课 程 设 计 报 告

实训名称 Oracle 课程实训

设计题目 网上商城购物系统

实习时间 2020.12.21-2020.12. 31

专业班级 软件 186

学　　号 201811050370

姓　　名 徐贝宁

实训时间 2020.12.21-2020.12. 31

目录

网上商城购物系统的设计与实现

1 实训目的

随着现代科学技术的发展，计算机的应用几乎进入了生活中的每一个领域，数据库技术在软件开发中也得到了越来越深入而广泛的应用。本实训的目的旨在通过项目实战，理解数据库的基本概念，熟悉 Oracle 的基本操作，创建用户、表、检查约束，编写 SQL 语句执行数据库查询、新增、修改、删除，为 Java、.Net 开发提供数据存储及数据操作的基础。

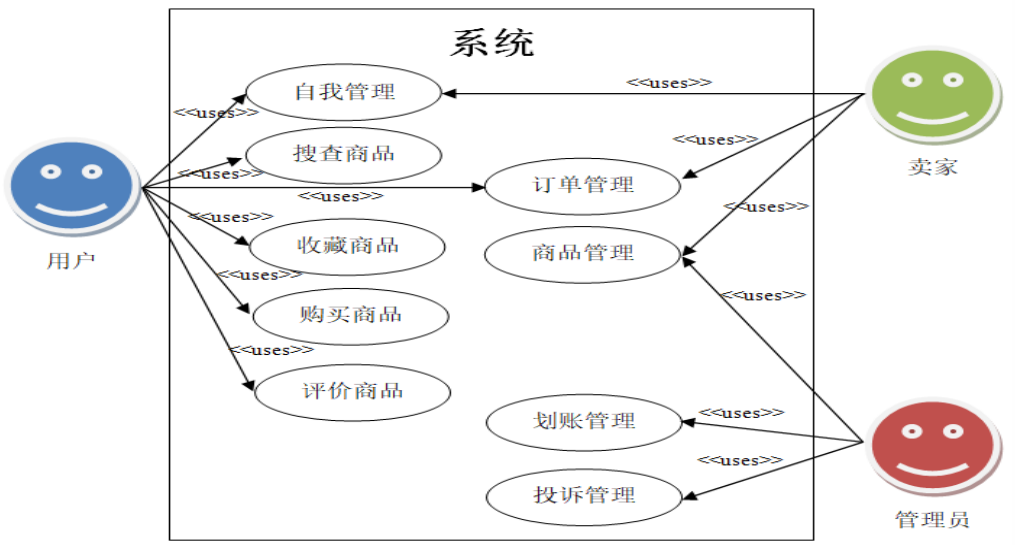
2 需求分析

2.1 系统用户需求分析

角色名称	职责描述
网站管理员	本网站维护人员，负责本网站的日常维护工作。
普通用户	本网站的注册人员。

2.2 系统功能性需求

2.2.1 业务整体用例图



2.2.2 整体功能划分

在本系统中，主要包括用户、卖家和管理员三个角色。本系统初期重点突出用户的管理的权限，具体用户的权限如下：

1.用户自我管理。主要包括新用户的注册、登录和退出登录以及用户基本信息的管理三个部分，实现用户的相关自我的一些基本操作。

2.查看商品。实现用户对商品的搜索查看等功能。

3.收藏夹。实现用户对商品的收藏。

4.购买商品。实现用户对商品的购买。

5.评价商品。实现用户收货后对商品的评价。

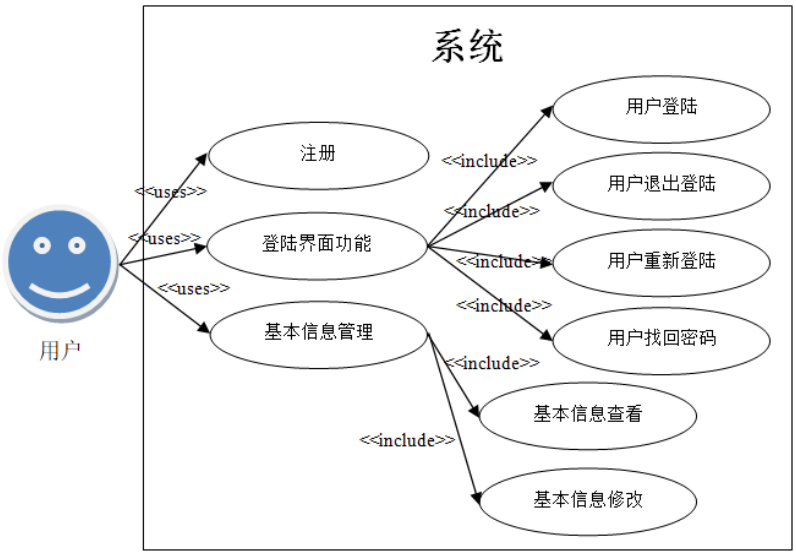
管理员权限如下：

1.商品管理。体现管理员对各卖家的商品进行管理，如发现不合格商品，马上下架。

2.支付宝。体现用户和卖家之间的桥梁，用户收到货物后管理员将用户打到支付宝的钱转到卖家卡上。

3.举报管理。实现用户对卖家的投诉，核实实情，解决用户与卖家的矛盾，如果卖家货物不合格，则将卖家账号整顿，合格后继续上架。

2.2.3 用户自我管理



用例描述：

<table>
<tr><td>用例名称</td><td colspan="3">用户注册</td></tr>
<tr><td>用例编号</td><td colspan="3">SRS-01</td></tr>
<tr><td>用例简介</td><td colspan="3">注册一个新用户</td></tr>
<tr><td>优先级</td><td colspan="3">High</td></tr>
<tr><td>前置条件</td><td colspan="3">无</td></tr>
<tr><td>后置条件</td><td colspan="3">显示注册成功提示，并跳到登录页。</td></tr>
<tr><td rowspan="4">操作流程</td><td>步骤</td><td>触发者</td><td>描述</td></tr>
<tr><td>1</td><td>用户</td><td>输入个人信息。</td></tr>
<tr><td>2</td><td>系统</td><td>检测个人信息是否合法。</td></tr>
<tr><td>3</td><td>系统</td><td>将个人信息录入数据库。</td></tr>
<tr><td>替代流程</td><td colspan="3"></td></tr>
<tr><td>例外流程</td><td colspan="3"></td></tr>
<tr><td>约束条件</td><td colspan="3">注册用户名要合法，并检测用户名是否为空或已存在。</td></tr>
<tr><td>输入及约束</td><td colspan="3">用户名(name) ：字符串，长度 0-10，直接输入，必填。
密码(password)：字符串，长度 0-30，直接输入，必填。
确认密码(passwordCheck)：字符串，长度 0-30，直接输入，必填。
省份(province)：整型，必填。
城市(city)：整型，必填。
邮箱(email)：字符串，长度 0-30，直接输入。
手机号(telephone)：字符串，长度 11，直接输入。
验证码(validateCode)：字符串，长度 0-5，用户输入。必填。</td></tr>
</table>

用例名称	用户登录
用例编号	SRS-02
用例简介	用户登录
优先级	High
前置条件	用户已经注册

后置条件	转到用户管理主页。		
操作流程	*步骤*	*触发者*	*描述*
	1	用户	输入登录信息，勾选是否记住用户。
	2	系统	检测是否为空。
	3	系统	若勾选，则写入 Cookie，访问数据库，检测登陆信息是否正确。若正确，改写 Session，进入个人管理首页；否则，请用户重新登录，到用例 SRS-05。
替代流程			
例外流程			
约束条件	检测输入的用户名是否合法，并检测用户名是否为空。		
输入及约束	用户名(name)：字符串，长度 0-10，直接输入，必填。 密码(password)：字符串，长度 0-30，直接输入，必填。 是否记住用户(isRemember)：布尔型，可选输入。		

用例名称	用户退出登录		
用例编号	SRS-03		
用例简介	用户退出登录		
优先级	High		
前置条件	用户已经登录		
后置条件	转到用户登录页。		
操作流程	*步骤*	*触发者*	*描述*
	1	用户	单击退出链接。
	2	系统	置 Session 为 Null。转到用户登录页面。
替代流程			
例外流程			
约束条件	无。		
输入及约			

束	

用例名称	用户找回密码		
用例编号	SRS-04		
用例简介	用户找回密码		
优先级	High		
前置条件			
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	单击找回密码链接，输入想要找回账号的登录名。
	2	系统	访问数据库，检测邮箱是否存在。
	3	用户	若存在，发送密码到邮箱；否则，提示用户错误信息。
替代流程			
例外流程			
约束条件	无。		
输入及约束	用户名(name)：字符串，长度 0-10，直接输入，必填。		

用例名称	用户重新登录		
用例编号	SRS-05		
用例简介	用户重新登录		
优先级	High		
前置条件	用户登录失败		
后置条件	转到用户管理主页。		
操作流程	*步骤*	*触发者*	*描述*
	1	用户	输入登录信息，勾选是否记住用户。

	2	系统	检测是否为空。
	3	系统	若勾选，则写入 Cookie，访问数据库，检测登陆信息是否正确。若正确，改写 Session，进入个人管理首页；否则，重复本用例。
替代流程			
例外流程			
约束条件	无。		
输入及约束	用户名(name)：字符串，长度 0-10，直接输入，必填。 密码(password)：字符串，长度 0-30，直接输入，必填。		

用例名称	用户基本信息查看		
用例编号	SRS-06		
用例简介	用户分类别的查看自己或者他人的详细信息。		
优先级	High		
前置条件	用户已经登录		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	进入详细信息显示页。
	2	系统	按标签显示用户各类信息（基本信息、联系方式、个人爱好），初始显示基本信息。
	3	用户	单击其它未展开选项卡。
	4	系统	展开相应选项卡，显示相应信息。
替代流程			
例外流程			
约束条件	无。		
输入及约束			

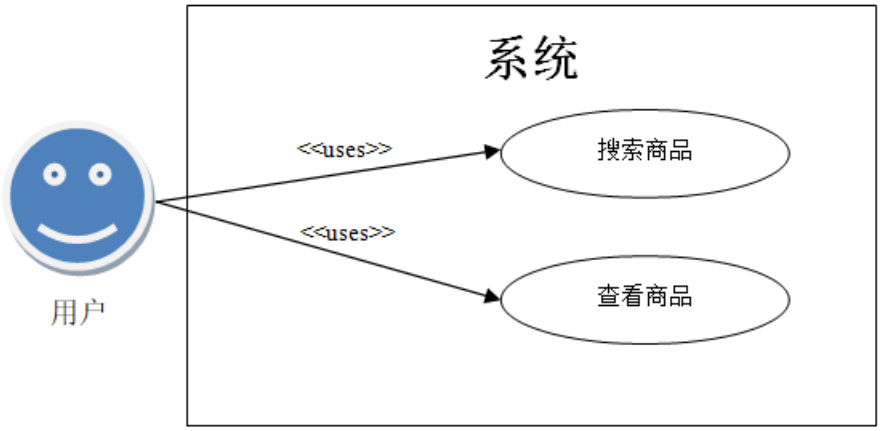
<table>
<tr><td>用例名称</td><td colspan="3">用户基本信息修改</td></tr>
<tr><td>用例编号</td><td colspan="3">SRS-07</td></tr>
<tr><td>用例简介</td><td colspan="3">用户修改自己的详细信息。</td></tr>
<tr><td>优先级</td><td colspan="3">High</td></tr>
<tr><td>前置条件</td><td colspan="3">用户已经登录</td></tr>
<tr><td>后置条件</td><td colspan="3">进入用户详细信息显示页</td></tr>
<tr><td rowspan="6">操作流程</td><td>步骤</td><td>触发者</td><td>描述</td></tr>
<tr><td>1</td><td>用户</td><td>打开自己的详细信息显示页，并进入修改页面。</td></tr>
<tr><td>2</td><td>系统</td><td>分类显示用户的基本信息，要求以文本框的形式显示。</td></tr>
<tr><td>3</td><td>用户</td><td>修改自己的基本信息(姓名、昵称、姓名可见度、性别、年龄、生日、所在省份、所在城市、头像、个性签名)。
修改自己的联系方式(联系电话、QQ、MSN)。
修改自己的个人爱好(爱好书籍、爱好音乐、爱好电影、爱好运动、爱好游戏)。</td></tr>
<tr><td>4</td><td>用户</td><td>单击确定按钮。</td></tr>
<tr><td>5</td><td>系统</td><td>将用户修改的信息提交到数据库，并返回修改结果。</td></tr>
<tr><td>替代流程</td><td colspan="3"></td></tr>
<tr><td>例外流程</td><td colspan="3"></td></tr>
<tr><td>约束条件</td><td colspan="3">无。</td></tr>
<tr><td>输入及约束</td><td colspan="3">性别(sex)：布尔型，选择输入。
年龄(age)：整型，选择输入。
账户余额(money):money 型，默认为 0。</td></tr>
</table>

	生日(birthday)：日期类型，选择输入。 所在省份(provinceID)：外键(Province 表)，长度 0-255，列表框选择输入。 所在城市(cityID)：外键(City 表)，长度 0-255，列表框选择输入。 头像(photo)：字符串，长度 0-500，记录文件保存在网站文件夹的位置，用户上传。 个性签名(dsp)：字符串，长度 0-200，直接输入。 邮箱(email)：字符串，长度 0-30，直接输入。 手机号(telephone)：字符串，长度 11，直接输入。 QQ 号码(qq)：字符串，长度 0-15，只要求数字，直接输入。 MSN(msn)：字符串，长度 0-50，邮箱格式，直接输入。 爱好书籍(loveBook)：字符串，长度 0-200，直接输入。 爱好音乐(loveMusic)：字符串，长度 0-200，直接输入。 爱好电影(loveMovie)：字符串，长度 0-200，直接输入。 爱好运动(loveSport)：字符串，长度 0-200，直接输入。 爱好游戏(loveGame)：字符串，长度 0-200，直接输入。

系统是直接面对使用人员的，而使用人员往往对计算机并不是非常熟悉。这就要求系统能够提供良好的用户接口，易用的人机交互界面。要实现这一点，就要求系统应该尽量使用用户熟悉的术语和中文信息的界面；针对用户可能出现的使用问题，要提供足够的在线帮助，缩短用户对系统熟悉的过程。

对系统中涉及到的数据，系统要提供方便的手段供系统维护人员进行数据的备份，日常的安全管理，系统意外崩溃时数据的恢复等工作。

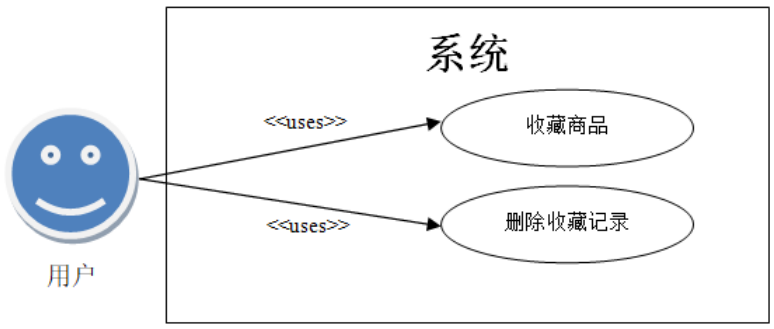
2.2.4 搜查商品



用例名称	搜索商品		
用例编号	SRS-08		
用例简介	用户输入关键字来搜索相应的所有商品。		
优先级	High		
前置条件			
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	进入搜索页面，输入搜索商品的关键字，可选择一项搜索关键字的条件（如商品类别、品牌、价格等）来搜索商品。
	2	系统	根据用户搜索的关键字查询数据库，若找到符合要求的商品则显示出来，否则显示没有相应商品信息存在。
替代流程			
例外流程			
约束条件	输入搜索商品关键字不为空。		
输入及约束	关键字(searchKey)：字符串，长度 0-20，直接输入，必填。		

用例名称	查看商品		
用例编号	SRS-09		
用例简介	用户查看相应商品的详细信息。		
优先级	High		
前置条件			
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	点击已搜索到的想要查看其信息的商品。
	2	系统	根据用户点击的商品查询数据库，若找到符合要求的商品则将该商品的详细信息显示出来，否则显示没有相应商品信息存在。
替代流程			
例外流程			
约束条件			
输入及约束			

2.2.5 收藏商品

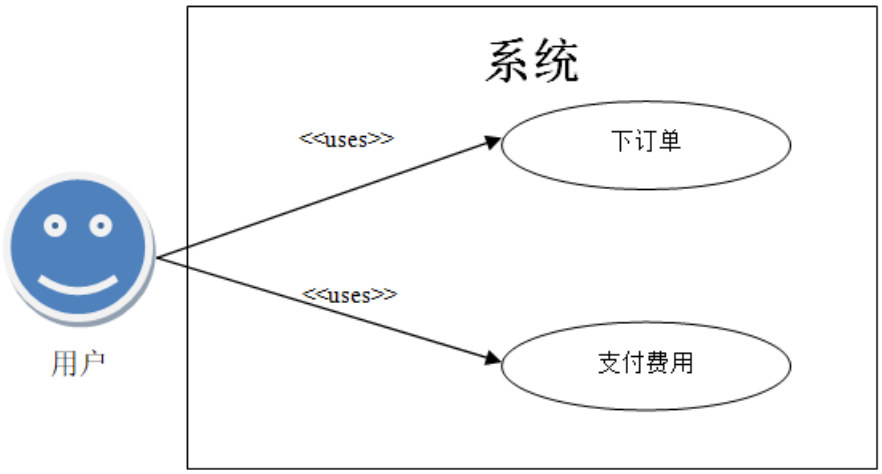


用例名称	收藏商品
用例编号	SRS-10

用例简介	用户收藏看中的相应商品。		
优先级	High		
前置条件	用户已经登录		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	点击收藏商品。
	2	系统	将相应收藏的商品收藏到用户的收藏商品中。
替代流程			
例外流程			
约束条件			
输入及约束			

用例名称	删除收藏记录		
用例编号	SRS-11		
用例简介	用户删除收藏商品的记录。		
优先级	High		
前置条件	用户已经登录		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	选中收藏的商品，点击删除收藏。
	2	系统	将相应收藏商品的记录删除。
替代流程			
例外流程			
约束条件			
输入及约束			

2.2.6 购买商品

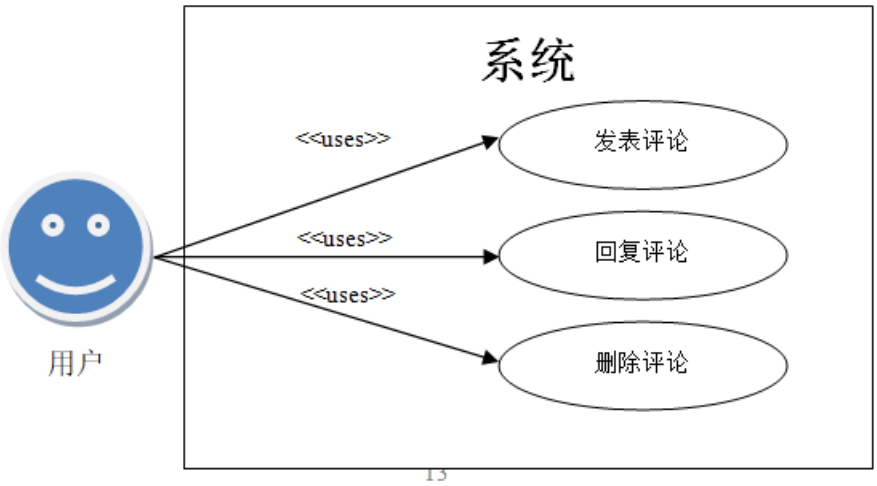


用例名称	下订单		
用例编号	SRS-12		
用例简介	用户将想要购买的商品订单下好，生成新的订单。		
优先级	High		
前置条件	用户已经登录		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	将订单的具体信息填写完整。
	2	系统	将这份订单的信息存入数据库，生成一份新订单，没有付款则订单的付款项为：未付款，订单只下发给卖家但卖家不发货。
替代流程			
例外流程			
约束条件	商品信息不为空、商品库存足够订单要求、用户账户余额足够、一个订单号只给一次订单使用，不管订单完成与否订单号只能使用一次。		
输入及约	买家用户名：从买家信息表中调出该信息，存入订单表中。		

束	卖家用户名：从卖家信息表中调出该信息，存入订单表中。 商品信息：从商品表中调出该商品的各类信息。 购买数量：整型，直接输入，必填。 送货地址：字符串，长度 0-100，必填。 买家手机号：字符串，长度 11，必填。

用例名称	支付费用		
用例编号	SRS-13		
用例简介	用户支付费用到商城（作为中间人），并在订单中显示已付款，卖家可发货。		
优先级	High		
前置条件	用户已经登录，并下订单。		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	选择支付此订单。
	2	系统	将订单中显示为已付款，提醒卖家发货。
替代流程			
例外流程			
约束条件	买家支付宝中余额足够。		
输入及约束			
备注	将订单中支付状态改为已支付。		

2.2.7 评价商品

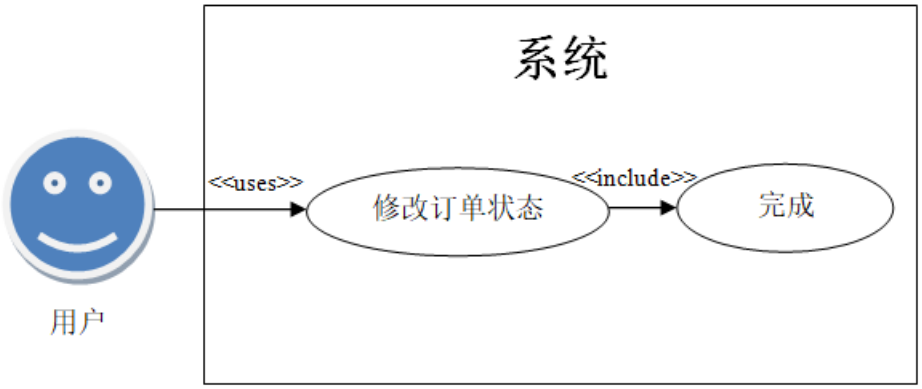


用例名称	发表、回复评论		
用例编号	SRS-14		
用例简介	发表一条新的留言，或者对已有的留言进行回复。		
优先级	High		
前置条件	用户是已注册成员，且已收到商品。		
后置条件	跳转到该商品或商家的评论区。		
操作流程	*步骤*	*触发者*	*描述*
	1	用户	进入留言面板。
	2	系统	显示相关有效评论。
	3	用户	单击回复按钮。
	4	系统	在新建留言文本框里面显示“回复 XX：”。
	5	用户	在新建留言文本框里输入留言内容。
	6	用户	单击发表按钮。
	7	系统	进入数据库，进行相关操作。若成功，刷新留言列表；若失败，通过提示框提示错误信息。
替代流程			
例外流程			

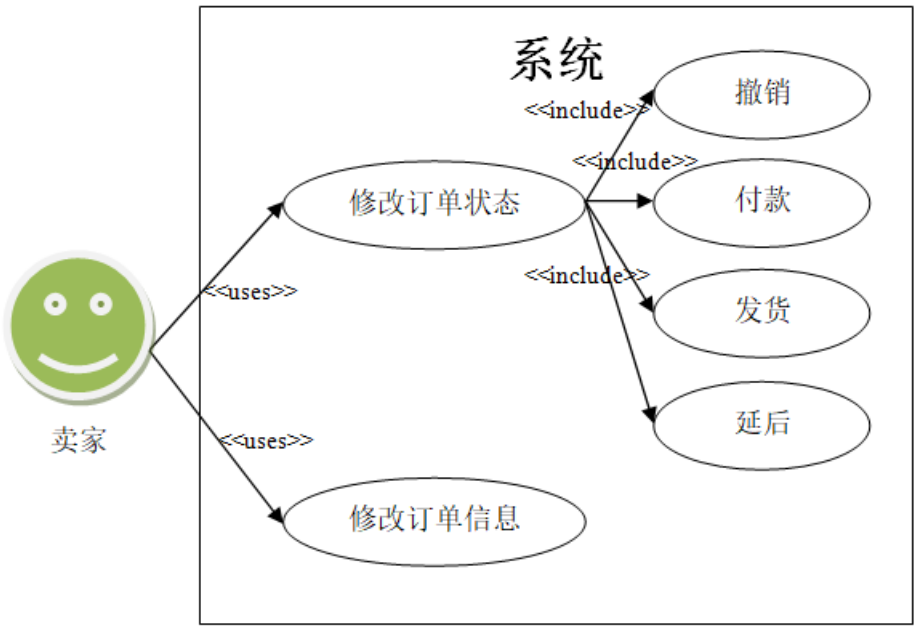
约束条件	无。
输入及约束	评论用户名：依赖于买家用户名

用例名称	删除评论		
用例编号	SRS-15		
用例简介	对已发表的评论进行删除。		
优先级	High		
前置条件	用户是已注册成员，且已发表评论。		
后置条件	跳转到该商品或商家的评论区。		
操作流程	*步骤*	*触发者*	*描述*
	1	用户	进入留言面板。
	2	系统	显示相关有效评论。
	3	用户	单击删除按钮。
	4	系统	删除评论。
	5	系统	进入数据库，进行相关操作。若成功，刷新留言列表；若失败，通过提示框提示错误信息。
替代流程			
例外流程			
约束条件	卖家不可以删除买家评论。		
输入及约束	评论用户名：依赖于买家用户名		

2.2.8 订单管理



用例名称	用户修改订单状态		
用例编号	SRS-16		
用例简介	用户收到货物后对订单状态改为完成，管理员可进行转账。		
优先级	High		
前置条件	有用户已对卖家下订单，并且订单状态为发货状态。		
后置条件	显示订单状态修改为：XX，后跳转到该订单信息页面。		
操作流程	*步骤*	*触发者*	*描述*
	1	用户	收到货后用户打开订单。
	2	系统	从数据库调出订单信息。
	3	用户	用户将订单状态改为完成。
	4	系统	数据库记录修改后的订单状态（完成）。
替代流程			
例外流程			
约束条件			
输入及约束	商品信息（从订单所对应的外键商品表中调取信息）。 卖家用户名（从订单表中调出信息）。 用户信息（从数据库调出发货地址、手机号等）。 修改订单状态：完成。		

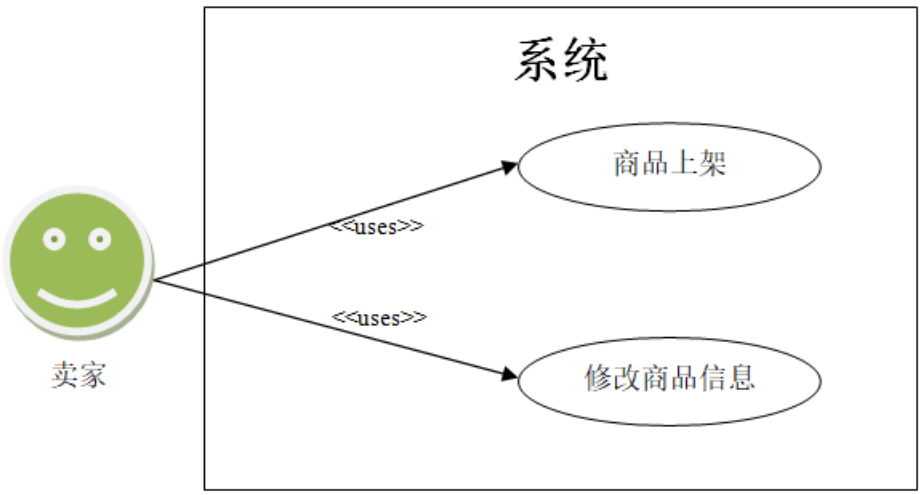


用例名称	卖家修改订单状态		
用例编号	SRS-17		
用例简介	卖家对订单状态进行修改。		
优先级	High		
前置条件	有用户已对卖家下订单，卖家可修改订单状态。		
后置条件	显示订单状态修改为：XX，后跳转到该订单信息页面。		
操作流程	*步骤*	*触发者*	*描述*
	1	卖家	查询用户对自己所下的订单。
	2	系统	从数据库调出订单信息。
	3	卖家	根据货物库存对订单状态进行修改。
	4	系统	数据库记录修改后的订单状态（撤销、付款、发货、延后）。
替代流程			
例外流程			
约束条件			
输入及约束	商品信息（从订单所对应的外键商品表中调取信息）。 卖家用户名（从订单表中调出信息）。		

	用户信息（从数据库调出发货地址、手机号等）。 修改订单状态：选择输入，分为 4 种状态（撤销、付款、发货、延后）。
用例名称	卖家修改订单信息

用例名称	卖家修改订单信息		
用例编号	SRS-18		
用例简介	卖家对订单信息进行修改。		
优先级	High		
前置条件	该订单商品为此用户所卖才能修改。		
后置条件	跳转到该订单信息页面。		
操作流程	*步骤*	*触发者*	*描述*
	1	卖家	查询用户对自己所下的订单。
	2	系统	从数据库调出订单信息。
	3	卖家	将想要修改的订单信息进行修改。
	4	系统	调取数据库记录，判断修改后订单是否合理（货物库存是否足够，且商品类型不能修改，数量只能减少不能增加，若要增加只能卖家与买家再次进行一次订单交易），修改后保存。
替代流程			
例外流程			
约束条件			
输入及约束	商品信息（从订单所对应的外键商品表中调取信息）。 卖家用户名（从订单表中调出信息）。 用户信息（从数据库调出发货地址、手机号等）。		

2.2.9 商品管理

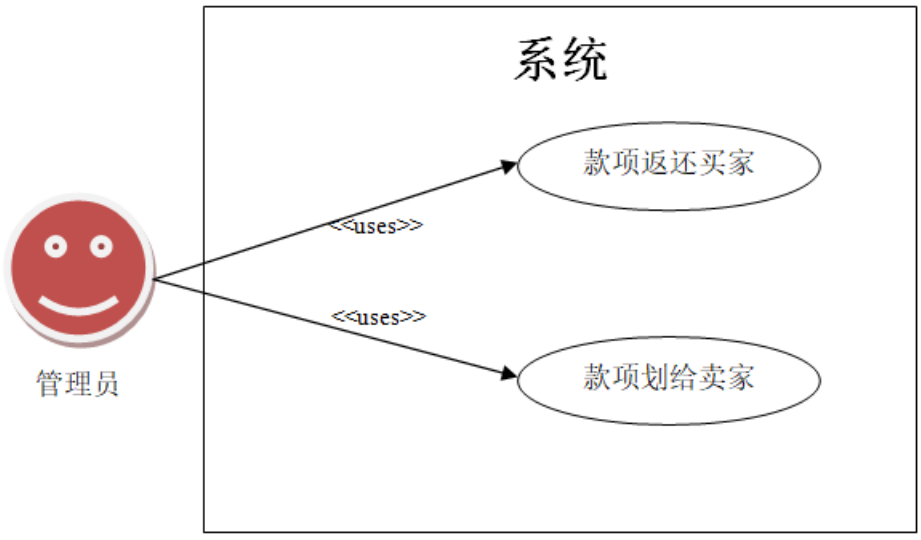


用例名称	商品上架		
用例编号	SRS-19		
用例简介	卖家发布要出售的商品的信息。		
优先级	High		
前置条件	用户是已注册成员。		
后置条件	跳转到该商品的信息页面。		
操作流程	*步骤*	*触发者*	*描述*
	1	卖家	发布商品各项信息。
	2	系统	进入数据库，添加商品信息。
替代流程			
例外流程			
约束条件			
输入及约束	买家用户名： 商品信息：		

用例名称	修改商品信息
用例编号	SRS-20

用例简介	修改卖家发布的商品信息。		
优先级	High		
前置条件	用户已登陆，且修改的是自己发布的商品。		
后置条件	跳转到该商品的信息页面。		
操作流程	*步骤*	*触发者*	*描述*
	1	卖家	修改商品信息。
	2	系统	进入数据库，修改商品信息。
替代流程			
例外流程			
约束条件			
输入及约束	买家用户名: 商品信息：		

2.2.10 划账管理



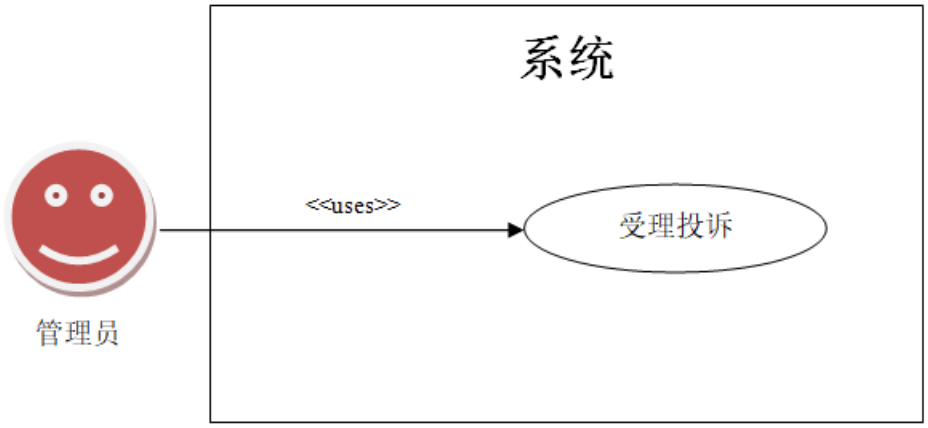
用例名称	管理员将款项返还买家
用例编号	SRS-21
用例简介	用户已付款给中介管理员，但因退货或货物丢失所导致的订单交易失败，用户可申请退款请求，管理员将用户转账资金返还买家。

优先级	High		
前置条件	用户已登陆，并且订单付款项为已付款、订单状态为撤销则为退货或货物未到，款项返还买家。		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	订单付款项为已付款且订单状态被撤销时用户可点击请求退款。
	2	管理员	核实交易情况，若属实则点击同意请求。
	3	系统	系统将用户在此订单所转账冻结的资金返还给用户的账户上，此项交易完成。
替代流程			
例外流程			
约束条件			
输入及约束			

用例名称	管理员将款项划给卖家		
用例编号	SRS-22		
用例简介	交易成功，管理员将买家已付款的款项划给卖家。		
优先级	High		
前置条件	订单付款项为已付款、订单状态为完成则视为交易成功，管理员将款项划给卖家。		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	系统	系统检测到某一订单付款项为已付款，订单状态为完成，则视为交易成功，提醒管理员转账。
	2	管理员	核实交易情况，若属实则点击同意转账。

	3	系统	系统将用户在此订单所转账冻结的资金划给该订单的卖家，此项交易完成。
替代流程			
例外流程			
约束条件			
输入及约束			

2.2.11 投诉管理



用例名称	受理投诉		
用例编号	SRS-23		
用例简介	受理用户的投诉。		
优先级	High		
前置条件	用户已登陆，且投诉的交易为下过的订单且订单状态为已完成。		
后置条件			
操作流程	*步骤*	*触发者*	*描述*
	1	用户	收到货物不满意，针对此订单发起投诉消息。
	2	管理员	受理、查看投诉。
替代流程			

例外流程	
约束条件	
输入及约束	

2.3 系统的数据需求

系统的数据需求包括如下几点：

2.3.1 数据录入和处理的准确性和实时性

数据的输入是否准确是数据处理的前提，错误的输入会导致系统输出的不正确和不可用，从而使系统的工作失去意义。

在系统中，数据的输入往往是大量的，因此系统要有一定的处理能力，以保证迅速的处理数据。

2.3.2 数据的一致性与完整性

由于系统的数据是共享的，所以如何保证这些数据的一致性，是系统必须解决的问题。要解决这一问题，要有一定的人员维护数据的一致性，在数据录入处控制数据的去向，并且要求对数据库的数据完整性进行严格的约束。

对于输入的数据，要为其定义完整性规则，如果不能符合完整性约束，系统应该拒绝该数据。

2.3.3 数据的共享与独立性

系统的数据是共享的。然而，从系统开发的角度上看，共享会给设计和调试带来困难。因此，应该提供灵活的配置，使各个分系统能够独立运行，而通过人工干预的手段进行系统数据的交换。这样，也能提供系统的强壮性。

2.4 软硬件环境需求

需求名称	详细要求
数据库	Oracle
操作系统	Windows7 及以上
浏览器	Windows 系统下的 IE8.0 及以上版本或 FireFox
处理器	推荐：Intel 服务器 CPU　Xeon 5355 2.66G
内存	推荐：1G 以上

2.5 产品质量需求

主要质量属性	详细要求
正确性	在系统设计和开发过程中，要充分考虑网站当前和将来可能承受的工作量，使系统的处理能力和响应时间能够满足企业对信息处理的需求。
性能，效率	网站的相应速度应该满足及时与用户进行交互的基本要求。占用客户端的资源也不应太多。
易用性	本网站面对广大用户，包括计算机水平较低的用户。因此，应该力求操作的简洁与已于接受，而且界面也应力求较好的交互性和友好性。
安全性	本网站涉及到较多的用户个人隐私信息，因此，较高的安全性应当是建站的一个基础。所有的工作的展开都应保证是在安全性的基础上来展开的。
可扩展性	在网站运行的过程中，网站应该随着用户的需求的不断改变而改变，要求网站具有良好的可扩展性以适应广大用户的需求。

3 数据库设计

3.1 数据库环境说明

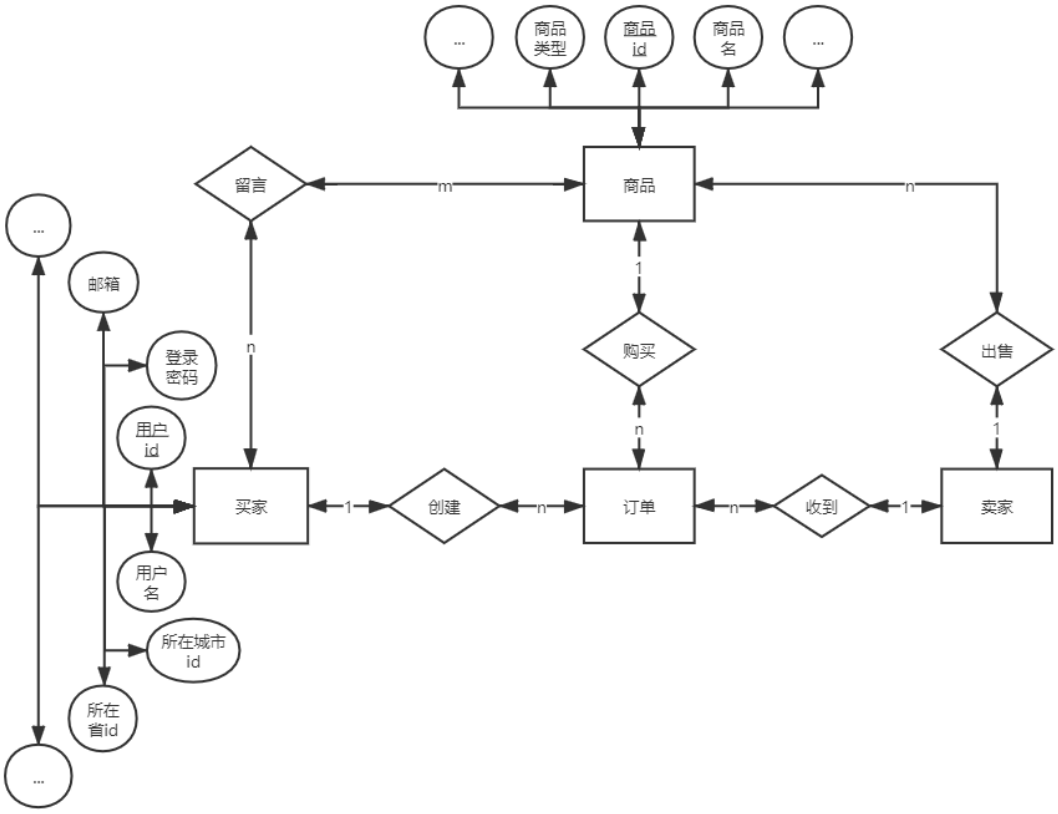
本系统采用 Oracle 数据库系统。

3.2 数据库命名规则

（1）表名以 tb_开头，当表名中含有两个以上的英文单词时，单词间不加空格，但是从第二个单词开始的单词首字母要大写。

（2）表中的字段名首字母小写，字段名中含有两个以上的英文单词时，单词间不加空格，但是从第二个单词开始的单词首字母要大写。

3.3 逻辑设计



3.4 物理设计

3.4.0 表汇总

表名	功能说明
tb_user	用户表
tb_product	产品表（商品表）

tb_productType	产品类型
tb_order	订单表
tb_userAccount	用户资金流水表（只允许新增记录，不允许修改、删除）
tb_review	评论表
tb_city	城市表
tb_province	省份表
tb_productSize	产品型号表
tb_productColor	产品颜色

3.4.1 tb_user 表(用户信息表)

表名	tb_user			
列名	描述	数据类型（精度范围）	允许空	约束条件
userID	用户 id	number	否	主键，自动增量(1)
name	用户名	varchar2		唯一约束
password	登录密码	varchar2(30)	否	
email	邮箱	varchar2(30)	是	
provinceID	所在省 id	number	是	
cityID	所在城市 id	number	是	外键，tb_city 表 cityID 项
telephone	手机号	char(11)	是	
sex	性别	char(3)	是	默认约束“男”
age	年龄	number	是	
birthday	生日	date	是	
money	账户余额	number	是	默认约束“0”
photo	头像路径	varchar2(500)	是	
dsp	个性签名	varchar2(200)	是	

qq	QQ 号码	varchar2(15)	是	
msn	msn 号码	varchar2(50)	是	
loveBook	爱好书籍	varchar2(200)	是	
loveMusic	爱好音乐	varchar2(200)	是	
loveMovie	爱好电影	varchar2(200)	是	
loveSport	爱好运动	varchar2(200)	是	
loveGame	爱好游戏	varchar2(200)	是	
补充说明				

3.4.2 tb_product 表(商品信息表)

表名	tb_product			
列名	描述	数据类型（精度范围）	允许空	约束条件
productID	商品 id	number	否	主键，自动增加(1)
userID	卖家 id	number	否	外键，tb_user 表 userID 项
productName	商品名	varchar2(50)	否	
typeID	商品类型 id	number	否	外键，tb_productType 表 productTypeID 项
price	商品价格	number	否	
photo	商品图片路径	varchar2(500)	是	
information	商品简介	varchar2(500)	是	
补充说明				

3.4.3 tb_productType 表（商品类型表）

表名	tb_productType			
列名	描述	数据类型（精度范	允	约束条件

		围）	许空	
productTypeID	商品类型 id	number	否	主键，自动增加(1)
productTypeName	商品名称	varchar2(100)	否	
补充说明				

3.4.4 tb_order 表（订单表）

表名	tb_order			
列名	描述	数据类型（精度范围）	允许空	约束条件
orderID	订单 id	number	否	主键，自动增加(1)
toID	买家 id	number	否	外键，tb_user 表 userID 项
colorID	颜色 id	number	否	外键，tb_productColor 表 productColorID 项
productID	商品 id	number	否	外键，tb_ product 表 productID 项
address	送货地址	varchar2(100)	否	
telephone	买家手机	char(11)	否	
orderState	订单状态	number	否	
count	商品数量	number	否	
price	总金额	number	否	
orderDate	订单日期	date	否	
consignmentDate	发货日期	date	否	
补充说明				

3.4.5 tb_userAccount 表（用户资金流水表）

表名	tb_userAccount			
列名	描述	数据类型（精度范围）	允许空	约束条件
userAccountID	资金流水表 id	number	否	主键，自动递增(1)
userID	用户 id	number	否	外键，tb_user 表 userID 项
orderID	订单 id	number	否	外键，tb_order 表 orderID 项
type	类型	number	否	
time	时间	date	否	
补充说明				

3.4.6 tb_review 表（评论表）

表名	tb_review			
列名	描述	数据类型（精度范围）	允许空	约束条件
reviewID	评论 id	number	否	主键，自动递增(1)
review	评论内容	varchar2(250)	否	
isReply	是否回复	boolean	否	默认值(false)
isDel	是否删除	boolean	否	默认为(false)
senderID	发表人 id	number	否	外键，tb_user 表 userID 项
productID	留言的商品 id	number	否	外键，tb_product 表 productID 项
addTime	发表时间	date	否	
replyID	回复人 id	number	是	

补充说明	是否回复：回复为 true，未回复为 false，默认为 false。 是否删除：删除为 true，未删除为 false，默认为 false。

3.4.7 tb_province 表（省份数据字典）

表名	tb_province			
列名	描述	数据类型（精度范围）	允许空	约束条件
provinceID	省份 id	number	否	主键，自动递增(1)
provinceName	省份名称	varchar2(30)	否	
补充说明				

3.4.8 tb_city 表（城市数据字典）

表名	tb_city			
列名	描述	数据类型（精度范围）	允许空	约束条件
cityID	城市 id	number	否	主键，自动递增(1)
provinceID	省份 id	number	否	外键，tb_province 表 provinceID 项
cityName	城市名称	varchar2(30)	否	
补充说明				

3.4.9 tb_productSize 表（商品型号表）

表名	tb_productSize			
列名	描述	数据类型（精度范围）	允许空	约束条件
productSizeID	商品型号 id	number	否	主键，自动递增(1)
productSize	商品型号	varchar2(20)	否	

productID	商品 id	number	否	外键，tb_product 表 productID 项
补充说明				

3.4.10 tb_productColor 表（商品颜色表）

表名	tb_productColor			
列名	描述	数据类型（精度范围）	允许空	约束条件
productColorID	商品颜色 id	number	否	主键，自动增量(1)
productColor	商品颜色	varchar2(20)	否	
stockpile	商品库存	number	否	
productSizeID	商品型号 id	number	否	外键，tb_productSize 表 productSizeID 项
补充说明				

4 总结

实训练习了 Oracle 的用户管理、权限管理、表和视图的建立、约束的建立与删除、表的增删改查、表的连接以及 PL/SQL 流程控制语句：触发器、存储过程、函数。

附录：

注意:代码下附上图示

一、创建数据库

```
-- /*创建用户*/
conn sys/1@orcl as sysdba;
create user admin identified by admin;
create user usr identified by usr;
```

```
grant connect, resource, dba  to admin;

grant create table,create session  to admin;

conn admin/admin@orcl as sysdba;
```

```
Connected to Oracle Database 11g Enterprise Edition Release 11.2.0.1.0
Connected as sys@ORCL AS SYSDBA

SQL>
SQL> conn sys/1@orcl as sysdba;
Connected to Oracle Database 11g Enterprise Edition Release 11.2.0.1.0
Connected as sys@ORCL AS SYSDBA
SQL> create user admin identified by admin;

User created


SQL> create user usr identified by usr;

User created


SQL> grant connect, resource, dba  to admin;

Grant succeeded


SQL> grant create table,create session  to admin;

Grant succeeded


SQL> conn admin/admin@orcl as sysdba;
Connected to Oracle Database 11g Enterprise Edition Release 11.2.0.1.0
Connected as admin@ORCL AS SYSDBA
```

二、创建表

```
-- /*创建 tb_province 表*/

create table tb_province(

provinceID number primary key,

provinceName varchar2(30) not null

);
```

```
SQL>
SQL> create table tb_province(
  2  provinceID number primary key,
  3  provinceName varchar2(30) not null
  4  );

Table created
```

```
create sequence tb_province_provinceID_seq

increment by 1

start with 1
```

```
    nomaxvalue

    nominvalue
nocache;
```

```
SQL>
SQL> create sequence tb_province_provinceID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

```
    create or replace trigger tr_tb_province_provinceID

    before insert on tb_province

    for each row

    begin

    select tb_province_provinceID_seq.nextval into :new.provinceID from
dual;

    end;

    /
```

```
SQL> create or replace trigger tr_tb_province_provinceID
  2  before insert on tb_province
  3  for each row
  4  begin
  5  select tb_province_provinceID_seq.nextval into :new.provinceID from dual;
  6  end;
  7  /

Trigger created
```

```
    -- /*创建 tb_city 表*/

    create table tb_city(

    cityID number primary key,

    provinceID number references tb_province(provinceID) not null,

    cityName varchar2(30) not null,

    );
```

```
SQL> create table tb_city(
  2  cityID number primary key,
  3  provinceID number references tb_province(provinceID) not null,
  4  cityName varchar2(30) not null
  5  );

Table created
```

```
create sequence tb_city_cityID_seq

increment by 1

start with 1

nomaxvalue

nominvalue

nocache;
```

```
SQL> create sequence tb_city_cityID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

```
create or replace trigger tr_tb_city_cityID

before insert on tb_city

for each row

begin

select tb_city_cityID_seq.nextval into :new.cityID from dual;

end;

/
```

```
SQL> create or replace trigger tr_tb_city_cityID
  2  before insert on tb_city
  3  for each row
  4  begin
  5  select tb_city_cityID_seq.nextval into :new.cityID from dual;
  6  end;
  7  /

Trigger created
```

```
-- /*创建 tb_user 表*/

create table tb_user(
```

```
userID number primary key,
name varchar2(10) unique,
password varchar2(30) not null,
email varchar2(30),
provinceID number,
cityID number references tb_city(cityID),
telephone char(11),
sex char(3) default('男'),
age number ,
birthday date,
money number default 0,
photo varchar2(500),
dsp varchar2(200),
qq   varchar2(15),
msn varchar2(50),
loveBook varchar2(200),
loveMusi varchar2(200),
loveMovi varchar2(200),
loveSpor varchar2(200),
loveGame varchar2(200)
);
```

```
SQL> create table tb_user(
  2  userID number primary key,
  3  name varchar2(10) unique,
  4  password varchar2(30) not null,
  5  email varchar2(30),
  6  provinceID number,
  7  cityID number references tb_city(cityID),
  8  telephone char(11),
  9  sex char(3) default('男'),
 10  age number ,
 11  birthday date,
 12  money number default 0,
 13  photo varchar2(500),
 14  dsp varchar2(200),
 15  qq  varchar2(15),
 16  msn varchar2(50),
 17  loveBook varchar2(200),
 18  loveMusi varchar2(200),
 19  loveMovi varchar2(200),
 20  loveSpor varchar2(200),
 21  loveGame varchar2(200)
 22  );

Table created
```

```
create sequence tb_user_userID_seq

increment by 1

start with 1

nocache;
```

```
SQL> create sequence tb_user_userID_seq
  2  increment by 1
  3  start with 1
  4  nocache;

Sequence created
```

```
create or replace trigger tr_tb_user_userID

before insert on tb_user

for each row

begin

   select tb_user_userID_seq.nextval into :new.userID from dual;

end;
```

```
SQL> create or replace trigger tr_tb_user_userID
  2  before insert on tb_user
  3  for each row
  4  begin
  5     select tb_user_userID_seq.nextval into :new.userID from dual;
  6  end;
  7  /

Trigger created
```

```
-- /*创建 tb_productType 表*/


create table tb_productType(

productTypeID number primary key,

productTypeName    varchar2(100) not null

);
```

```
SQL> create table tb_productType(
  2  productTypeID number primary key,
  3  productTypeName    varchar2(100) not null
  4  );

Table created
```

```
create sequence tb_productType_productTypeID_seq

increment by 1

start with 1

nomaxvalue

nominvalue

nocache;
```

```
SQL> create sequence tb_prodType_prodTypeID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

```
create or replace trigger tr_tb_prodType_prodTypeID

before insert on tb_productType

for each row

begin

   select tb_prodType_prodTypeID_seq.nextval into :new.productTypeID
from dual;

end;

/
```

```
SQL> create or replace trigger tr_tb_prodType_prodTypeID
  2  before insert on tb_productType
  3  for each row
  4  begin
  5     select tb_prodType_prodTypeID_seq.nextval into :new.productTypeID from dual;
  6  end;
  7  /

Trigger created
```

```
-- /*创建 tb_product 表*/

create table tb_product(

productID number primary key,

userID number references tb_user(userID) not null,

productName   varchar2(50) not null,

typeID number references tb_productType(productTypeID) not null,

price number  not null,

photo varchar2(500)  ,

information varchar2(500)
```

```
);
```

```
SQL> create table tb_product(
  2  productID number primary key,
  3  userID number references tb_user(userID) not null,
  4  productName varchar2(50) not null,
  5  typeID number references tb_productType(productTypeID) not null,
  6  price number not null,
  7  photo varchar2(500)  ,
  8  information varchar2(500)
  9  );

Table created
```

```
create sequence tb_product_productID_seq

increment by 1

start with 1

nomaxvalue

nominvalue

nocache;
```

```
SQL> create sequence tb_product_productID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

```
create or replace trigger tr_tb_product_productID

before insert on tb_product

for each row

begin

select tb_product_productID_seq.nextval into :new.productID from
dual;

end;

/
```

```
SQL> create or replace trigger tr_tb_product_productID
  2  before insert on tb_product
  3  for each row
  4  begin
  5  select tb_product_productID_seq.nextval into :new.productID from dual;
  6  end;
  7  /

Trigger created
```

```
   -- /*创建 tb_productSize 表*/

   create table tb_productSize(

   productSizeID number primary key,

   productSize varchar2(20) not null,

   productID number references tb_product(productID) not null,

   );
```

```
SQL> create table tb_productSize(
  2  productSizeID number primary key,
  3  productSize varchar2(20) not null,
  4  productID number references tb_product(productID) not null
  5  );

Table created
```

```
   create sequence tb_productSize_productSizeID_seq

   increment by 1

   start with 1

   nomaxvalue

   nominvalue

   nocache;
```

```
    SQL> create sequence tb_prodSize_prodSizeID_seq
      2  increment by 1
      3  start with 1
      4  nomaxvalue
      5  nominvalue
      6  nocache;

    Sequence created
```

```
create or replace trigger tr_tb_user_userID
before insert on tb_user
for each row
begin
select tb_user_userID_seq.nextval into :new.userID from dual;
end;
/
```

```
SQL> create or replace trigger tr_tb_user_userID
  2  before insert on tb_user
  3  for each row
  4  begin
  5  select tb_user_userID_seq.nextval into :new.userID from dual;
  6  end;
  7  /

Trigger created
```

```
-- /*创建 tb_productColor 表*/
create table tb_productColor(
productColorID number primary key,
productColor varchar2(20) not null,
stockpile number not null,
productSizeID number references tb_productSize(productSizeID) not
null
);
```

```
SQL> create table tb_productColor(
  2  productColorID number primary key,
  3  productColor varchar2(20) not null,
  4  stockpile number not null,
  5  productSizeID number references tb_productSize(productSizeID) not null
  6  );

Table created
```

```
create sequence tb_prodColor_prodColorID_seq
increment by 1
```

```
start with 1

nomaxvalue

nominvalue

nocache;
```

```
SQL> create sequence tb_prodColor_prodColorID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

```
create or replace trigger tr_tb_prodColor_prodColorID

before insert on tb_productColor

for each row

begin

select                          tb_prodColor_prodColorID_seq.nextval
into :new.productColorID from dual;

end;

/
```

```
SQL> create or replace trigger tr_tb_prodColor_prodColorID
  2  before insert on tb_productColor
  3  for each row
  4  begin
  5  select tb_prodColor_prodColorID_seq.nextval into :new.productColorID from dual;
  6  end;
  7  /

Trigger created
```

```
-- /*创建 tb_order 表(orderState,orderID)*/

create table tb_order(

orderID number primary key,

toID number references tb_user(userID) not null,

colorID number references tb_productColor(productColorID) not null,

productI number references tb_product(productID) not null,
```

```
   addressvarchar2(100) not null,

   telephone char(11) not null,

   orderState number not null,

   count number not null,

   price number not null,

   orderDate date not null,

   consignmentDate date not null

   );
```

```
SQL> create table tb_order(
  2  orderID number primary key,
  3  toID number references tb_user(userID) not null,
  4  colorID number references tb_productColor(productColorID) not null,
  5  productI number references tb_product(productID) not null,
  6  address  varchar2(100) not null,
  7  telephone char(11) not null,
  8  orderState number not null,
  9  count number not null,
 10  price number not null,
 11  orderDate date not null,
 12  consignmentDate date not null
 13  );

Table created
```

```
   create sequence tb_order_orderID_seq

   increment by 1

   start with 1

   nomaxvalue

   nominvalue

   nocache;
```

```
SQL> create sequence tb_order_orderID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

```
   create or replace trigger tr_tb_order_orderID
```

```
before insert on tb_order

for each row

begin

select tb_order_orderID_seq.nextval into :new.orderID from dual;

end;

/
```

```
SQL> create or replace trigger tr_tb_order_orderID
  2  before insert on tb_order
  3  for each row
  4  begin
  5  select tb_order_orderID_seq.nextval into :new.orderID from dual;
  6  end;
  7  /

Trigger created
```

```
/*创建 tb_userAccount 表*/

create table tb_userAccount(

userAccountID number primary key,

userID number references tb_user(userID) not null,

orderIDnumber references tb_order(orderID) not null,

type number not null,

time date  not null

);
```

```
 SQL> create table tb_userAccount(
   2  userAccountID number primary key,
   3  userID number references tb_user(userID) not null,
   4  orderID  number references tb_order(orderID) not null,
   5  type number not null,
   6  time date  not null
   7  );

 Table created
```

```
create sequence tb_userAccount_userAccountID_seq

increment by 1
```

```
start with 1

nomaxvalue

nominvalue

nocache;
```

```
SQL> create sequence tb_Account_AccountID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

```
create or replace trigger tr_tb_Account_AccountID

before insert on tb_userAccount

for each row

begin

select tb_Account_AccountID_seq.nextval into :new.userAccountID from
dual;

end;

/
```

```
SQL> create or replace trigger tr_tb_Account_AccountID
  2  before insert on tb_userAccount
  3  for each row
  4  begin
  5  select tb_Account_AccountID_seq.nextval into :new.userAccountID from dual;
  6  end;
  7  /

Trigger created
```

```
-- /*创建 tb_review 表*/

create table tb_review(

reviewID number   primary key,

review  varchar2(250) not null,

isReply boolean default(false) not null,

isDel   boolean default(false) not null,
```

```
senderID number references tb_user(userID) not null,
productID  number references tb_product(productID) not null,
addTime date not null,
replyID number
);
```

```
SQL> create table tb_review(
  2  reviewID number  primary key,
  3  review  varchar2(250)  not null,
  4  isReply   char default(0) not null,
  5  isDel   char default(0) not null,
  6  senderID number references tb_user(userID) not null,
  7  productID  number references tb_product(productID) not null,
  8  addTime date not null,
  9  replyID number
 10  );

Table created
```

```
create sequence tb_review_reviewID_seq
increment by 1
start with 1
nomaxvalue
nominvalue
nocache;
```

```
SQL> create sequence tb_review_reviewID_seq
  2  increment by 1
  3  start with 1
  4  nomaxvalue
  5  nominvalue
  6  nocache;

Sequence created
```

三、添加约束

```
/*给每个表添加约束*/
alter table tb_city add constraint fk_cityID_provinceID foreign
key(provinceID) references tb_province(provinceID) on delete cascade;
alter table tb_user add constraint fk_userID_cityID foreign
key(cityID) references tb_city(cityID) on delete cascade;
```

```
    alter table tb_product add constraint fk_productID_userID foreign
key(userID) references tb_user(userID) on delete cascade;
    alter table tb_product add constraint fk_productID_productTypeID
foreign key(typeID) references  tb_productType(productTypeID) on delete
cascade;
    alter table tb_productSize add constraint fk_SizeID_productID foreign
key(productID) references  tb_product(productID) on delete cascade;
    alter table tb_productColor add constraint fk_ColorID_productSizeID
foreign key(productSizeID) references  tb_productSize(productSizeID) on
delete cascade;
    alter table tb_userAccount add constraint fk_AccountID_userID foreign
key(userID) references  tb_user(userID) on delete cascade;
    alter table tb_userAccount add constraint fk_AccountID_orderID
foreign key(orderID) references  tb_order(orderID) on delete cascade;
    alter table tb_review add constraint fk_reviewID_userID foreign
key(senderID) references  tb_user(userID) on delete cascade;
    alter table tb_review add constraint fk_reviewID_productID foreign
key(productID) references  tb_product(productID) on delete cascade;
```

四、基础数据(商品类型+省份+城市)添加

```
    -- /*基础数据(商品类型+省份+城市)添加*/
    -- /*tb_productType 表添加商品类型*/
    insert                    into                    tb_productType
values(tb_prodType_prodTypeID_seq.nextval,'食品');
    insert                    into                    tb_productType
values(tb_prodType_prodTypeID_seq.nextval,'生活用品');
    insert                    into                    tb_productType
values(tb_prodType_prodTypeID_seq.nextval,'服饰');
    insert                    into                    tb_productType
```

```
values(tb_prodType_prodTypeID_seq.nextval,'书籍');

    insert                    into                    tb_productType
values(tb_prodType_prodTypeID_seq.nextval,'数码');
```

```
SQL> insert into tb_productType values(tb_prodType_prodTypeID_seq.nextval,'食品');
1 row inserted

SQL> insert into tb_productType values(tb_prodType_prodTypeID_seq.nextval,'生活用品')
1 row inserted

SQL> insert into tb_productType values(tb_prodType_prodTypeID_seq.nextval,'服饰');
1 row inserted

SQL> insert into tb_productType values(tb_prodType_prodTypeID_seq.nextval,'书籍');
1 row inserted

SQL> insert into tb_productType values(tb_prodType_prodTypeID_seq.nextval,'数码');
1 row inserted
```

```
    -- /*tb_province 表添加省份名称*/
    insert into tb_province values(tb_province_provinceID_seq.nextval,'
山东');
    insert into tb_province values(tb_province_provinceID_seq.nextval,'
山西');
    insert into tb_province values(tb_province_provinceID_seq.nextval,'
河南');
    insert into tb_province values(tb_province_provinceID_seq.nextval,'
河北');
    insert into tb_province values(tb_province_provinceID_seq.nextval,'
四川');
```

```
SQL> insert into tb_province values(tb_province_provinceID_seq.nextval,'山东');
1 row inserted

SQL> insert into tb_province values(tb_province_provinceID_seq.nextval,'山西');
1 row inserted

SQL> insert into tb_province values(tb_province_provinceID_seq.nextval,'河南');
1 row inserted

SQL> insert into tb_province values(tb_province_provinceID_seq.nextval,'河北');
1 row inserted

SQL> insert into tb_province values(tb_province_provinceID_seq.nextval,'四川');
1 row inserted
```

```
-- /*tb_city 表给山东省添加城市名称*/
insert into tb_city values(tb_city_cityID_seq.nextval,1,'济南');
insert into tb_city values(tb_city_cityID_seq.nextval,1,'青岛');
insert into tb_city values(tb_city_cityID_seq.nextval,1,'潍坊');
insert into tb_city values(tb_city_cityID_seq.nextval,1,'烟台');
insert into tb_city values(tb_city_cityID_seq.nextval,1,'日照');
```

```
SQL> insert into tb_city values(tb_city_cityID_seq.nextval,1,'济南');
1 row inserted

SQL> insert into tb_city values(tb_city_cityID_seq.nextval,1,'青岛');
1 row inserted

SQL> insert into tb_city values(tb_city_cityID_seq.nextval,1,'潍坊');
1 row inserted

SQL> insert into tb_city values(tb_city_cityID_seq.nextval,1,'烟台');
1 row inserted

SQL> insert into tb_city values(tb_city_cityID_seq.nextval,1,'日照');
1 row inserted
```

五、用户数据(注册+信息完善)添加

```
/*用户数据(注册+信息完善)添加*/
```

```
/*tb_user 表添加用户信息*/
insert into tb_user values(tb_user_userID_seq.nextval,'张三','zs','zs@163.com',1,1,'12300000000','男',11,to_date('1999-7-15','yyyy-MM-dd'),500,'1.jpg','无个性不签名','999999','44444','book1','music1','movie1','sport1','game1');
insert into tb_user values(tb_user_userID_seq.nextval,'李四','ls','ls@163.com',1,2,'12311111111','男',22,to_date('1999-3-11','yyyy-MM-dd'),500,'2.jpg','无个性不签名','888888','44444','book2','music2','movie2','sport2','game2');
insert into tb_user values(tb_user_userID_seq.nextval,'王五','ww','ww@163.com',1,3,'12333333333','男',33,to_date('1993-7-28','yyyy-MM-dd'),500,'3.jpg','无个性不签名','777777','44444','book3','music3','movie3','sport3','game3');
insert into tb_user values(tb_user_userID_seq.nextval,'赵六','zl','zl@163.com',1,4,'12344444444','女',44,to_date('1996-6-12','yyyy-MM-dd'),500,'4.jpg','无个性不签名','666666','44444','book4','music4','movie4','sport4','game4');
insert into tb_user values(tb_user_userID_seq.nextval,'刘七','lq','lq@163.com',1,5,'12355555555','女',55,to_date('1999-7-21','yyyy-MM-dd'),500,'5.jpg','无个性不签名','555555','44444','book5','music5','movie5','sport5
```

```
SQL> insert into tb_user values(tb_user_userID_seq.nextval,'张三','zs','zs@163.com',1,1,'12300000000','男',11,to_date(
1 row inserted

SQL> insert into tb_user values(tb_user_userID_seq.nextval,'李四','ls','ls@163.com',1,2,'12311111111','男',22,to_date(
1 row inserted

SQL> insert into tb_user values(tb_user_userID_seq.nextval,'王五','ww','ww@163.com',1,3,'12333333333','男',33,to_date(
1 row inserted

SQL> insert into tb_user values(tb_user_userID_seq.nextval,'赵六','zl','zl@163.com',1,4,'12344444444','女',44,to_date(
1 row inserted

SQL> insert into tb_user values(tb_user_userID_seq.nextval,'刘七','lq','lq@163.com',1,5,'12355555555','女',55,to_date(
1 row inserted
```

六、商品信息(商品信息+型号+颜色)添加

```
-- /*商品信息(商品信息+型号+颜色)添加*/
-- /*tb_product 表添加商品信息*/
insert into tb_product values(tb_product_productID_seq.nextval,1,'面包',2,100,'2.jpg','jianjie');
insert into tb_product values(tb_product_productID_seq.nextval,2,'胶带',3,200,'3.jpg','jianjie');
insert into tb_product values(tb_product_productID_seq.nextval,3,'鞋子',4,300,'4.jpg','jianjie');
insert into tb_product values(tb_product_productID_seq.nextval,4,'杂志',5,500,'5.jpg','jianjie');
insert into tb_product values(tb_product_productID_seq.nextval,5,'手机',1,600,'1.jpg','jianjie');
```

```
SQL> insert into tb_product values(tb_product_productID_seq.nextval,1,'面包',2,100,'2.jpg','jianjie');
1 row inserted

SQL> insert into tb_product values(tb_product_productID_seq.nextval,2,'胶带',3,200,'3.jpg','jianjie');
1 row inserted

SQL> insert into tb_product values(tb_product_productID_seq.nextval,3,'鞋子',4,300,'4.jpg','jianjie');
1 row inserted

SQL> insert into tb_product values(tb_product_productID_seq.nextval,4,'杂志',5,500,'5.jpg','jianjie');
1 row inserted

SQL> insert into tb_product values(tb_product_productID_seq.nextval,5,'手机',1,600,'1.jpg','jianjie');
1 row inserted
```

```
-- /*tb_productSize 表添加商品型号*/
insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type1',1);
insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type2',2);
insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type3',3);
insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type4',4);
insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type5',5);
```

```
SQL> insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type1',1);
1 row inserted

SQL> insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type2',2);
1 row inserted

SQL> insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type3',3);
1 row inserted

SQL> insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type4',4);
1 row inserted

SQL> insert into tb_productSize values(tb_prodSize_prodSizeID_seq.nextval,'type5',5);
1 row inserted
```

```
-- /*tb_productColor 表添加商品型号*/
insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'红',10,5);
insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'橙',20,4);
insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'蓝',30,3);
insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'绿',40,2);
insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'紫',50,1);
```

```
SQL> insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'红',10,5);
1 row inserted

SQL> insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'橙',20,4);
1 row inserted

SQL> insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'蓝',30,3);
1 row inserted

SQL> insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'绿',40,2);
1 row inserted

SQL> insert into tb_productColor values(tb_prodColor_prodColorID_seq.nextval,'紫',50,1);
1 row inserted
```

七、删除用户信息(使用 plsql)

```
-- /*删除用户为 XX 的用户信息*/
declare
   v_id number;
begin
   v_id:='&要删除的用户 id';
  delete from tb_user where  userID=v_id;
  --   alter   table   tb_user   disable   novalidate   constraint
constraint_name;
  if sql%notfound then
     dbms_output.put_line('删除失败!');
  else
    dbms_output.put_line('删除成功!');
  end if;
end;
/
```

删除成功!

八、商品信息查询

```
-- /*商品信息查询*/
-- /*单表查询*/
-- /*查询 tb_product 表中商品 id 为的商品信息*/
select * from tb_product where productID=1;

-- /*查询 tb_productSize 表中商品型号 id 为的型号信息*/
select * from tb_productSize where productSizeId=2;

-- /*查询 tb_productColor 表中商品色号 id 为的颜色与库存信息*/
select * from tb_productColor where productColorID=3;
```

```
SQL> select * from tb_product where productID=1;

 PRODUCTID     USERID PRODUCTNAME                                          TYPEID      PRICE PHOTO
---------- ---------- -------------------------------------------------- ---------- ---------- ----------
         1          1 面包                                                        2        100 2.jpg

SQL> select * from tb_productSize where productSizeId=2;

PRODUCTSIZEID PRODUCTSIZE           PRODUCTID
------------- -------------------- ----------
            2 type2                         2

SQL> select * from tb_productColor where productColorID=3;

PRODUCTCOLORID PRODUCTCOLOR          STOCKPILE PRODUCTSIZEID
-------------- -------------------- ---------- -------------
             3 蓝                            30             3
```

```
-- /*多表查询*/
-- /*用户在商城以商品类型查询一件商品的名称*/
select productName from tb_product tp join tb_productType tpt on
tp.typeId=tpt.productTypeId where productTypeName='服饰';

-- /*用户在商城以商品名称查询一件商品的所有信息*/
select * from tb_product tp
join tb_productType tpt
on tp.typeId=tpt.productTypeId
join tb_productSize tps
```

```
on tps.productID=tp.productID
join tb_productColor tpc
on tps.productSizeID=tpc.productSizeID;
```

```
SQL> select productName from tb_product tp join tb_productType tpt on tp.typeId=tpt.productTypeId where t

PRODUCTNAME
--------------------------------------------------
胶带

SQL>
SQL> select * from tb_product tp
  2  join tb_productType tpt
  3  on tp.typeId=tpt.productTypeId
  4  join tb_productSize tps
  5  on tps.productID=tp.productID
  6  join tb_productColor tpc
  7  on tps.productSizeID=tpc.productSizeID;

 PRODUCTID     USERID PRODUCTNAME                                          TYPEID      PRICE PHOTO
---------- ---------- -------------------------------------------------- ---------- ---------- ----------
         5          5 手机                                                        1        600 1.jpg
         4          4 杂志                                                        5        500 5.jpg
         3          3 鞋子                                                        4        300 4.jpg
         2          2 胶带                                                        3        200 3.jpg
         1          1 面包                                                        2        100 2.jpg
```

九、用户信息查询

```
-- /*用户信息查询*/
-- /*用户在商城可以查询已知用户(或卖家)的基本信息*/
declare
  cursor c_user is
    select * from tb_user;
  --定义记录变量
  user_info c_user%rowtype;
begin
  open c_user;--打开游标
  loop
    FETCH c_user
      INTO user_info;--获取记录值
    EXIT WHEN c_user%NOTFOUND;
    dbms_output.put_line('学号: ' || user_info.userID ||
 ',姓名: ' ||  user_info.name||
     ',密码: ' ||  user_info.password||
```

```
    ',email: ' ||  user_info.email||
    ',省份: ' ||  user_info.provinceID||
    ',城市: ' ||  user_info.cityID||
    ',电话: ' ||  user_info.telephone||
    ',性别: ' ||  user_info.sex||
    ',年龄: ' ||  user_info.age||
    ',生日: ' ||  user_info.birthday||
    ',余额: ' ||  user_info.money||
    ',照片: ' ||  user_info.photo||
    ',签名: ' ||  user_info.dsp||
    ',qq: ' ||  user_info.qq||
    ', 喜欢的书: ' ||  user_info.loveBook||
    ', 喜欢的音乐: ' ||  user_info.loveMusi||
    ', 喜欢的电影: ' ||  user_info.loveMovi||
    ', 喜欢的运动: ' ||  user_info.loveSpor||
    ', 喜欢的游戏: ' ||  user_info.loveGame
     );
  end loop;
  close c_user;--关闭游标
end;
/
```

```
学号: 1,姓名: 张三,密码: zs,email: zs@163.com,省份: 1,城市: 1,电话: 12300000000,性别: 男 ,年龄
学号: 2,姓名: 李四,密码: ls,email: ls@163.com,省份: 1,城市: 2,电话: 12311111111,性别: 男 ,年龄
学号: 3,姓名: 王五,密码: ww,email: ww@163.com,省份: 1,城市: 3,电话: 12333333333,性别: 男 ,年龄
学号: 4,姓名: 赵六,密码: zl,email: zl@163.com,省份: 1,城市: 4,电话: 12344444444,性别: 女 ,年龄
学号: 5,姓名: 刘七,密码: lq,email: lq@163.com,省份: 1,城市: 5,电话: 12355555555,性别: 女 ,年龄
```

十、用户下订单并确认付款

```
-- /*用户下订单并确认付款*/
-- /*用户***想要购买卖家***的产品颜色-大小-产品名*/
```

-- /*用户下订单并确认付款*/

```
declare
  this_userId number;
  this_productColorId number;
  this_productId number;
  this_count number;
  this_stockPile number;
  this_price number;
  this_money number;
  this_phone char(11);
  Ordernum number;
begin
  this_userid:='&买家 id';
  this_productColorId:='&颜色 Id';
  this_productId:='&商品 id';
  this_count:='&数量';
  select telephone into this_phone from tb_user where
userId=this_userId;
  select price into this_price from tb_product where
productId=this_productId;
-- /*用户****将订单信息填入到 tb_order 表中*/
   --orderState 订单状态为用户选择填入下订单(买家付款后自动转为已付款,买家付
款后卖家可选择修改为延后,卖家发货后选择修改为发货,买家未付款时卖家可选择修改为撤销,
买家收到货物后可选择修改为完成)
insert into tb_order values
(tb_order_orderID_seq.nextval,this_userid,this_productcolorid,thi
s_productid,'          默          认          地          址
',this_phone,0,this_count,(this_count*this_price),sysdate,(sysdate+3)
);
-- /*下订单后减少商品库存*/
```

```
    select stockpile into this_stockPile from tb_productColor where
productColorId=this_productColorId;
      if this_stockPile>this_count then
         begin
          update tb_productColor set stockPile=stockPile-this_count
where productColorId=this_productColorId;
         end;
      else
          dbms_output.put_line('错误! 库存不足');
      end if;
    -- /*确认付款后减少用户余额*/
    select money into this_money from tb_user where userId=this_userId;
      if this_money>this_count*this_price then
        begin
          update tb_user set money=this_money-this_count*this_price where
userId=this_userid;
          dbms_output.put_line('交易成功');
        end;
      else
         dbms_output.put_line( '错误! 账户余额不足');
      end if;
      commit;
    end;

    --select * from tb_order;
```

```
交易成功
```

十一、买家对订单信息进行修改（包括发货后的订单状态修改及修改商品

颜色）

```
    -- /*卖家对订单信息进行修改*/
    -- /*假设买家***想修改商品的颜色，则需要对相应颜色的库存进行修改*/
    --/*修改信息是各项信息均为可选择项，如不想修改某项信息，则选择以前的选项，但是会将数据重复的修改一次，即用相同的数据来覆盖之前的数据*/
    -- /*除此之外卖家发货后可将订单状态改为发货*/
    -- /*定义变量,用于存储本次订单的物品单价*/
    select * from tb_order;
    declare
       this_orderId number;
       this_address varchar2(20);
       this_telephone char(11);
       this_orderState number;
       this_colorId number;
       this_price number;
       this_count number;
       this_bef_price number;
       this_old_productColorId number;
    begin
       this_orderId:=&订单编号Id;
    -- /*修改tb_order表中的送货地址*/
      this_address:='&送货地址';
      update tb_order set address=this_address where
orderId=this_orderId;
      dbms_output.put_line('送货地址已修改');
    -- /*修改tb_order表中的联系方式*/
      this_telephone:='&联系方式';
      update tb_order set telephone=this_telephone where
orderId=this_orderId;
```

```
      dbms_output.put_line('联系方式已修改');
    -- /*修改tb_order表中的状态*/
    --若卖家因故不能发货，但已付款则卖家可修改订单状态为'延后'
     this_orderState:=&修改状态;
      update tb_order set orderState=this_orderState where
orderId=this_orderId;
      dbms_output.put_line('订单状态已修改');
    -- /*修改tb_order表中的商品颜色*/
      select colorId into this_old_productColorId from tb_order where
orderId=this_orderId;
      select count into this_count from tb_order where
orderId=this_orderId;
      this_colorId:='&颜色id';
      update tb_order set colorId=this_colorId where
orderId=this_orderId;

    -- /*修改tb_order表中的商品总价*/
    --  --若和买家商议后买家同意降价，但商品信息上还是不修改的，就可以在订单总价上
修改，付款后卖家修改总价，差价打回买家账户余额内

      select price into this_bef_price from tb_order where
orderId=this_orderId;
      this_price:='&当前价格';

    -- /*将与之前的差价加回买家账户余额内*/
    -- /*修改tb_user表中买家的账户余额*/
      update tb_user set money=money+this_bef_price-this_price where
userId=(select toId from tb_order where orderId=this_orderId);
      dbms_output.put_line('商品总价已修改');
```

```
      dbms_output.put_line('操作成功');

    -- /*修改颜色后要将之前选择的颜色库存加一再将修改后的颜色库存减一*/
      update tb_productcolor set stockPile =stockPile-this_count where
productColorId=this_orderId;
      update tb_productColor set stockPile=stockPile+this_count where
productColorId=this_old_productColorId;
      dbms_output.put_line('商品颜色已修改');

      commit;

      exception
      when no_data_found then
       dbms_output.put_line('数据未找到，操作失败');
      rollback;
    end;
```

```
送货地址已修改
联系方式已修改
订单状态已修改
商品总价已修改
操作成功
商品颜色已修改
```

十二、订单完成进行转账

```
    -- /*订单完成进行转账*/
    -- /*定义变量,用于存储流水表的用户 ID*/
    declare
      this_userId number;
      this_sellerId number;
      this_price number;
      this_orderId number;
```

```
    begin
      this_orderId:='&订单号';

    -- /*买家收到商品后，修改 tb_order 表中的状态*/
      update tb_order set orderState=3 where orderId=this_orderId;
      dbms_output.put_line('订单状态已完成');

    -- /*订单状态为'完成'后建立用户资金流水表一条卖家出售商品收入记录*/
      select userId into this_sellerId from tb_product where productId=
      (select productId from tb_order where orderId=this_orderId);
      insert into tb_useraccount
values(tb_Account_AccountID_seq.nextval,this_sellerid,this_orderId,1,
sysdate);
      -- dbms_output.put_line('用户资金流水表插入一条卖家出售商品收入记录');

    -- /*交易成功后卖家账户余额增加*/
       select price into this_price from tb_order where
orderId=this_orderId;
       update tb_user set money=money+this_price where
userId=this_sellerId;
       dbms_output.put_line('卖家账户余额增加');
       dbms_output.put_line('操作成功');
       COMMIT;
    exception
    when no_data_found then
      dbms_output.put_line('数据未找到，操作失败');
    rollback;
    end;
/
```

```
订单状态已完成
用户资金流水表插入一条卖家出售商品收入记录
卖家账户余额增加
操作成功
```

十三、货物丢件进行退款

```
-- /*货物丢件进行退款*/
-- /*前提条件:订单为发货状态，但物流出现丢件情况，由物流赔偿卖家，卖家确认丢件情况发生后，可进行退款操作*/
-- /*定义变量,用于存储流水表的用户ID*/
declare
  this_sellerId number;
  this_userId number;
  this_orderId number;
  this_price number;
begin
  this_orderId:='&订单编号';
-- /*卖家确认商品丢件后，修改tb_order表中的状态*/
  update tb_order set orderState=5 where orderId=this_orderId;
  dbms_output.put_line('订单状态:撤销');
-- /*订单状态为'撤销'后建立用户资金流水表一条买家商品退款收入记录*/
  select toid into this_userId from tb_order where orderId=this_orderId;
  insert into tb_useraccount values(tb_Account_AccountID_seq.nextval,this_userId,this_orderId,1,sysdate);
  dbms_output.put_line('用户资金流水表插入一条买家商品退款收入记录');
-- /*撤销成功后买家账户余额恢复*/
  select price into this_price from tb_order where orderId=this_orderId;
```

```
      update tb_user set money=money+this_price where userId=this_userId;
      dbms_output.put_line('买家账户余额恢复');
      dbms_output.put_line('操作成功');
      commit;

    exception
    when no_data_found then
        dbms_output.put_line('数据未找到，操作失败');
        rollback;
    end;
/
```

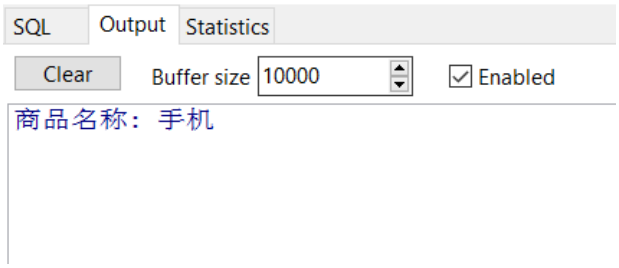
```
订单状态:撤销
用户资金流水表插入一条买家商品退款收入记录
买家账户余额恢复
操作成功
```

十四、视图

```
    --/*多表查询*/
    --/*用户在商城以商品类型查询一件商品的名称*/
    --drop view prodname_view;
    create view prodname_view
    as
       select * from tb_product tp ,tb_productType tpt where
tp.typeid=tpt.producttypeid;

    declare
       name varchar2(20);
       dname varchar2(20);
       cursor cur(xname varchar2)
       is
         select productName from prodname_view where
```

```
productTypeName=xname;
    begin
      dname:='&商品类型';
      open cur(dname);
        loop
          fetch cur into name;
          exit when cur%notfound;
          dbms_output.put_line('商品名称: '||name);
        end loop;
     close cur;
    end;
```




```
    --/*用户在商城以商品名称查询一件商品的所有信息*/
    --drop view allinfo_view;
    create view allinfo_view
    as
       select productcolorid,productcolor,stockpile,

c.productsizeid,productsize,s.productid,userid,productname,typeid,pri
ce,photo,information
       from tb_productcolor c
       join tb_productsize s on c.productsizeid=s.productsizeid
       join tb_product p on p.productid=s.productid;

    declare
```

```
dm  allinfo_view%rowtype;
pname varchar2(50);
cursor cur_t (product_name in varchar2)
is
 select *  from allinfo_view
   where productname=product_name;
begin
   pname:='&商品名称';
   open cur_t(pname);
   loop
   fetch cur_t into dm;
   exit when cur_t%notfound;
    dbms_output.put_line(
    '商品id：'||dm.productid||
    '商品名字：'||dm.productname||
    '商品颜色id：'||dm.productcolorid||
    '商品颜色：'||dm.productcolor||
    '商品库存：'||dm.stockpile||
    '商品型号id：'||dm.productsizeid||
    '商品型号：'||dm.productsize||
    '商品卖家id：'||dm.userid||
    '商品类型id：'||dm.typeid||
    '商品价格：'||dm.price||
    '商品图片路径：'||dm.photo||
    '商品简介：'||dm.information
   );
    end loop;
   close cur_t;
exception
```

```
   when no_data_found then
   dbms_output.put_line('无数据! ');
end;
/
```

```
商品id: 5商品名字: 手机商品颜色id: 1商品颜色: 红商品库存: 10商品型号
```

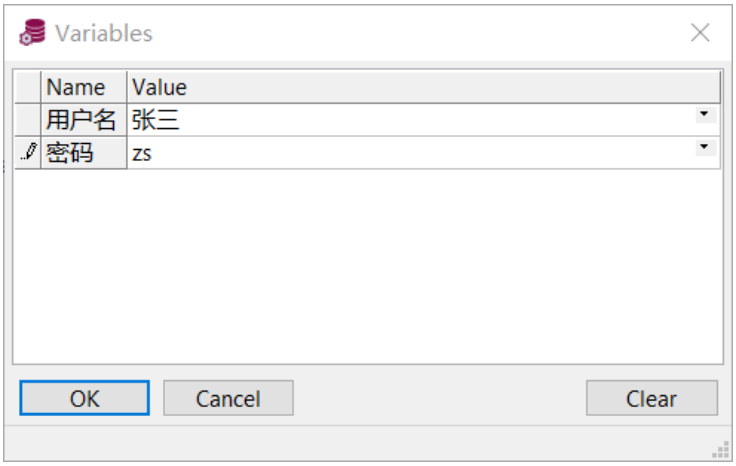
```
--/*用户信息查询*/
--/*用户在商城可以查询已知用户(或卖家)的基本信息*/
declare
   userId0 tb_user.userId%type;
   user tb_user%rowtype;
begin
  userId0:='&用户ID';
  select * into user from tb_user where userId=userId0;
  dbms_output.put_line(
   '用户id: '||user.userid||
   ', 用户名: '||user.name||
   ', 邮箱: '||user.email||
   ', 所在省份id: '||user.provinceid||
   ', 所在城市id: '||user.cityid||
   ', 手机号: '||user.telephone||
   ', 性别: '||user.sex||
   ', 年龄: '||user.age||
   ', 生日: '||user.birthday
   );
exception
   when no_data_found then
   dbms_output.put_line('没有发现您要找的数据! ');
end;--/*用户在商城可以查询已知用户(或卖家)的基本信息*/
```

```
/
```

用户id：1，用户名：张三，邮箱：zs@163.com，所在省份id：1，所在城市id：1，手机号

十五、用户登录

```
--/*用户登录(应用存储过程)*/
--自动登录
create or replace procedure login(login_name in varchar2,login_password in varchar2)
as
  huoqu_password varchar2(30);
begin
  select password into huoqu_password from tb_user where name=login_name;
  if huoqu_password=login_password then
    dbms_output.put_line(login_name||',欢迎登陆！');
  else
    dbms_output.put_line('密码错误！');
  end if;
  EXCEPTION
     when no_data_found then
       dbms_output.put_line('该用户尚未注册！');
end login;
/
begin
  login('张三','zs');
end;
```




```
--手动登录
declare
   login_name varchar2(10);
   login_password varchar2(30);
   get_password varchar2(30);
begin
   login_name:='&用户名';
   login_password:='&密码';
  select password into get_password from tb_user where
name=login_name;
  if get_password=login_password then
    dbms_output.put_line(login_name||',欢迎登陆! ');
  else
    dbms_output.put_line('密码错误! ');
  end if;
  EXCEPTION
     when no_data_found then
       dbms_output.put_line('该用户尚未注册! ');
end login;
```

```
张三,欢迎登陆!
```

成绩评分表

序号	评价内容	评价等级				
1	设计思路阐述清晰，格式符合要求	优 □	良 □	中 □	差 □	不及格 □
2	结构严谨，逻辑性强，语言表达准确	优 □	良 □	中 □	差 □	不及格 □
3	文字通顺，技术用语准确，基本概念清楚	优 □	良 □	中 □	差 □	不及格 □
4	基础理论知识扎实，回答问题有理论根据	优 □	良 □	中 □	差 □	不及格 □
5	有较强的计算机应用能力	优 □	良 □	中 □	差 □	不及格 □
6	功能齐全完善，能正确处理实验数据	优 □	良 □	中 □	差 □	不及格 □
7	系统设计与实现方法有技巧性、新有创新意识	优 □	良 □	中 □	差 □	不及格 □
8	有一定的理论或应用价值，设计思路新颖，对问题有较深刻的认识	优 □	良 □	中 □	差 □	不及格 □
9	学习态度端正，主动参与性强	优 □	良 □	中 □	差 □	不及格 □
10	与指导教师及时沟通，学风严谨务实，按期圆满完成规定的任务	优 □	良 □	中 □	差 □	不及格 □
备注						
总分		教师签字				