La 1FN ya esta hecha, es la base de datos que hiciste porque no tiene campos o datos duplicados, o sea no existe la duplicidad,

Todos los atributos contienen valores atómicos y cada fila es única

La 2FN

Para que una tabla esté en 2FN, debe estar en 1FN y todos los atributos no clave deben depender completamente de la clave primaria. Esto implica que si alguna columna depende solo de una parte de la clave primaria, se debe mover a otra tabla.

Cambios de la 2FN

Cambios propuestos:

* Carritos y Carrito1\_Items tienen atributos que dependen de id\_usuario y id\_producto, pero también se repiten. Vamos a consolidar estas tablas.

Tablas Modificadas:

CREATE TABLE Carritos (

id\_carrito INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT NOT NULL,

fecha\_agregado TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

CREATE TABLE Carrito\_Items (

id\_item INT AUTO\_INCREMENT PRIMARY KEY,

id\_carrito INT NOT NULL,

id\_producto INT NOT NULL,

cantidad INT NOT NULL CHECK (cantidad > 0),

FOREIGN KEY (id\_carrito) REFERENCES Carritos(id\_carrito),

FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto)

);

La 3FN

Para que una tabla esté en 3FN, debe estar en 2FN y no debe haber dependencias transitivas (es decir, un atributo no clave no debe depender de otro atributo no clave).

Cambios propuestos:

* La tabla Usuarios tiene atributos que podrían ser considerados como dependencias transitivas. Sin embargo, no hay atributos no clave que dependan de otros atributos no clave, así que esta tabla está bien.

La tabla ordenes tiene un atributo total que se puede calcular a partir de detalle\_ordenes. Para mantener la normalización, podríamos eliminar el campo total de la tabla ordenes y calcularlo cuando sea necesario.

Tablas modificadas:

CREATE TABLE ordenes (

id\_orden INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

estado ENUM('pendiente', 'pagado', 'enviado', 'entregado', 'cancelado') DEFAULT 'pendiente',

fecha\_orden TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

La 4FN

Para que una tabla esté en 4FN, debe estar en 3FN y no debe haber dependencias multivaluadas (es decir, dos o más atributos no clave que dependen de la clave primaria).

Sin embargo, si consideramos la tabla memebrsías, podríamos tener múltiples beneficios y es necesario tener una tabla para ello

Tablas:

CREATE TABLE Membresias (

id\_membresia INT AUTO\_INCREMENT PRIMARY KEY,

tipo VARCHAR(50) NOT NULL,

precio DECIMAL(10,2) NOT NULL

);

CREATE TABLE Beneficios (

id\_beneficio INT AUTO\_INCREMENT PRIMARY KEY,

id\_membresia INT NOT NULL,

descripcion TEXT,

FOREIGN KEY (id\_membresia) REFERENCES Membresias(id\_membresia)

);

Estructura de la base de datos final normalizada:

CREATE TABLE Usuarios (

id\_usuario INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

apellido VARCHAR(100) NOT NULL,

correo VARCHAR(100) UNIQUE NOT NULL,

contraseña VARCHAR(255) NOT NULL,

direccion TEXT,

telefono VARCHAR(15),

fecha\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE Productos (

id\_producto INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

descripcion TEXT,

precio DECIMAL(10,2) NOT NULL,

stock INT NOT NULL,

categoria VARCHAR(50)

);

CREATE TABLE Carritos (

id\_carrito INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT NOT NULL,

fecha\_agregado TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

CREATE TABLE Carrito\_Items (

id\_item INT AUTO\_INCREMENT PRIMARY KEY,

id\_carrito INT NOT NULL,

id\_producto INT NOT NULL,

cantidad INT NOT NULL CHECK (cantidad > 0),

FOREIGN KEY (id\_carrito) REFERENCES Carritos(id\_carrito),

FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto)

);

CREATE TABLE ordenes (

id\_orden INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

estado ENUM('pendiente', 'pagado', 'enviado', 'entregado', 'cancelado') DEFAULT 'pendiente',

fecha\_orden TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

CREATE TABLE detalle\_ordenes (

id\_detalle INT AUTO\_INCREMENT PRIMARY KEY,

id\_orden INT,

id\_producto INT,

cantidad INT,

precio\_unitario DECIMAL(10,2),

FOREIGN KEY (id\_orden) REFERENCES ordenes(id\_orden),

FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto)

);

CREATE TABLE resenas (

id\_resena INT AUTO\_INCREMENT PRIMARY KEY,

id\_producto INT,

id\_usuario INT,

calificacion INT CHECK (calificacion BETWEEN 1 AND 5),

comentario TEXT,

fecha\_resena TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto),

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

CREATE TABLE lista\_deseos (

id INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

id\_producto INT,

fecha\_agregado TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario),

FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto)

);

CREATE TABLE Membresias (

id\_membresia INT AUTO\_INCREMENT PRIMARY KEY,

tipo VARCHAR(50) NOT NULL,

precio DECIMAL(10,2) NOT NULL

);

CREATE TABLE Beneficios (

id\_beneficio INT AUTO\_INCREMENT PRIMARY KEY,

id\_membresia INT NOT NULL,

descripcion TEXT,

FOREIGN KEY (id\_membresia) REFERENCES Membresias(id\_membresia)

);