## 一、数据库设计思路

我在设计这个数据库时，先分析了系统的业务逻辑：  
系统中只有一个卖家，卖家可以上架商品，每个商品又可能有多个买家来填写购买意向。  
所以我按照“卖家 → 商品 → 买家”的逻辑，设计了三张表：

seller：存放卖家账户信息（用户名、密码、更新时间）

products：存放商品信息（名称、价格、图片、上架状态等）

buyers：存放买家的购买信息（姓名、电话、地址、购买的商品等）

其中 buyers.product\_id 是外键，关联 products.id，  
表示哪个买家购买了哪个商品。  
这样就能实现一对多的关系：一个商品可以被多个买家购买。

## ⚙️ 二、创建数据库

**创建数据库**

CREATE DATABASE single\_shop CHARACTER SET utf8mb4;

USE single\_shop;

**创建卖家表（seller）**

CREATE TABLE seller (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(255),

password VARCHAR(255),

updated\_at DATETIME(6)

);

**创建商品表（products）**

CREATE TABLE products (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255),

description VARCHAR(1000),

price DOUBLE NOT NULL,

image\_url VARCHAR(255),

is\_active BIT(1) NOT NULL,

is\_frozen BIT(1) NOT NULL,

created\_at DATETIME(6),

updated\_at DATETIME(6)

);

**创建买家表（buyers）**

CREATE TABLE buyers (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255),

phone VARCHAR(255),

address VARCHAR(255),

notes VARCHAR(255),

is\_completed BIT(1) NOT NULL,

created\_at DATETIME(6),

product\_id BIGINT,

CONSTRAINT fk\_buyer\_product FOREIGN KEY (product\_id) REFERENCES products(id)

);

**插入一些测试数据**

INSERT INTO seller (username, password, updated\_at) VALUES ('admin', '123456', NOW());

INSERT INTO products (name, description, price, is\_active, is\_frozen, created\_at)VALUES ('手工木杯', '手工制作的木质茶杯', 39.9, b'1', b'0', NOW());

## 🧩 三、表之间的关系说明

表之间通过外键建立关系：

products.id 被 buyers.product\_id 引用；

seller (1) → products (1) → buyers (N)

我这个数据库是自己在 MySQL 里建的，一共有三张表。  
我先根据业务逻辑分析出三类实体：卖家、商品、买家。  
卖家表只存登录信息；商品表存商品详情；买家表存购买信息。  
商品和买家通过外键 product\_id 关联，形成一对多关系。  
表的字段和类型都是根据系统功能需要设计的，  
创建完成后我还插入了一些测试数据，方便前端联调。

我做的部分前端页面

## 一、商品列表页主要对应的表

商品列表页（SellerDashboard.vue 和ProductHistory.vue）  
主要涉及 **两个表**：

| **表名** | **作用** | **说明** |
| --- | --- | --- |
| products | 商品主表 | 用来显示每个商品的名称、价格、描述、图片、上架状态等 |
| buyers | 买家表 | 用来统计某个商品被多少人购买、查看买家详情（通过外键 product\_id） |

## 🧩 二、表之间的关系

products (商品表)

↑

│ 一对多关系（一个商品可对应多个买家）

│

buyers (买家表)

也就是说：

商品列表页主要查询 products 表；

## 💡 三、页面和表的典型对应逻辑

| **页面** | **涉及表** | **功能** |
| --- | --- | --- |
| **SellerDashboard.vue** | products | 显示正在售卖或上架的商品列表 |
| **ProductHistory.vue** | products | 显示所有商品（包括下架/冻结的） |
| **BuyerList.vue** | buyers（通过 product\_id 关联 products） | 查看某一商品的买家信息 |

商品列表页主要用到的是 products 表，里面存放所有商品的基本信息，比如商品名、价格、图片地址、是否上架等。  
如果页面上还显示了买家的数量或购买状态，那就会通过 buyers 表里的 product\_id 去关联查询。  
所以核心是 products 表，buyers 是关联辅助表。