

Taller 3: Normalización de Bases de datos - Forma Normal

Camilo Esteban Paez - 1110450891

Problemas Identificados

La tabla **productos** tiene las siguiente columnas: **PedidoID** | **ClienteID** | **ProductoID** | **Cantidad** | **Fecha** | **MetodoPago**, **PedidoID** es su PRIMARY KEY, **ClienteID** | **ProductoID** son sus FOREIGN KEYS que permiten la no repetición de la información tanto de los clientes como del producto, **Cantidad** es un INT lo cual es tal-vez algo excesivo, **Fecha** es un DATE y **MetodoPago** es un VARCHAR(50) aquí también hay un problema.

No relacionados a la normalización

- **Cantidad** tiene tipo INT firmado, al ser una tienda minorista no es necesario una cantidad posible tan grande, ademas no es posible pedir una cantidad negativa

Relacionados con la normalizacion

- **MetodoPago** se comporta como un Enum pero esta definido como string, se puede extraer a otra tabla, compuesta por las columnas **MetodoPagoId** | **descripcion** o en su defecto usar el datatype **Enum**.
- **Categoria** se comporta como un Enum pero esta definido como string, se puede extraer a otra tabla, compuesta por las columnas **Categoria** | **descripcion** o en su defecto usar el datatype **Enum**.
- Direcciones podria extraerse a una tabla .

Cambios realizados y nueva estructura

Atomizacion

La base de datos esta suficientemente atomizada, se podria separar **Nombre** aunque no lo considero necesario.

Eliminar valores multivaluados y repetidos

- Creacion tabla Categoria

Categoria pasara a residir en una tabla donde contará con una ID y una descripcion, esto permitira que se evite redundancia en la tabla **Productos**

```
create table Categoria (  
  CategoriaID INT AUTO_INCREMENT PRIMARY KEY,  
  descripcion VARCHAR(50)  
);
```

- Creacion tabla MetodoPago

MetodoPago pasara a residir en una tabla donde contará con una ID y una descripcion, esto permitira que se evite redundancia en la tabla **MetodoPago**

```
create table MetodoPago (  
  MetodoPagoID INT AUTO_INCREMENT PRIMARY KEY,  
  descripcion VARCHAR(50)  
);
```

- Nueva tabla Productos

```
create table Productos (  
  ProductoID INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre VARCHAR(100),  
  CategoriaID INT, -- <----- Nuevo Tipo  
  Precio DECIMAL(10, 2),  
  FOREIGN KEY (CategoriaID) REFERENCES Categoria(CategoriaID) -- <-----  
  Nueva Foreign Key  
);
```

- Nueva tabla Pedidos

```
create table Pedidos (  
  PedidoID INT AUTO_INCREMENT PRIMARY KEY,  
  ClienteID INT,  
  ProductoID INT,  
  Cantidad INT,  
  Fecha DATE,  
  MetodoPagoID INT, -- <----- Nuevo Tipo  
  FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID),  
  FOREIGN KEY (ProductoID) REFERENCES Productos(ProductoID),  
  FOREIGN KEY (MetodoPagoID) REFERENCES MetodoPago(MetodoPagoID) -- <-----  
  - Nueva Foreign Key  
);
```

Clientes se mantiene con su misma estructura

```
create table Clientes (  
  ClienteID INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre VARCHAR(100),  
  Direccion VARCHAR(150),  
  Telefono VARCHAR(15)  
);
```

Capturas

```
<> setup ● <> correccion x <> datos Categoria [all] Clientes [all] MetodoPago [all] Productos [all]

1 create table Categoria (
2     CategoriaID INT AUTO_INCREMENT PRIMARY KEY,
3     descripcion VARCHAR(50)
4 );
5
6 create table Clientes (
7     ClienteID INT AUTO_INCREMENT PRIMARY KEY,
8     Nombre VARCHAR(100),
9     Direccion VARCHAR(150),
10    Telefono VARCHAR(15)
11 );
12
13 create table MetodoPago (
14     MetodoPagoID INT AUTO_INCREMENT PRIMARY KEY,
15     descripcion VARCHAR(50)
16 );
17
18 create table Productos (
19     ProductoID INT AUTO_INCREMENT PRIMARY KEY,
20     Nombre VARCHAR(100),
21     CategoriaID INT,
22     Precio DECIMAL(10, 2),
23     FOREIGN KEY (CategoriaID) REFERENCES Categoria(CategoriaID)
24 );
25
26 create table Pedidos (
27     PedidoID INT AUTO_INCREMENT PRIMARY KEY,
28     ClienteID INT,
29     ProductoID INT,
30     Cantidad INT,
31     Fecha DATE,
32     MetodoPagoID INT,
33     FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID),
34     FOREIGN KEY (ProductoID) REFERENCES Productos(ProductoID),
35     FOREIGN KEY (MetodoPagoID) REFERENCES MetodoPago(MetodoPagoID)
36 );
```

```
<> setup ● <> correccion <> datos x Categoria [all] Clientes [all] MetodoPago [all] Productos [all] Pedido

1 INSERT INTO Clientes (Nombre, Direccion, Telefono) VALUES
2 ('Juan Pérez', 'Calle 123', '3101234567'),
3 ('María Gómez', 'Avenida 45', '3129876543'),
4 ('Carlos Ruiz', 'Carrera 67', '3152345678'),
5 ('Luisa López', 'Diagonal 89', '3203456789'),
6 ('Pedro Martínez', 'Calle 45', '3104567890'),
7 ('Ana Torres', 'Calle 78', '3145678901'),
8 ('Jorge Castillo', 'Carrera 11', '3167890123');
9
10 insert into Categoria(descripcion) values
11 ('Calzado'),
12 ('Ropa');
13
14 INSERT INTO Productos (Nombre, CategoriaID, Precio) VALUES
15 ('Zapatos', 1, 60.00),
16 ('Camisa', 2, 30.00),
17 ('Pantalón', 2, 50.00);
18
19 insert into MetodoPago(descripcion) VALUES
20 ('Tarjeta de crédito'),
21 ('Efectivo'),
22 ('Transferencia bancaria'),
23 ('Tarjeta débito');
24
25 insert into Pedidos (ClienteID, ProductoID, Cantidad, Fecha, MetodoPagoID) VALUES
26 (1, 1, 2, '2024-12-01', 1),
27 (1, 2, 1, '2024-12-01', 1),
28 (1, 3, 1, '2024-12-01', 1),
29 (2, 1, 1, '2024-12-02', 2),
```

```
30 (3, 2, 3, '2024-12-02', 3),
31 (3, 1, 2, '2024-12-03', 3),
32 (4, 3, 2, '2024-12-03', 2),
33 (4, 1, 1, '2024-12-04', 2),
34 (5, 2, 2, '2024-12-04', 4),
35 (5, 1, 1, '2024-12-04', 4),
36 (5, 3, 1, '2024-12-05', 4),
37 (6, 1, 2, '2024-12-05', 2),
38 (6, 2, 1, '2024-12-06', 2),
39 (7, 1, 3, '2024-12-06', 1),
40 (7, 3, 2, '2024-12-07', 1);
```

bd24 - Beekeeper S

<> setup ● <> correccion <> datos

Categoria [all] x Clientes [all]

<> CategoriaID equals Enter Value

🔑 CategoriaID int ▲

descripcion varchar(50) ▲

1	1	Calzado
2	2	Ropa

<> setup ● <> correccion <> datos

Categoria [all] Clientes [all] x MetodoPa

<> ClienteID equals Enter Value

🔑 ClienteID int ▲

Nombre varchar(100) ▲

Direccion varchar(150) ▲

Telefono varchar(15) ▲

1	1	Juan Pérez	Calle 123	3101234567
2	2	María Gómez	Avenida 45	3129876543
3	3	Carlos Ruiz	Carrera 67	3152345678
4	4	Luisa López	Diagonal 89	3203456789
5	5	Pedro Martínez	Calle 45	3104567890
6	6	Ana Torres	Calle 78	3145678901
7	7	Jorge Castillo	Carrera 11	3167890123

<> setup ● <> correccion <> datos Categoria [all] Clientes [all]

<> MetodoPagoID equals Enter Value

	MetodoPagoID int ▲	descripcion varchar(50) ▲
1	1	Tarjeta de crédito
2	2	Efectivo
3	3	Transferencia bancaria
4	4	Tarjeta débito

<> setup ● <> correccion <> datos Categoria [all] Clientes [all] MetodoPago [all] Productos [all]

<> PedidoID equals Enter Value

	PedidoID int ▲	ClienteID int ▲	ProductoID int ▲	Cantidad int ▲	Fecha date ▲	MetodoPagoID int ▲
1	1	1	1	2	2024-12-01	1
2	2	1	2	1	2024-12-01	1
3	3	1	3	1	2024-12-01	1
4	4	2	1	1	2024-12-02	2
5	5	3	2	3	2024-12-02	3
6	6	3	1	2	2024-12-03	3
7	7	4	3	2	2024-12-03	2
8	8	4	1	1	2024-12-04	2
9	9	5	2	2	2024-12-04	4
10	10	5	1	1	2024-12-04	4
11	11	5	3	1	2024-12-05	4
12	12	6	1	2	2024-12-05	2
13	13	6	2	1	2024-12-06	2
14	14	7	1	3	2024-12-06	1
15	15	7	3	2	2024-12-07	1

<> setup ● <> correccion <> datos

Categoria [all] Clientes [all] MetodoPago [all]

<>	ProductID	equals	Enter Value	
	ProductID int ▲	Nombre varchar(100) ▲	CategoriID int ▲	Precio decimal(10,2) ▲
1	1	Zapatos	1	60.00
2	2	Camisa	2	30.00
3	3	Pantalón	2	50.00

Conclusiones

La normalizacion es esencial en las bases de datos relacionales ya que sacan partido de las propias características de esta, permiten que ademas de eficiente se mantenga mantenible a lo largo del tiempo.