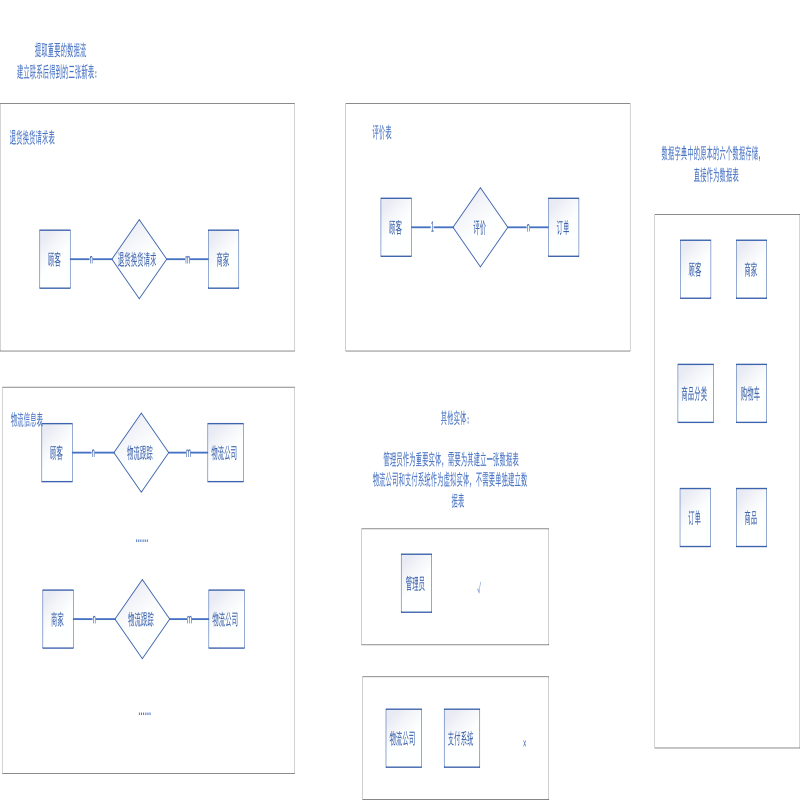
**数据库设计**

**第一版内容**

1. **概念设计：根据需求分析得到的数据流图和数据字典，可以绘制局部E-R图和总体E-R图。由于局部E-R图存在过多冗余，这里只挑选用于生成数据表的实体及联系展示：**

局部E-R图如下：



整体E-R图在修改中。

1. **概念设计：根据E-R图的逻辑模型，生成对应的逻辑模型，即数据表。**

通过E-R图得到的数据表共有10张分别是，admin(管理员表)、cart(购物车表)、customer(顾客表)、logisticsinfo(物流信息表)、merchant(商家表)、orders(订单表)、product(商品表)、productcategory(商品分类表)、returnexchangerequest(退货换货申请表)、review(评价表)。

**以下是设计批注：**

\* 在该数据库中，订单的每一个订单项都进行独立的运输和退货换货处理，因为订单中的订单项可能来自于不同商家，需要独立处理物流运输和退货换货。为了契合这一点，将物流运输表和退货换货表中的外键都改为订单项表的ID，而非订单表的ID。

\* 但如果是这样的话，是否需要把订单表中的商家ID转移到订单项表中？还是说在购物车环节就能把不同商家的订单区分开来？

答：可以将商家ID从订单表中删去，因为订单表中可能存储不同商家的商品；同时，根据第三范式要求，订单项可以通过外键ProductID来查询商品表中的商家ID数据，因此也不用将商家ID加入订单项中。

\* 同样的，单个购物车表无法实现装载多个商品的功能，因此需要进行拆分为cart和cartItem表，cart表存储cartID和顾客ID，cartItem表则将cartID作为外键，同时存储单件商品的各种属性，这样就能通过cartID查询到所有该购物车下的商品，同时也方便对单件商品进行管理。单个商品表也无法存储多张图片，因此也进行拆分。

总体来说：对于从ER图导出的10张数据表，为了减少数据冗余，

其中订单表(orders)分解为订单表(orders)和订单项表(orderItem)

购物车表(cart)分解为购物车表和购物车项表(cartItem)

商品表(product)分解为商品表(product)和商品图片表(productImages)

共计13张表。

1. **数据表格式汇总：**
2. **admin**

| **名** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AdminID | int |  |  | 是 | 否 | 主键 | 管理员ID |
| AdminName | varchar | 50 |  | 是 | 否 |  | 管理员姓名 |
| Password | varchar | 100 |  | 是 | 否 |  | 登录密码 |

1. **cart**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CartID | int |  |  | 是 | 否 | 主键 | 购物车ID |
| CustomerID | int |  |  | 是 | 否 | 外键 | 顾客ID, 外键, 关联顾客表 |
| CreatedAt | datetime |  |  | 否 | 否 |  | 可自动记录该记录的创建时间 |
| UpdatedAt | datetime |  |  | 否 | 否 |  | 可自动记录该记录的修改时间 |

1. **cartitem**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CartItemID | int |  |  | 是 | 否 | 主键 | 购物车具体项目ID |
| CartID | int |  |  | 是 | 否 | 外键 | 购物车ID，外键，关联购物车表 |
| ProductID | int |  |  | 是 | 否 | 外键 | 商品ID，外键，关联商品表 |
| Quantity | int |  |  | 否 | 否 |  | 商品数量 |
| SelectedOptions | varchar | 255 |  | 否 | 否 |  | 商品的规格(颜色、大小等，用于高端功能) |
| Status | enum |  |  | 否 | 否 |  | 状态跟踪（是不是已经被移除了，用于高端功能） |
| AddedAt | datetime |  |  | 否 | 否 |  | 可自动记录该记录的创建时间 |
| UpdatedAt | datetime |  |  | 否 | 否 |  | 可自动记录该记录的修改时间 |

1. **customer**

| **字段名** | **类型** | **长度** | **小数点** | **是否允许为空** | **键** | **备注** |
| --- | --- | --- | --- | --- | --- | --- |
| CustomerID | int |  |  | 否 | 主键 | 顾客ID |
| CustomerName | varchar | 255 |  | 否 |  | 顾客名称 |
| Password | varchar | 255 |  | 否 |  | 登录密码 |
| Money | decimal | 10 | 2 | 是 |  | 余额 |
| PhoneNumber | int |  |  | 是 |  | 手机号码 |
| Address | varchar | 255 |  | 是 |  | 顾客地址 |

1. **logisticsinfo**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LogisticsID | int |  |  | 是 | 否 | 主键 | 物流ID |
| OrderItemID | int |  |  | 是 | 否 | 外键 | 订单项目ID, 外键, 关联订单项表 |
| LogisticsCompany | varchar | 100 |  | 否 | 否 |  | 物流公司 |
| TrackingNumber | varchar | 50 |  | 否 | 否 |  | 物流单号 |
| Status | varchar | 50 |  | 否 | 否 |  | 物流状态 |

1. **merchant**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MerchantID | int |  |  | 是 | 否 | 主键 | 商家ID |
| MerchantName | varchar | 255 |  | 是 | 否 |  | 商家名称 |
| MerchantAddress | varchar | 255 |  | 否 | 否 |  | 商家地址 |
| Password | int |  |  | 是 | 否 |  | 登录密码 |
| PhoneNumber | varchar | 100 |  | 否 | 否 |  | 联系方式 |

1. **orderitem**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OrderItemID | int |  |  | 是 | 否 | 主键 | 订单具体项目ID |
| OrderID | int |  |  | 否 | 否 | 外键 | 订单ID, 外键, 关联订单表 |
| ProductID | int |  |  | 否 | 否 | 外键 | 商品ID, 外键, 关联商品表 |
| Quantity | int |  |  | 否 | 否 |  | 数量 |
| Price | decimal | 10 | 2 | 否 | 否 |  | 单价 |

1. **orders**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OrderID | int |  |  | 是 | 否 | 主键 | 订单ID |
| CustomerID | int |  |  | 否 | 否 | 外键 | 顾客ID, 外键, 关联顾客表 |
| OrderDate | datetime |  |  | 否 | 否 |  | 订单日期 |
| TotalAmount | decimal | 10 | 2 | 否 | 否 |  | 订单总金额 |
| Status | varchar | 50 |  | 否 | 否 |  | 订单状态 |

1. **product**

| **名** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ProductID | int |  |  | 是 | 否 | 主键 | 商品ID |
| CategoryID | int |  |  | 否 | 否 | 外键 | 分类ID, 外键, 关联商品分类表 |
| MerchantID | int |  |  | 否 | 否 | 外键 | 商家ID, 外键, 关联商家表 |
| ProductName | varchar | 255 |  | 否 | 否 |  | 商品名称 |
| Description | text |  |  | 否 | 否 |  | 商品描述 |
| Price | decimal | 10 | 2 | 否 | 否 |  | 商品价格 |

1. **productcategory**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CategoryID | int |  |  | 是 | 否 | 主键 | 分类ID |
| CategoryName | varchar | 100 |  | 是 | 否 |  | 分类名称 |
| Description | text |  |  | 否 | 否 |  | 分类描述 |

1. **productimages**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ImageID | int |  |  | 是 | 否 | 主键 | 图片ID |
| ProductID | int |  |  | 否 | 否 | 外键 | 商品ID，外键，关联商品表 |
| image\_data | mediumblob |  |  | 否 | 否 |  | 图片数据 |

1. **returnexchangerequest**

| **名称** | **类型** | **长度** | **小数点** | **不是 null** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RequestID | int |  |  | 是 | 否 | 主键 | 退货换货请求ID |
| OrderItemID | int |  |  | 否 | 否 | 外键 | 订单项目ID, 外键, 关联订单项表 |
| CustomerID | int |  |  | 否 | 否 | 外键 | 顾客ID, 外键, 关联顾客表 |
| RequestType | varchar | 50 |  | 否 | 否 |  | 请求类型(是退货、换货, 还是没有操作) |
| Reason | text |  |  | 否 | 否 |  | 退货换货原因 |
| Status | tinyint | 1 |  | 否 | 否 |  | 退货换货状态(即商家是否同意, 值取0 / 1) |

1. **review**

| **名称** | **类型** | **长度** | **小数点** | **是否允许为空** | **虚拟** | **键** | **注释** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ReviewID | int |  |  | 否 | 否 | 主键 | 评价ID |
| CustomerID | int |  |  | 是 | 否 | 外键 | 顾客ID, 外键, 关联顾客表 |
| OrderItemID | int |  |  | 是 | 否 | 外键 | 订单项目ID, 外键, 关联订单项表 |
| Rating | decimal | 2 | 2 | 是 | 否 |  | 评分 |
| Comment | text |  |  | 是 | 否 |  | 评论 |
| ReviewDate | datetime |  |  | 是 | 否 |  | 评价日期 |

1. **主要的SQL查询语句：**

**累了，下次弄。**

**第二版内容**

1. **数据库第一次整体更改：**

**①给所有表增加两个字段：创建时间和更改时间，除了管理员表。** AddedAt DATETIME DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

这两个字段会自动读取对应的时间并登记，对实际的插入和更改记录操作不会有任何影响。同时，也能方便使用一些功能。相应的，原有的所有与时间有关的字段全部删除。

**②为所有的表增加主键自动递增功能：**

备注：这个改动非常麻烦，下次还是一开始就弄好。不然每修改一张表都要删除增加好几张表的外键。(╥╯^╰╥)

**③细节修改：**

1. 为顾客表、商家表、商品分类表新增头像 / 图标属性。
2. 将所有图片的数据类型改为varchar，因为所有图片都会存放在resource的静态区域，数据库只需要存储图片路径，真实的图片可以交给vue前端渲染。
3. 将所有和“钱”有关的数据类型都改为double，长度10，小数点2。
4. 将text、enum、tinyint、time、decimal等各种的数据类型，统一为int、double、varchar这三种数据类型，以便与实体类对接。
5. 修改第11张表的名称，从productimages到productimage。
6. **不完备的地方**

**①本次数据库中，logisticsinfo、orders、productcategory、productimage、returnexchangerequest这5张表的主键，不等同与表名+ID，在写代码时需要注意。**

**②本次数据库还没有对一些重要的用户自定义完整性约束进行规范，如用户对同一商品不能发送多次退货换货申请、一个orders表中不应该有多个productId相同的ordersitem项。**

1. **数据库与后端连接(DAO层设计)**

**①设计采用Mybatis进行设计，将传统的Repository包的功能和部分Service包的功能进行封装，用mapper包和存放在resource文件夹下的mapper.xml包定义与数据库的交互功能，包括增删改查和其他拓展功能。**

**②如果要增加系统的功能，如查询价格达到XXX的所有订单，操作步骤如下：**

1. **通过Controller层传入请求后，在Service层增加对应的功能函数function1，输入是查询请求，返回值是List<订单类>。再增加对应数据表的mapper接口私有变量（此处为订单接口）。**
2. **同时，在订单接口）增加相同名称的功能函数function1，Service层就可以调用订单接口私有变量的function1来满足Service层function1需求。**
3. **订单接口的功能函数名function1将被添加到订单类的mapper.xml配置文件，只要在mapper.xml层配置好SQL语句和返回值类型，订单层就可以通过function1的名称来使用SQL查询，并将查询结构返回给Service层的function1函数。**
4. **修改后的数据库：**

**累了，下次再传。**