**图书销售管理系统**

**——课程作业报告**

崔德方、陈卓、热合玛·阿不力克木、张玉婷

# 需求分析

## 功能划分

### 前台系统内容

会员事务

订单处理

### 后台系统内容

## 数据流图

**用户信息数据流图**

****

**会员检索数据流图**



**购物车数据流图**



**评价数据流图**

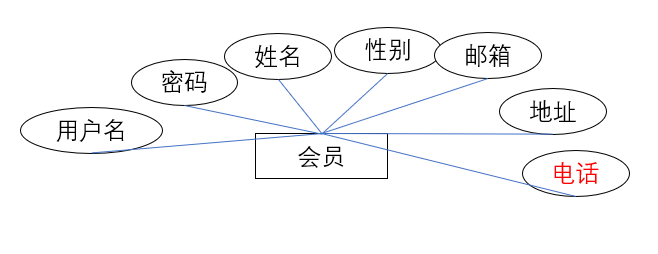
**管理员图书管理数据流图**

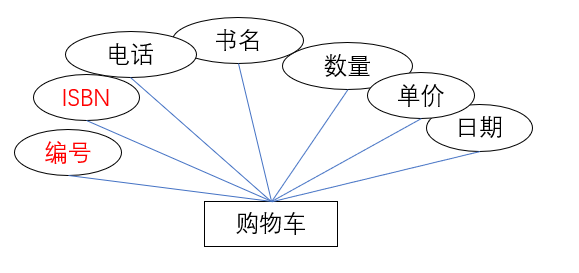


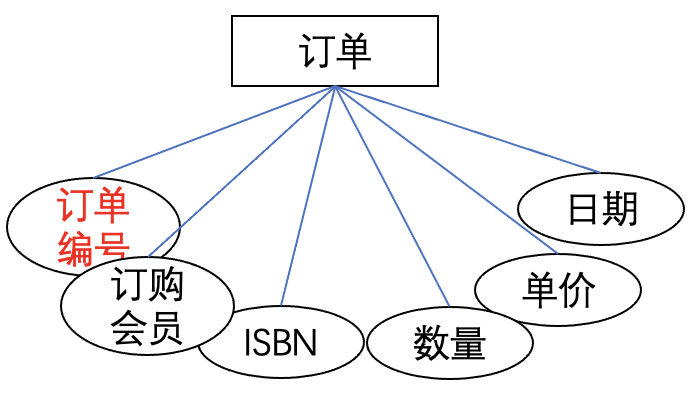
# E-R模型分析

## 确定实体

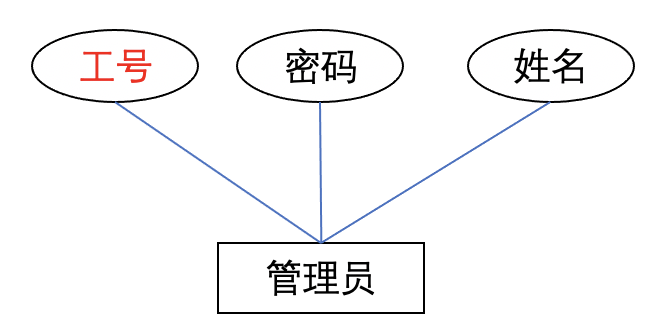
**会员及其购物车、订单**



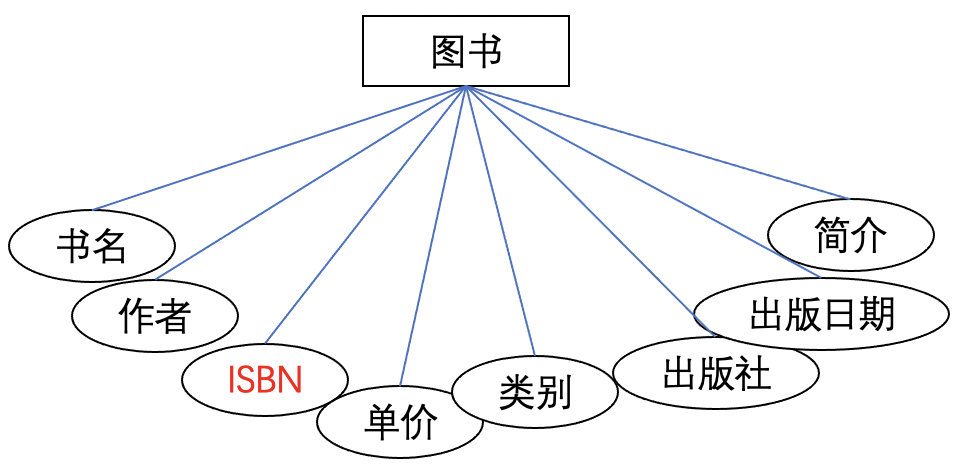




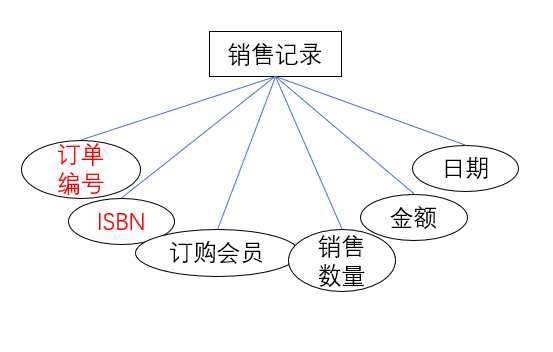
**管理员**



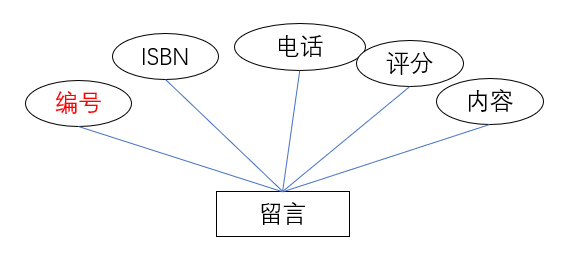
**图书信息**

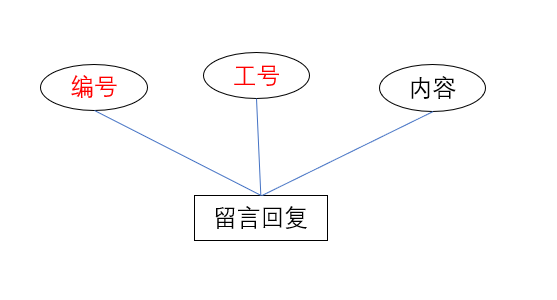


**销售记录**

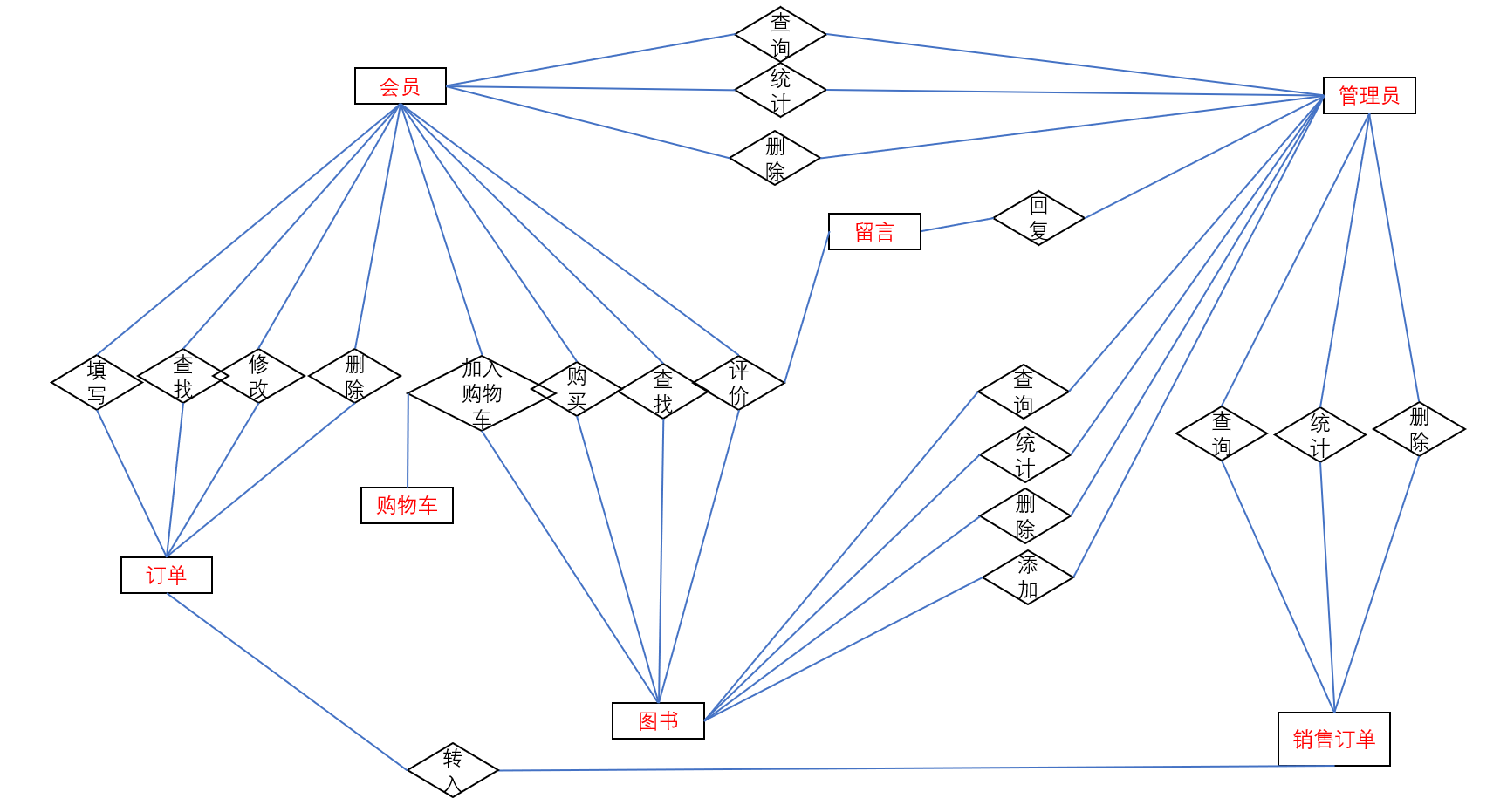


**留言**





## E-R模型



# 逻辑结构设计

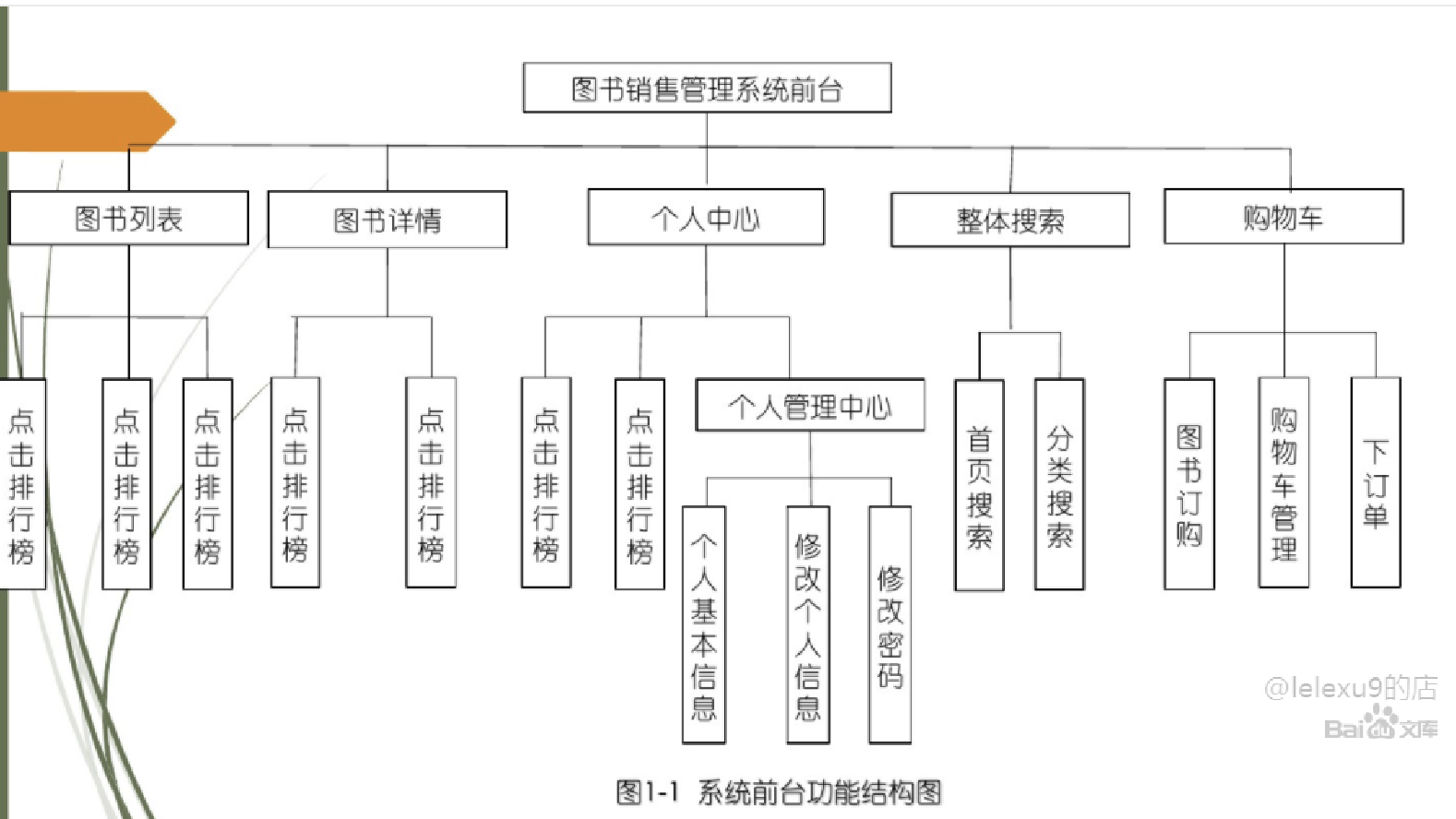
## ER模型向关系模式转换

见上

## 数据库表结构设计

见上

## 应用程序设计

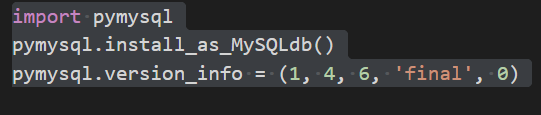


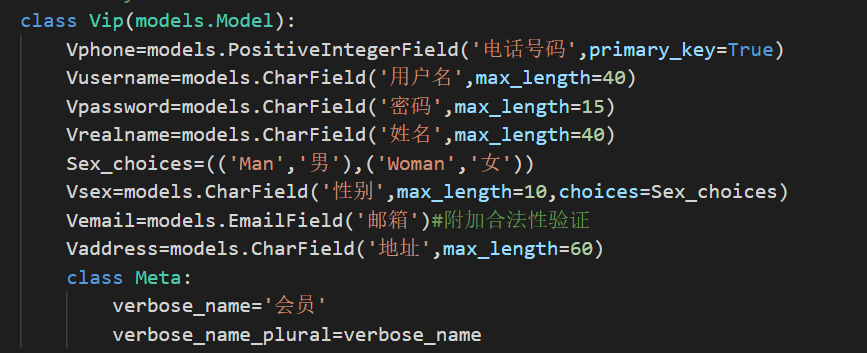
# 实现

后端中数据库的建立使用Django实现，Django中默认的数据库使用的是sqlite3数据库，我们将其改成了mysql，需要在setting.py中修改，

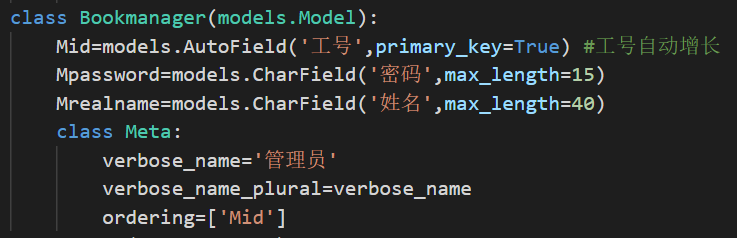


同时在与setting.py同级中的\_init\_.py中做如下修改，接下来就可以在model.py中进行我们的数据库设计了。



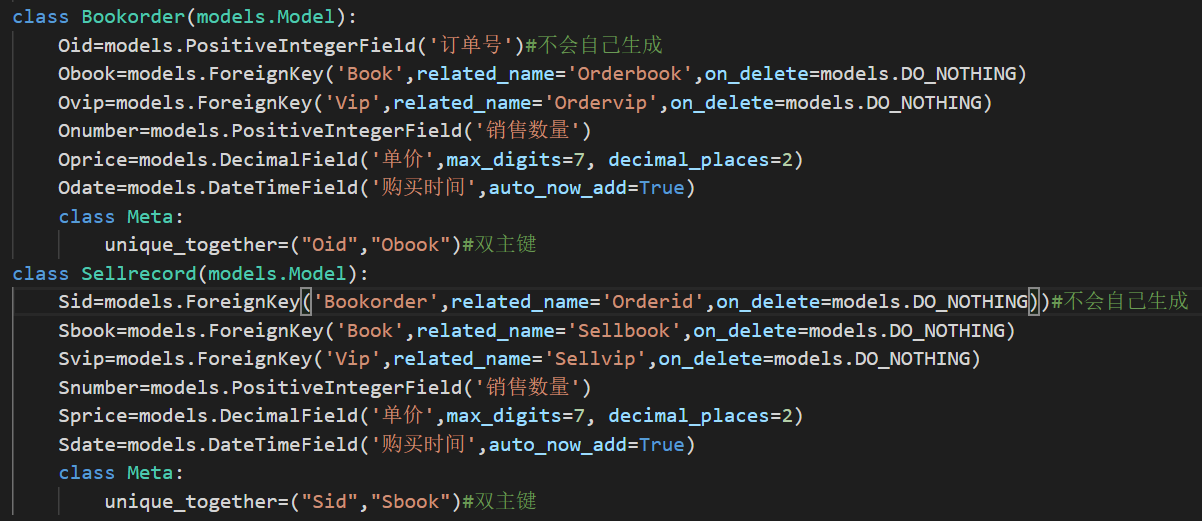


我们的VIP类中使用电话号码作为主码，因为每个用户注册只能有一个电话号，性别部分我们用的是选择的方法，只能选择“男”或者“女”，邮箱部分我们使用的是Django中自带的EmailFIeld，可以对邮箱进行合法性验证。

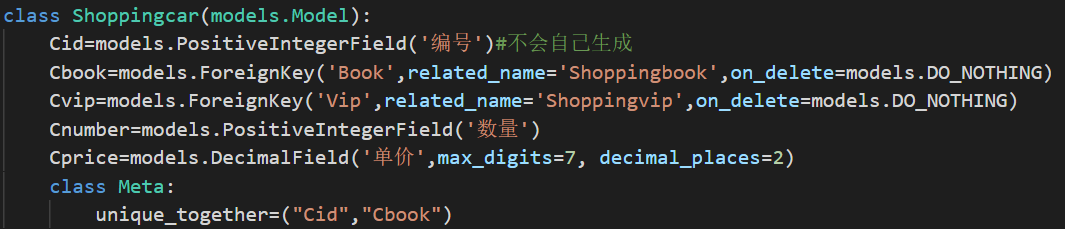


管理员类，工号我们采用了AutoField，它会自动生成一个自动增长的数，我们把它设置为主码。

书籍我们的主码是ISBN。



销售记录和订单结构上应该是完全一样的因为订单完成后就会进入销售记录中，两个类中的book和vip都是外码，销售记录中的Sid也是外码对应着订单，这两类采用了双主码的方式因为一个订单可以有好多本书，用订单号和书的ISBN两个为主码，相同订单号的是一个订单，不同的ISBN表示同一订单的不同种书，而Django中不能设置两个primary\_key，因此在Meta中将两个属性绑定起来，让他们unique，同时不设主码即可（系统自动生成）购买时间是自动添加的不需要用户填写。购物车类基本差不太多。





最后是评论和回复评论类，由于是多对多的关系，因此又开了一个commentlist记录回复的管理员和回复的留言，如果放到comment里会导致很多空值，因为回复的留言会比留言少。