstadoPedido (id, nombre) si se requiere controlar estados y evitar valores libres en texto.
 - Validar que cantidad > 0 mediante CHECK si el SGBD lo soporta.
 - Añadir timestamps (created_at, updated_at) en tablas de transacciones para auditoría.

Índices y performance recomendada

- Índices sugeridos:
 - PRIMARY KEY en cada tabla (ya definidos).
 - INDEX idx_pedido_cliente ON Pedido(cliente_id).
 - INDEX idx_detalle_pedido ON DetallePedido(pedido_id).
 - INDEX idx_detalle_producto ON DetallePedido(producto_id) si se buscan ventas por producto.
- Considerar índices compuestos si se realizan consultas filtrando por múltiples columnas.