

# MANUAL

la base de datos ha sido diseñado para gestionar de manera eficiente las operaciones de ventas e inventario de una tienda, con la capacidad de escalar a múltiples sucursales, tanto dentro como fuera de la ciudad, su estructura permite un crecimiento flexible y una administración centralizada de los datos.

## ¿Qué es una tabla?

Una tabla es una entidad la cual guarda registros tales como una factura, las categorías disponibles o los productos

- **category:** informacion de las categorias de los productos.
- **documenttype:** informacion de los tipos de identificacion usada por los diferentes usuarios.
- **invoice:** informacion de facturas generadas.
- **invoiceproduct:** productos correspondientes a las facturas generadas.
- **invoicestatus:** informacion de los estados de las facturas, “cancelado”, “pagado”....
- **order:** informacion de envios de productos correspondientes a una factura.
- **orderstatus:** estado del envio correspondiente al envio.
- **paymentgatewayrecords:** respuesta del servicio de pagos.
- **paymentmethod:** información de los métodos de pago.
- **people:** información de todas las personas que hacen parte de la tienda, sea como cliente, como vendedor o como proveedor.
- **product:** información de productos de la tienda.
- **productcategory:** categorías correspondientes a su producto.
- **productdesc:** productos los cuales tienen descuento.
- **productsstore:** productos los cuales pertenecen a determinada sucursal, por lo general a la principal ya que es de una sola tienda, antes de escalar.
- **productsupplier:** proveedor de determinado productos.
- **roles:** roles de los usuarios(e.g., customer, admin).
- **store:** sucursales de la tienda(como minimo 1, la principal)
- **user:** usuarios de la tienda, es el encargado de relacionar con todas las entidades, pero solo en el caso del vendedor almacenara valores como email(user), password, para acceder al sistema.
- **userroles:** los roles de los usuarios(un usuario puede tener mas de un rol).

## RELACIONES DE TABLAS(DEV):

- **UNO A MUCHOS:**

- category Y productcategory (una categoria puede tener muchos productos).
- documenttype y people (un tipo de documento puede tener muchas personas).
- invoice y invoiceproduct (una factura puede tener muchos productos).
- invoicestatus y invoice (una factura puede tener un solo estado, pero un estado tiene muchas facturas).
- orderstatus y order (una orden puede tener un estado, pero un estado puede tener muchas ordenes).
- paymentmethod y invoice (un metodo de pago puede tener muchas facturas, pero una factura un solo método de pago).

- **Uno a uno:**

- invoice y order (una factura puede tener un envio o una orden).
- invoice y paymentgatewayrecords (un pago por servicio externo de transferencia puede tener una sola respuesta de pago).
- people and user (una persona puede tener un usuario).
- product y productdesc (un producto puede tener un descuento).

- **Muchos a muchos:**

- Roles, user y userroles (un usuario puede tener muchos roles y un rol puede tener muchos usuarios)
- Producto, supplier y productsupplier (un producto puede tener muchos proveedores y un proveedor muchos productos).
- Producto, category y productcategory (un producto puede tener muchas categorias y una categoría muchos productos).
- Product, store y productsstore (un producto puede estar en múltiples tiendas y una tienda tiene múltiples productos).

## CONSULTAS SIMPLES:

1. Buscar productos cuando su categoría sea 'granos'  
"SELECT p.name  
FROM product AS p  
INNER JOIN productcategory AS pc ON p.id = pc.productId  
INNER JOIN category AS c ON pc.categoryId = c.id  
WHERE c.name = 'granos'"

2. Busca todas las facturas correspondientes a un usuario:

```
"SELECT i.id, i.createdAt
FROM invoice AS i
WHERE i.userId = 1"
```

3. Buscar los 5 productos mas vendidos:

```
"SELECT p.name, SUM(ip.cant) AS total_sold
FROM product p
INNER JOIN invoiceproduct AS ip ON p.id = ip.productId
GROUP BY p.name
ORDER BY total_sold DESC
LIMIT 5"
```

4. Buscar productos que tienen descuento:

```
"SELECT p.name, pd.discount
FROM product p
INNER JOIN productdesc AS pd ON p.id = pd.productId
WHERE pd.discount > (
    SELECT AVG(discount)
    FROM productdesc
)"
```

5. Buscar producto que no se han vendido en el ultimo mes:

```
"SELECT p.name
FROM product AS p
WHERE p.id NOT IN (
    SELECT ip.productId
    FROM invoiceproduct AS ip
    INNER JOIN invoice AS i ON ip.invoiceId = i.id
    WHERE i.createdAt > DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
)"
```

6. Buscar productos cuando su precio es mayor al promedio de los productos en categoría "granos":

```
"SELECT p.name, p.price
FROM product AS p
WHERE p.price > (
    SELECT AVG(p2.price)
    FROM product AS p2
    INNER JOIN productcategory pc ON p2.id = pc.productId
    INNER JOIN category c ON pc.categoryId = c.id
    WHERE c.name = 'granos'
)"
```

## Flujo de Información en la Base de Datos

La base de datos está diseñada para gestionar de manera eficiente las operaciones de venta e inventario de una tienda. A continuación se muestra cómo fluye la información:

### 1. Gestión de Productos:

- Se agregan productos a la base de datos, clasificándolos en categorías.
- Se detalla información sobre los productos, como descuentos y descripciones.
- Se asigna la disponibilidad de productos a diferentes tiendas.
- Se registran los proveedores de cada producto.

### 2. Gestión de Clientes:

- Se almacenan los datos personales de los clientes.
- Se crea un usuario, solo si su rol es vendedor se registrara un usuario y una contraseña de lo contrario quedara como null.

### 3. Ventas y Facturación:

- Los clientes realizan pedidos.
- Se generan facturas con los productos seleccionados y sus cantidades.
- Se registra el método de pago utilizado.
- Se actualiza el estado de las facturas (pagadas, canceladas, etc.).
- Se almacenan los detalles de los pagos realizados.

### Flujo de una Venta:

#### 1. Un cliente realiza un pedido:

- Se crea un nuevo registro del pedido.

#### 2. Se procesa el pedido:

- Se seleccionan los productos del inventario.
- Se registran los detalles del pedido, como cantidades y precios.

#### 3. Se genera una factura:

- Se crea una nueva factura, vinculándola a un pedido si es necesario y especificando el total, método de pago y cliente.

#### 4. Se procesa el pago:

- Se registra el método de pago utilizado.
- Se almacena la información del proceso de pago.

5. **Se actualiza el estado de la factura:**

- Se marca la factura como pagada o cancelada.

**Flujo de una orden de Envío;**

1. Se genera un orden de envío.
2. Se debe asignar un estado al envío
3. Se actualiza el estado de la orden de estado del envío.

**Gestión de productos y categorías:**

1. Se crea una nueva categoría única
2. Se crea un nuevo producto al cual se lo asigna a la nueva categoría
3. Si el producto puede permanecer a mas de una categoría, se las puede agregar

