

1. Introducción

El objetivo del diseño es construir una base de datos relacional que gestione clientes, productos y órdenes de compra de manera eficiente, evitando redundancias y asegurando la integridad de los datos. Se aplicaron las reglas de normalización hasta la Tercera Forma Normal (3FN), garantizando que cada tabla represente una sola entidad y que los atributos dependan únicamente de la clave primaria.

2. Proceso de Normalización

1FN (Primera Forma Normal)

- Todos los atributos son atómicos (no se almacenan listas ni valores compuestos).
- Cada tabla tiene una clave primaria definida.
- Ejemplo: en la tabla clientes, el campo direccion se descompuso en una tabla aparte para evitar valores compuestos.

2FN (Segunda Forma Normal)

- Se eliminaron dependencias parciales.
- Ejemplo: en productos, la categoría no se almacena como texto repetido, sino como clave foránea hacia la tabla categorías.
- Esto evita duplicación de nombres de categorías y facilita su gestión.

3FN (Tercera Forma Normal)

- Se eliminaron dependencias transitivas.
- Ejemplo: en órdenes, el estado no se almacena como texto dentro de la tabla, sino como clave foránea hacia estados_orden.
- Esto permite modificar o ampliar los estados sin alterar la estructura de la tabla principal.

3. Modelo de Datos en 3FN

1. Entidad Clientes

- id_cliente (PK)
- nombre
- apellido
- email (único)
- telefono
- fecha_registro
- activo (booleano)

Relación: Direcciones

Cada cliente puede tener una o varias direcciones.

- id_direccion (PK)
- id_cliente (FK → Clientes)
- calle
- ciudad
- provincia
- codigo_postal

2. Entidad Categorías

- id_categoria (PK)
- nombre_categoria (único)

3. Entidad Productos

- id_producto (PK)
- nombre
- id_categoria (FK → Categorías)
- precio
- stock
- fecha_alta
- activo (booleano)

4. Entidad Estados de Orden

- id_estado (PK)
- nombre_estado (ej.: PENDIENTE, PAGADA, CANCELADA)

5. Entidad Órdenes

- id_orden (PK)
- id_cliente (FK → Clientes)
- fecha_orden
- id_estado (FK → Estados de Orden)

6. Entidad Detalle de Órdenes

Relaciona productos con órdenes (N:M).

- id_detalle (PK)
- id_orden (FK → Órdenes)
- id_producto (FK → Productos)
- cantidad
- precio_unitario

4. Justificación del Diseño

(id_orden, fecha_orden, estado, id_cliente, nombre_cliente, apellido_cliente, email, telefono, direccion, fecha_registro, id_producto, nombre_producto, categoria, precio, stock, fecha_alta, cantida

d,precio_unitario,subtotal)

A. Dependencias del Cliente

- id_cliente → nombre_cliente, apellido_cliente, email, telefono, direccion, fecha_registro

B. Dependencias del Producto

- id_producto → nombre_producto, categoria, precio, stock, fecha_alta

C. Dependencias de la Orden

- id_orden → id_cliente, fecha_orden, estado

D. Dependencias del Detalle

- (id_orden, id_producto) → cantidad, precio_unitario, subtotal

Clientes y Direcciones: separación para permitir múltiples direcciones por cliente y evitar redundancia.

Productos y Categorías: separación para evitar repetición de nombres de categorías y facilitar su administración.

Órdenes y Estados: separación para que los estados sean gestionables y extensibles sin modificar la tabla principal.

Detalle de Órdenes: necesaria para representar la relación N:M entre órdenes y productos, evitando duplicación de datos.