

Sistema de gestión de ventas.

Tablas: Productos, clientes y ordenes de compras.

Justificación del Diseño de la Base de Datos

Introducción

El diseño de la base de datos para un sistema de ventas en línea es fundamental para garantizar la integridad, eficiencia y escalabilidad del sistema. En este documento, se presenta la justificación del diseño de la base de datos, explicando cómo se alcanzó la **Tercera Forma Normal (3NF)** y la importancia de las decisiones tomadas en el diseño.

1. Entidades y Atributos

1.1. Identificación de Entidades

Las entidades principales identificadas son:

- **Productos:** Representa los artículos disponibles para la venta.
- **Clientes:** Representa los compradores que realizan órdenes.
- **Órdenes:** Registra los pedidos realizados por los clientes.

1.2. Definición de Atributos

Cada entidad tiene los atributos mínimos necesarios para almacenar datos relevantes:

Productos

- **product_id** (PK): Identificador único del producto.
- **name**: Nombre del producto.
- **category**: Categoría a la que pertenece el producto.
- **price**: Precio unitario.
- **stock**: Cantidad disponible en inventario.

Clientes

- **client_id** (PK): Identificador único del cliente.
- **nombre**: Nombre del cliente.
- **correo**: Dirección de envío.
- **telefono**: Número telefónico.

Órdenes

- **order_id** (PK): Identificador único de la orden.
- **client_id** (FK): Cliente asociado con la orden.
- **product_id** (FK): Producto incluido en la orden.
- **fecha**: Fecha en que se realizó la orden.
- **quantity**: Cantidad de productos solicitados.

2. Normalización

2.1. Primera Forma Normal (1NF)

Para alcanzar la 1NF:

- Cada tabla tiene un identificador único (llave primaria).
- Todos los atributos contienen valores atómicos (no repetidos ni compuestos).
- No existen grupos repetitivos.

Ejemplo aplicado: En la tabla **Órdenes**, cada registro incluye un solo producto con una cantidad específica y no contiene listas de productos o cantidades.

2.2. Segunda Forma Normal (2NF)

Para alcanzar la 2NF:

- Se eliminaron dependencias parciales: Todos los atributos no clave dependen completamente de la clave primaria.

Ejemplo aplicado: En la tabla **Órdenes**, **client_id** y **product_id** dependen completamente de la clave primaria **order_id**. No existen atributos que dependan solo de una parte de la clave.

2.3. Tercera Forma Normal (3NF)

Para alcanzar la 3NF:

- Se eliminaron dependencias transitivas: Los atributos no clave son independientes entre sí y dependen únicamente de la clave primaria.

Ejemplo aplicado: En la tabla **Cientes**, no se incluye información como **nombre del producto** o **precio**, ya que esto corresponde a la entidad **Productos**. La relación entre tablas se maneja mediante claves foráneas.

3. Relación entre Tablas

Se establecieron las siguientes relaciones:

- **Productos y Órdenes:** Relación 1:N, ya que un producto puede aparecer en múltiples órdenes.
- **Cientes y Órdenes:** Relación 1:N, ya que un cliente puede realizar múltiples órdenes.

Estas relaciones están representadas mediante claves foráneas:

- **client_id** en **Órdenes** referencia a **Cientes(client_id)**.
- **product_id** en **Órdenes** referencia a **Productos(product_id)**.

4. Restricciones de Integridad

Para mantener la consistencia de los datos:

- **Llaves primarias:** Garantizan unicidad de registros.
- **Llaves foráneas:** Aseguran consistencia referencial entre tablas.
- **Restricciones adicionales:**

- **NOT NULL:** Campos como **name**, **price**, y **quantity** no pueden estar vacíos.
- **UNIQUE:** El correo electrónico de los clientes debe ser único.
- **ON DELETE RESTRICT:** Impide la eliminación de registros en **Clientes** o **Productos** si están referenciados en **Órdenes**.
- **ON UPDATE CASCADE:** Permite actualizar claves foráneas si cambian en la tabla referenciada.

5. Beneficios del Diseño

1. **Reducción de Redundancia:** La normalización elimina duplicados, como almacenar el nombre del producto en varias órdenes.
2. **Escalabilidad:** Las relaciones bien definidas permiten agregar más entidades, como proveedores o promociones.
3. **Mantenimiento:** El uso de claves foráneas y restricciones facilita mantener la integridad de los datos.
4. **Eficiencia:** La creación de índices en atributos como **product_id** y **client_id** optimiza las consultas más frecuentes.

Conclusión

El diseño de esta base de datos cumple con los principios de la 3NF, garantizando un sistema eficiente, libre de redundancias y preparado para futuras ampliaciones. Además, las relaciones entre las entidades permiten consultas complejas mientras se mantiene la integridad de los datos.

