**Diseño del Modelo de Datos para nuestra Tienda de Ropa**

El modelo de datos define la estructura de la base de datos. Piénsalo como el PLANO DE UNA CASA antes de construirla. En este plano, identificamos:

**Entidades (Tablas):** Son los objetos principales sobre los que queremos almacenar información. Nuestras entidades principales serán:

* Categories (Categorías)
* Products (Productos)
* Customers (Clientes)
* Orders (Pedidos)
* Order\_Items (Ítems del Pedido)

**Atributos (Columnas):** Son las características o propiedades de cada entidad. Por ejemplo, la entidad Products tendrá atributos como name (nombre), price (precio), stock (existencias), etc.

**Relaciones:** Son las conexiones lógicas entre las entidades. Por ejemplo, un Product pertenece a una Category, y un Order es realizado por un Customer.

* **Entidad: Categories (Categorías)**
  + **Propósito:** Almacenar información sobre las diferentes categorías de productos (ej. "Camisetas", "Pantalones", "Accesorios").
  + **Atributos:**
    - id (PRIMARY KEY): Identificador único para cada categoría.
    - name (VARCHAR): Nombre de la categoría (ej. "Camisetas"). Debe ser único.
    - description (TEXT): Descripción opcional de la categoría.
* **Entidad: Products (Productos)**
  + **Propósito:** Almacenar los detalles de cada artículo de ropa que vendemos.
  + **Atributos:**
    - id (PRIMARY KEY): Identificador único para cada producto.
    - name (VARCHAR): Nombre del producto (ej. "Camiseta Básica Algodón").
    - size (VARCHAR): Talla del producto (ej. "S", "M", "L", "XL").
    - color (VARCHAR): Color del producto (ej. "Azul", "Negro").
    - price (DECIMAL): Precio del producto.
    - stock (INT): Cantidad de unidades disponibles en inventario.
    - category\_id (FOREIGN KEY): Relaciona el producto con su categoría. Si la categoría se elimina, este campo se pone a NULL.
    - created\_at (DATETIME): Fecha y hora de creación del registro del producto.
* **Entidad: Customers (Clientes)**
  + **Propósito:** Almacenar la información de nuestros clientes.
  + **Atributos:**
    - id (PRIMARY KEY): Identificador único para cada cliente.
    - name (VARCHAR): Nombre completo del cliente.
    - email (VARCHAR): Correo electrónico del cliente. Debe ser único.
    - phone (VARCHAR): Número de teléfono del cliente.
    - created\_at (DATETIME): Fecha y hora de registro del cliente.
* **Entidad: Orders (Pedidos)**
  + **Propósito:** Almacenar información sobre cada compra realizada por un cliente.
  + **Atributos:**
    - id (PRIMARY KEY): Identificador único para cada pedido.
    - customer\_id (FOREIGN KEY): Relaciona el pedido con el cliente que lo realizó. Si el cliente se elimina, sus pedidos también se eliminan (ON DELETE CASCADE).
    - order\_date (DATETIME): Fecha y hora en que se realizó el pedido.
* **Entidad: Order\_Items (Ítems del Pedido)**
  + **Propósito:** Almacenar los productos específicos y sus cantidades dentro de cada pedido. Esta tabla es crucial porque un pedido puede contener múltiples productos, y un producto puede estar en múltiples pedidos (relación muchos a muchos entre Orders y Products).
  + **Atributos:**
    - id (PRIMARY KEY): Identificador único para cada ítem del pedido.
    - order\_id (FOREIGN KEY): Relaciona el ítem con el pedido o compra al que pertenece. Si el pedido se elimina, sus ítems también se eliminan.
    - product\_id (FOREIGN KEY): Relaciona el ítem con el producto específico. Si el producto se elimina, sus ítems en pedidos también se eliminan.
    - quantity (INT): Cantidad de ese producto específico en el pedido. Debe ser mayor que 0.