## 会员管理

某电商，计划在其公众号上实现会员管理。

具体需求：

1. 老会员可以邀请新会员加入，形成上下级关系。每一个会员可以有多个下级，但是只有一个上级。
2. 每个会员，都有多个消费记录，记录其消费明细。
3. 每个月底，统计每个会员及其下级会员的消费总额

请完成下列任务：

1. 设计这个会员管理功能所需的数据库表（PostgreSQL）

数据库名字：common\_number

表1：order

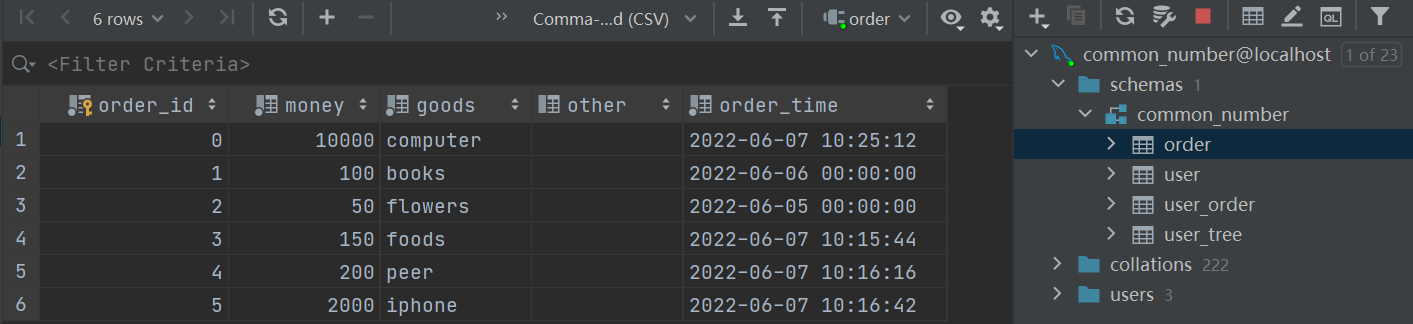


表2：user

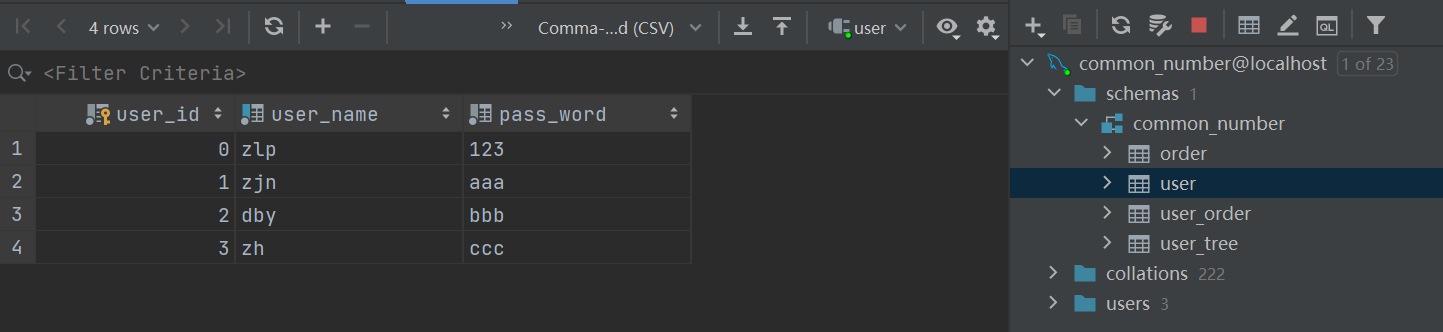


表3：user\_order

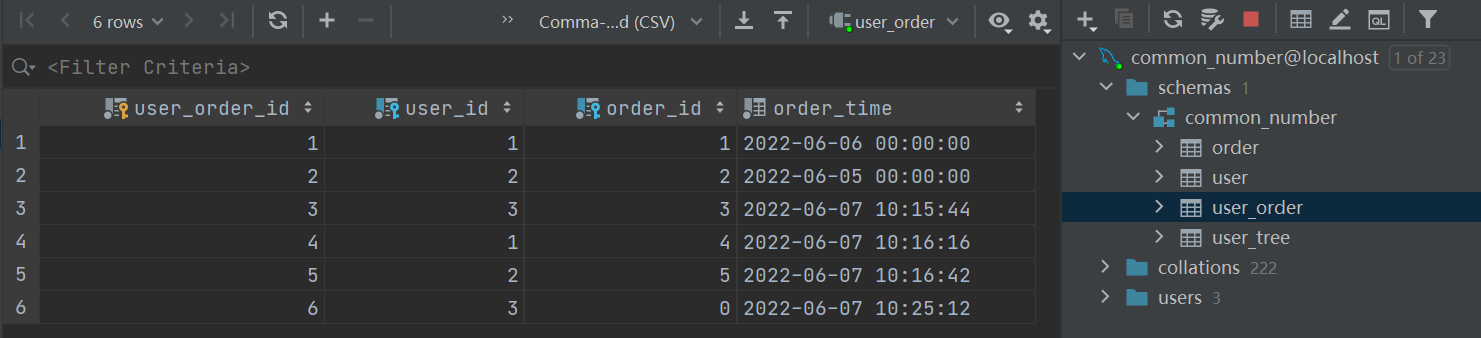
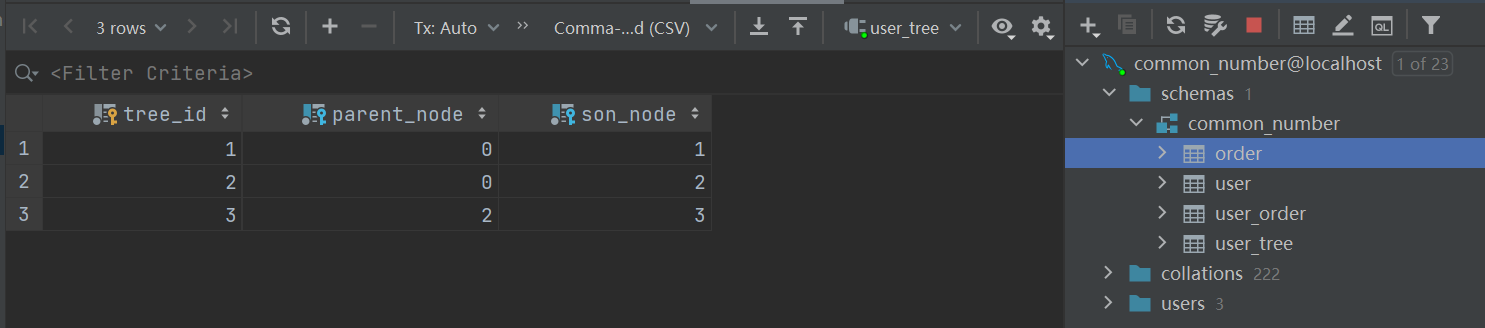
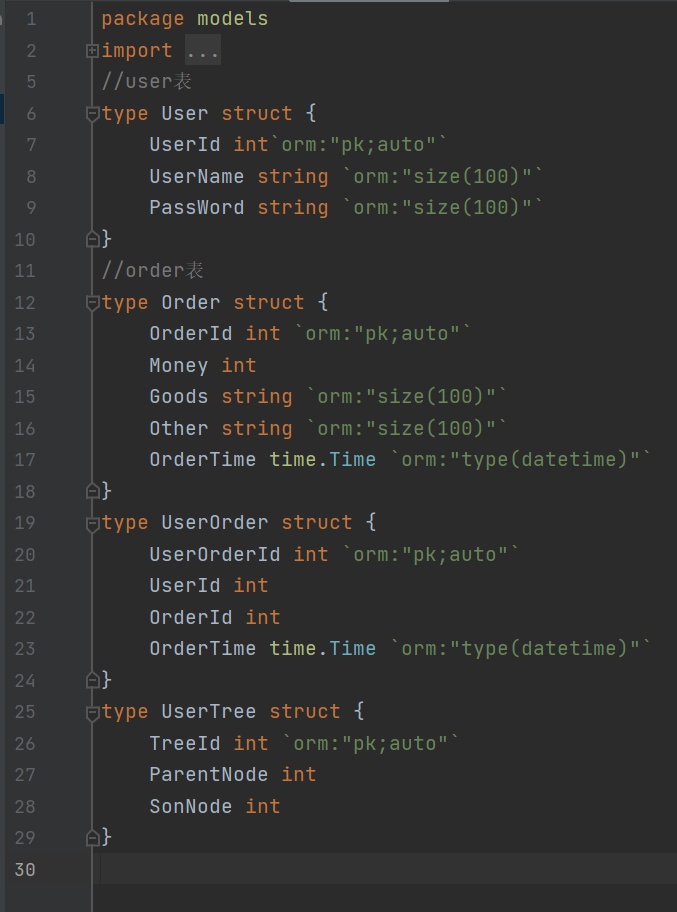


表4：user\_tree



1. 用你所熟悉的go语言的一种ORM，描述所设计的数据表

使用gorm



1. 编写计算某个会员，在某个月消费总额：其自身消费和所有下级会员的消费的总和

func GetUserCost(userId int, begin time.Time,end time.Time) int {

fmt.Println(userId,begin,end)

var ans = 0

//打开数据库

db,err := gorm.Open("mysql","root:123@tcp(127.0.0.1:3306)/common\_number?charset=utf8&parseTime=True&loc=Local")

db.SingularTable(true)

if err!=nil{

panic(err)

}

defer db.Close()

//查询user\_order

var userOrders []models.UserOrder

db.Where("user\_id=? AND order\_time>=? AND order\_time<=?",userId,begin,end).Find(&userOrders)

//查询某用户的money

for \_,userOrder:=range userOrders{

var order models.Order

db.Where("order\_id=?",userOrder.OrderId).Find(&order)

//获取单笔订单的金额

ans+=order.Money

}

//获取下级会员

var trees []models.UserTree

db.Where("parent\_node=?",userId).Find(&trees)

for \_,node:=range trees{

ans+=GetUserCost(node.SonNode,begin,end)

}

return ans

}