回想一下，在你网上购物之前，想要成功下单就必须先成为某购物网站的会员。大多数网站会根据会员的消费水平来划分会员等级，制定相应的优惠策略。现在，如果你是一名电子商务系统中订单系统的开发者，你很有可能会遇到这样一个应用场景——

假设我们定义一个会员最近的一年之内消费超过10000元就会成为该购物网站的白金会员（注意：这里不考虑会员的充值金额，即仅考虑会员购买商品时的消费累积总额）。同时我们定义一个活动，有某个商品白金会员将享受6折的优惠。这时如果一个新会员在该购物网站第一次购买便购买了10000元的该商品，将会有以下三种情形出现：

（1）系统按照订单支付前累积，判断该新会员满足消费超过10000元的要求，升级为白金会员，同时享受该商品的6折折扣；

（2）系统按照订单支付后累积，判断若打折之后不满足10000元，则不给予折扣，但是升级该新会员为白金会员，同时如果该会员紧接的再下一个订单，则该笔订单便可以享受折扣；

（3）系统需要在会员完成订单之后（即签收订单，前面两种情况存在会员不签收，取消订单的可能）升级为白金会员，接下来再下第二个订单时才能享受折扣。

考虑以上3种情形，显然从消费者的角度出发，消费者会更倾向于第（1）、（2）种情形，对于商家而言，这样的策略也更能刺激消费者消费，但是同时商家也要承担恶意消费者进行骗单的风险，即可能存在某些消费者先下一个10000元的订单，升级为白金会员，然后紧接着再下第二个订单享受6折优惠，接着却把第一个订单退掉。根据消费者心理，第二笔订单往往不足10000元，比如只有1000元。这时商家便损失了400元。而在第（3）种情形中，商家承受骗单的风险则大大减小，但是相应地对消费者的吸引力也会有所下降。在一家购物网站发展的初期，往往会选用第（1）、（2）中情形，在发展日益成熟的时候，则会选用第（3）种情形。

因此，作为开发者的你，从网站长远发展的角度考虑一下如何在订单系统的开发中实现可设置的会员级别增长功能，也就是说所开发的订单系统中允许管理人员通过设置系统配置完成以上至少两种情形的切换。

初步的相关数据库模式如下所示，你也可以根据需要进行相应改动。

CustomerInfo (customerID, customerName, customerTelephone, gradeType, totalAmount )

// CustomerInfo表示客户相关信息，其中customerID 为主键，表示会员编号，gradeType表示会员等级，totalAmount表示消费总额。

Order (orderID, customerName, telephoneID, orderTime, orderAmount )

// Order表示订单相关信息，其中orderID为主键，orderAmount表示单次下单金额。