**湖南农业大学课程设计论文**

学 院：信息与智能科学技术学院

班 级：18计算机2班

学生姓名：肖思达、胡隆辉、史凡雨

学生学号：201841841309（肖思达）

2018440409118（胡隆辉）

201841842220（史凡雨）

课程设计题目：京东电商系统数据库设计

课程名称：数据库原理与应用

**京东系统的设计与实现**

作 者：肖思达、胡隆辉

指导老师：陈光仪

（湖南农业大学信智学院2018级计算机科学与技术专业2班，长沙410128）

**摘 要：**本文在分析基于多层体系结构的京东电商系统………………………………….。

(摘要围绕如下4个方面的内容来书写：

1.目的(Objective):简明指出此项工作的目的，研究的范围。

　2.方法(Methods):简要说明研究课题的基本做法。

　3.结果(Results):简要列出主要结果(需注明单位)、数据、统计学意义等，并说明其价值和局限性。

　4.结论(Conclusion):简要说明从该项研究结果取得的正确观点、理论意义或实用价值、推广前景。)

**关键词：**京东电商系统；数据库；关系表；B2C模式

前 言

随着科学技术的不断发展，全球化的不断推进，如今，信息时代已经到来。在互联网技术的快速发展下，商业经济活动发生了很大的变化，这是一个“互联网+”的时代，商业活动与互联网的结合，产生了电子商务，同时，电子商务也越来越融入到了人们的生活当中。而在电子商务模式中，B2C电子商务模式随着电子商务的发展而越发的重要，B2C电子商务模式，是企业针对个人发展的电子商务活动的总称，面向消费者，提供了在线网络购物的平台。

而京东电子商务系统更加是B2C电子商务模式的代表之一，通过京东商城系统这个平台，用户可以进行相应的在线购物活动，作为一个大型电商交易平台，我们应着重于销售与用户体验上，而用户所需求的功能包括对后台数据的检索，对买家、卖家信息的管理与展示等等。 作为买家，需要有自己的主页，包括购物车，收藏夹，搜索记录，浏览足迹等等，方便对种草的商品进行购买等操作，同时有购买后的服务包括物流记录、售后提交申请等功能。

因此，该数据库系统设计必须兼顾效率和性能；为了面对复杂多变的用户需求，还需要使系统架构简单清晰，各个功能模块间的耦合度低，容易水平扩展；同时，作为一个面对用户的开放型的平台，该数据库时刻运转，更加需要做到保证数据库的安全性和稳定性。

1.需求分析

需求分析分为三个部分：需求的文字表述、数据流图、数据字典。

**1.1** 调查用户需求

本系统的最终用户是广大人群，我们可以根据自己的电商行为比如网购等以及网络渠道获取所设计出的数据库的相关的需求，根据我们的日常生活经验，以及相应的调查得出用户的实际要求：

1.1.1 京东商城组织结构及对应业务活动

电子商务的四要素为：商城、消费者、产品和物流，因此，我们的所有功能行为都以这四要素为核心而展开。

1. 商城

对于商城，首先需要为消费者提供质优价廉的产品即商品；同时，当消费者进行下单后，商城根据用户购买的对应的商品，生成相应的订单及金额，当消费者下单并付款后，商城通过与物流公司建立的合作关系，将商品通过相应的物流公司发送到用户订单中填写的对应的收货地址；为了商城能够很多很好的商品，商城还需在足够吸引消费者的基础上吸引更多商家的入驻，只有在商家提供了物美价廉的商品，商城才能在商家和消费者之间形成良性循环。

关于商城的流程关系如图1所示

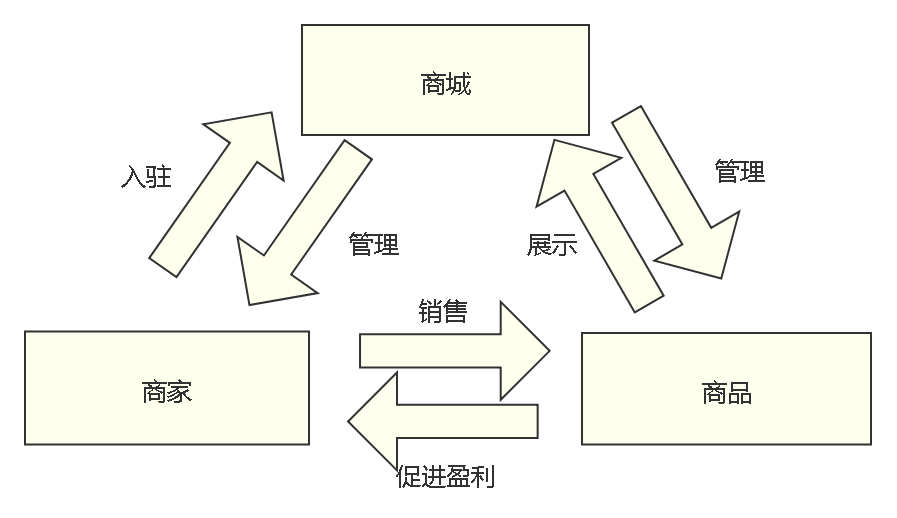


图1 商城流程关系图

1. 消费者

对于消费者而言，当消费者通过商城购买商品后，商城通过计算消费者添加到购物车的选中购买的商品对应的金额**以及商品的折扣**之后，得到消费者应该缴纳的金额，消费者在订单界面填写相应的信息（如收货地址、电话号码、收货人等相关信息、**支付方式、是否开发票**等信息），点击提交订单，支付对应的金额即可购买选中的商品；当消费者购买了相应商品后，还可以对该商品进行评价以及对店铺进行评价；当消费者收到了相应商品，如果对商品不满意，可以联系店铺选择根据订单编号选择退换货，物流公司根据消费者填写的订单信息确定好退货需要的运费，用户支付相应的运费即可选择退货。

商城

会员注册

会员登录

生成订单并配送

售后记录

退货换货

加入购物车

挑选商品

购买商品

收货评价

图2 会员流程关系图

1. 产品

对于产品，即商城中上架的商品，商城需要确定好对应商品的分类、商品的品牌、商品的规格、商品的库存、商品的价格、**商品的折扣活动**等信息，当消费者购买指定店铺的指定商品时，根据商品的库存以及相应价格还有相应折扣信息计算出相应商品对应的金额，否则消费者无法购买相应商品，**消费者此时可提醒商家进行补货。**

上架商品

商品折扣活动

库存足够

库存不足

计算金额

产品分类

购买商品

提醒补货

根据商品

相关信息

图3 产品流程关系图

1. 物流

当消费者购买了指定商品，根据商品对应的物流规则以及用户填写的订单信息中的收货地址，消费者支付指定的邮费，物流公司进行商品的运送；同时，当消费者对已购买商品不满意时，物流根据消费者购买该商品填写的订单信息计算出对应的邮费，消费者支付对应的邮费即可将商品进行退换货。

退换货成功

支付邮费

评价商品

支付邮费

申请退换货

确认收货

运送商品

接受订单

图4 物流流程关系图

1.1.2用户对系统的要求

（1）信息要求

概括起来，可以将京东电商系统的功能根据用户的不同分为三个部分，分别针对买家、卖家和系统管理员，各用户对系统的信息要求也有不同，可分为以下几个方面：

1. 买家会员信息

当普通游客进行查看商品后，对于想要购买的商品，需要首先注册成为会员成为买家才能购买相应的商品，注册成为会员需要输入会员的相关信息，包括用户名、密码、手机号码，注册成功，还需要完成会员安全验证，包括实名认证、收货地址**、邮箱关联、账号关联**等信息，其中实名认证包括姓名和身份证，只有实名认证了购买商品才能够支付；收货地址包括收货人、手机号码、所在地区（省、市、县/区、乡/街道）、详细地址、标签、是否为默认地址。邮箱关联和账号关联即对应的邮箱和qq账号进行关联。

成为京东会员之后，会员可以关注喜欢的商品，收藏喜欢的店铺，查看店铺某一商品的评价信息，店铺的评价信息；当会员浏览相应商品时，会记录下会员的浏览记录。

1. 卖家店铺信息

有两种称为卖家的方式，一种是会员申请成为卖家，一种是公司申请成为卖家；要成为卖家，首先了解招商信息，接着进行入驻准备，准备相关的资质材料，开通京东钱包。然后进行入驻申请，注册京东账号，填写相关信息，提交申请，最后只需要提交申请之后等待官方审核就行了。当成为卖家之后，卖家的店铺信息主要包括店铺名称（可供买家直接搜索店铺名称购买商品，也可以让系统管理员根据店铺名称进行管理）、掌柜签名（表达自己店铺的个性化，吸引买家）、主营宝贝（指的是卖家店铺的主要卖的宝贝的类型，也是区分其他店铺的重要方式）、店铺动态（指的是卖家店铺最近的促销信息，折扣信息以及店铺最近的上新动态）、店铺成立时间（时间越长往往更容易让买家放心）、店铺评分和关注人数（买家可能会重视该店铺的评分高低和人气高低）等等。

1. 卖家商品信息

进入卖家店铺之后，可以查看卖家销售的商品，点击商品就会有相关的商品信息。商品信息里面主要包括：商品价格（往往放在最显眼的位置）、商品视频或图片介绍（翻动可查看多张图片）、商品简介（该商品的品牌、型号等）、商品的优惠信息（该店铺对该商品的优惠信息，折扣，满减等）、商品规格（点击可选择相应的规格，可以加入购物车或收藏或直接下单购买）、商品包邮和运费险以及七天无理由、商品的发货地以及买家的收货地址信息等。

1. 订单物流信息

订单中信息应该包括积分情况（购买商品可能获得积分）、交易快照（作为交易的证据）、订单编号（可以通过订单编号来查看相应的物流信息）、物流动态、创建时间（该笔交易的创建的时间记录）、付款时间（买家付款的时间记录）、发货时间（发货时间的记录）、成交时间（该笔交易成交的时间记录）、投诉商家、交易号（交易的订单编号）等等。下单之后，该商品就开始跟踪物流情况了。包括已下单、已发货、等待揽收、运输中、派件中等等物流情况，遇到特殊情况可以与卖家或者管理员进行交流来获取最新的物流动态。

1. 广告活动信息

好的广告，不论是对于平台的宣传还是对于平台内的商品的宣传都能对整个平台有一个好的影响，宣传商品也是对于平台的一种变相宣传，起到的效果是差不多的。广告的主要信息就包括：图片介绍、产品介绍、优势特点介绍、联系方式介绍、活动信息等等。

同时对于一些大型的活动，如著名的京东618购物节，需要在商城界面推送相关的活动信息，同时商家店铺也可以推送相关的活动，活动信息主要包括：活动时间、活动规则、活动优惠信息、优惠券获取（优惠券使用范围、优惠券使用金额、优惠券使用时间）等相关信息。

（2）处理要求

系统应当完成以下的信息处理：

a．存储买家信息、卖家信息、管理员信息、各种订单和证明的信息来供相关人员查询；

b．对买家信息进行及时的更新和统计；

c．对卖家信息进行及时更新和统计；

d．各种订单、物流信息、交易记录、退换货记录、运费险记录都要进行及时的跟踪以及更新。如买家下单之后的物流情况，要退换货时的运费险情况以及交易成功之后的评价功能都要处理；

e．对各个卖家所售商品的库存情况进行及时的更新和统计以及相关的自动处理，系统根据库存量、入库量、出库量、自动更新的即时的库存量完成更新，当库存少到一定的情况或者不足时，系统要提示相关人员进行管理，来做出相应的处理；

对原始数据和统计数据进行相关分析，如店铺销售量、销售金额、会员购买量、购买金额、商城销售量、商城销售金额、店铺评价等级等

f．根据相应的数据分析，进行排名，并在商城销售排行模块进行展示，同时商家店铺也可以有本店铺的销售量以及销售金额还有好评率等显示，会员也可以查看店铺的相关统计信息，会员还可以查看自己当月的花费金额等。

（3）安全性与完整性要求

安全性要求：

a．系统应设置访问用户的标识即用户名或手机号码验证等来鉴别是否是合法用户，并要求用户设置相关密码，保证用户身份不被盗用；

b．系统应对不通的数据设置不同的访问级别，限制访问用户可查询和处理数据的类别和内容，如会员无法查看商家月销售额，会员无法处理商家广告活动信息等；

c．系统应对不同用户设置不同的权限，区别不同的用户，如区分会员（只能查询相关商品、店铺的基本信息，同时可以购买相关商品，支付、查看物流订单、退货等权限），商家（查询本店铺的详细信息，对本店铺进行增删查改的权限，同时还拥有会员的相关权限）、商城管理员（可查询平台所有权限的信息，类似于超级管理员，可赋予其他用户权限，修改、审核会员、商家等权限）

完整性要求：

a．各种信息记录的完整性，信息记录内容不能为空；

b．各种数据间的相互的联系的正确性；

c．相同数据在不同记录中的一致性；

d．即要满足数据库设计的三大完整性要求（实体完整性、参照完整性、用户自定义完整性）

1.1.3确定系统边界

经过前面的需求调查和初步分析，确定由计算机完成的工作是对数据进行各种管理和处理，具体工作内容将在后续部分进行详细阐述， 由手工完成的工作主要有对原始数据的录入；不能由计算机生成的，各种数据的更新，包括数据变化后的修改，数据的增加，失效数据或无用数据的删除等；以及系统的日常维护。

1.2系统功能设计

根据前文得到的用户需求，将医院信息系统按照所需完成的功能分为如下几个子系统模块：

会员管理子系统

商家管理子系统

订单管理子系统

商城管理子系统

1.2.1会员管理子系统

（1）会员注册

游客只能浏览商品，当需要将商品添加到购物车并进行购买时，需要进行会员注册。

（2）会员登陆

当游客注册了会员后，需要根据注册时的会员账号及密码即可进行登录；

会员登陆了，还可以进行一些信息管理，如填写收货地址、填写账号关联、邮箱关联等信息。

（3）查询及购买商品

会员登录后，即可进行商品查询，当会员遇到心仪的商品，可以将商品添加到购物车，当会员选择好需要购买的商品后，选择购买，即可进入支付页面，支付相应的金额即可购买商品。

（4）收藏管理

会员对于心仪的商品或店铺可以进行收藏。

（5）订单评价及售后

会员购买商品后，会收到相应的订单信息，可以查看订单详情，如订单号、购买数量、金额、物流状态等；

当会员收到商品后，如果对商品满意，可以选择确认收货，确认收货后，会员可以对商品进行评价；

如果对商品不满意，可以跟商家联系，选择退货换货。

（6）优惠券管理

当商城进行活动时，或者店铺进行活动时，会员可以通过参加活动获取一定的优惠券，根据优惠券的使用期限、使用范围等信息，会员可以在购买商品的时候进行使用，使用后，优惠券即作废。

（7）购物车管理

当会员看到心仪的商品，可以将商品加入购物车后，然后统一进行购买。

1.2.2商家管理子系统

（1）商家申请或入驻

当会员想要成为商家后，需要提交相关信息，等待商城管理员审核通过后即可成为商家，成为商家后，即可拥有自己的店铺，可以对店铺进行管理；

同时一些企业收到商城的吸引也可以申请直接成为商家，当满足一定条件后既可成为商家，拥有自己的店铺。

（2）店铺管理

当商家成功注册后，即可拥有自己的店铺，商家可以对店铺进行管理，首先需要给店铺一个名字，其次可以完善店铺信息，如店铺联系人信息、公司信息、营业执照、财务信息、店铺类型等。

（3）商品管理

同时商家还可以对店铺进行管理，如添加商品，添加商品包括添加相应的信息（大小、尺寸、颜色、品牌等），价格、库存等；

当商家不再需要销售某商品时或者库存不足时，商家可以下架商品或者给商品添加新的库存；

同时商家还可以编辑商品，如修改商品的大小、尺寸、颜色、品牌、价格、库存等信息。

（4）订单管理

当会员购买该店铺的商品后，商家即需要联系物流公司进行商品运送，将商品交给物流公司进行运送后，订单状态需要实时更新；

会员收到商品后，如果满意可以进行评价，商家可以管理评价，如在一定情况下可以删除评价，修改评价等；

如果会员不满意，需要进行退换货，商家则需要根据实际情况是否允许会员进行退换货，确定退换货的运费。

（5）活动管理

当商家为了提高销量或者迎合商城的活动时，商家可以推出一些活动，如抢购、满减、优惠券等相关活动。

（6）店铺客服

如果商家足够大时，店铺还可以添加客服，会员购买商品时或购买商品后有疑问时，可以咨询店铺客服。

1.2.3订单管理子系统

（1）订单管理与生成

会员可以编辑订单、取消订单、添加商品、移除商品等，当会员收到商品对商品不满意时，可以根据订单信息选择退换货商品。

当会员选择购物车的商品时，会根据会员购买的店铺的不同来生成不同的订单。

（2）支付方式

订单提交时，对应着不同的支付方式，可以使用手工支付订单或者使用线上支付订单，线上支付订单也有不同的支付方式。

（3）订单状态变化

当将订单交给物流公司，物流公司对订单状态进行实时更新，订单状态分为待付款->待发货->待收货->待评价->已完成->退换货。

（4）订单运费

一个订单对应着某一物流公司，物流公司根据会员填写的地址计算出对应的邮费，该邮费可以由商家承担也可以由会员承担，具体根据商家购买商品的具体规则。

1.2.4商城管理子系统

（1）商城内容管理

包括商品的分类、店铺分类、广告管理、活动管理，其中广告管理又可分为添加、删除、修改广告，活动包括活动内容，活动时限等。

（2）数据报表统计

统计新增用户数、新增订单、新增商户、新发布商品；

统计店铺销量，根据店铺销量进行排名；

统计商品销量，根据商品销量进行排名；

统计会员相关信息，如诚信度、月购买量等，也可以进行排名。

（3）会员管理

管理会员违规信息，如随意发表不符规则的评论、恶意评价商品等；

管理会员申请成为商家的信息；

管理会员举报信息，当会员举报相关店铺，需要进行介入管理。

（4）商家店铺管理

管理店铺违规信息，如不合法刷评论、虚假商品，用户举报店铺等；

管理店铺商品，如果商品违规，对商品进行处理，同时记录该店铺的违规记录。

（5）订单管理

当会员与商家之间的订单有误或者出错时，需要对订单进行管理，包括更改订单状态、删除订单等。

（6）提供商城相关信息的查询

当会员需要购买商品时，提供会员对商品进行查询，包括精确查询、模糊查询等，同时还提供会员对店铺的查询；

对会员提供当月或者当日店铺销量排行、商品销量排行、商品好评排行等信息；

当店铺进行销售时，对店铺提供销量统计查询等，给商家提供一个好的销售参考意见。

通过以上分析，得到了京东电商系统的基本功能要求和系统模块划分，现对会员管理子系统、商家管理子系统、订单管理子系统、商城管理子系统进行具体的数据库设计。

1.3数据流图

需求分析阶段应该形成的成果包括数据流图和数据字典，该节内容给出以上会员管理子系统、商家管理子系统、订单管理子系统、商城管理子系统的数据流图

**会员**

申请查询商品

查询

结果

物流

注册/登录

优惠券管理

填写

会员

相关

信息

返回登陆结果

查询

结果

申请查询物流

使用/查看优惠券

优惠券被使用

收藏商品/店铺

支付金额

生成订单

申请收藏

收藏

结果

订单及评价

查看订单信息，收货后可评价或退换货

返回

结果

购物车

添加商品

显示结果

图5 会员管理子系统数据流图

**商家**

影响店铺销量

推出/编辑活动

店铺管理

申请入驻

商品管理

提交相关资料

返回审核结果

返回结果

完善店铺信息

返回管理结果

编辑增删商品

管理客服信息

客服服务结果

订单管理

查看订单信息，审核会员的退换货需求并确定运费

返回

结果

客服管理

活动管理

图6 商家管理子系统数据流图

信息反馈

物流公司接货

库存管理

商家

运输

买家

确认收货

订单评价

退换货

运输

接收订单

有效订单

无效订单

取消订单

确认订单

信息反馈

图7 订单管理子系统数据流图

**商城管理员**

商城活动管理

商城广告管理

添加删除活动

返回结果

审核结果

添加删除广告

返回结果

商家、会员、管理员权限划分

划分结果

审核会员、商家、管理员注册、违规等行为

统计结果

进行统计

图8 商城管理子系统数据流图

1.4数据字典

需求分析阶段还应该形成数据字典。本节内容给出京东电商系统中各实体及联系的详细数据字典

表1：会员收藏数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 收藏id | id | String | 10 |  |
| 商品或店铺id | favorite\_id | String | 10 |  |
| 创建时间 | add\_time | Date | 10 |  |
| 收藏类型 | id\_type | Char | 1 | 1：商品 2：店铺 |
| 状态 | status | Bool | 1 | 0：无效 1：有效 |
| 会员id | user\_id | String | 10 |  |

表2：优惠券信息数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 优惠券编号 | coupon\_id | String | 10 | 主码 |
| 优惠券开始时间 | coupon\_start\_time | Date&Time | 20 |  |
| 优惠券结束时间 | coupon\_end\_time | Date&Time | 20 |  |
| 优惠券金额 | coupon\_money | Real | 10 |  |
| 优惠券类型 | coupon\_type | Int | 2 | 1：满减 2：折扣 3：现金. |
| 优惠券使用范围 | coupon\_use\_range | Int | 2 | 1:平台通用 2:店铺通用 3:特定商品 |
| 优惠券使用条件 | coupon\_condition | Real | 10 | 某一金额 |
| 优惠券使用范围id | coupon\_use\_id | String | 10 | 根据类型确定ID，如果是平台通用ID为0，否则为对应店铺或商品ID |

表3：会员收获地址数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 地址id | add\_id | String | 10 |  |
| 会员id | user\_id | String | 10 |  |
| 收货人姓名 | receiver\_name | String | 20 |  |
| 手机号码 | receiver\_phone | String | 15 |  |
| 省 | province | String | 5 |  |
| 市 | city | String | 5 |  |
| 区/县 | county | String | 5 |  |
| 街道/镇 | town | String | 5 |  |
| 详细地址 | detail\_add | String | 50 |  |
| 是否默认 | is\_default | Bool | 1 |  |

表4：会员基本信息数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 会员id | user\_id | String | 10 |  |
| 用户名 | user\_name | String | 20 |  |
| 密码 | user\_pwd | String | 20 |  |
| 旧密码 | user\_former\_pwd | String | 20 |  |
| 注册时间 | register\_time | Date | 10 |  |
| 会员类型 | user\_type | Char | 1 | 1:会员 2:商家 3:管理员 |
| 是否绑定邮箱 | is\_bind\_email | Bool | 1 |  |
| 是否绑定qq | is\_bind\_qq | Bool | 1 |  |
| 手机号码 | user\_phone | String | 15 |  |
| 审核状态 | user\_status | Bool | 1 |  |

表5：会员详细信息数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 会员id | user\_id | String | 10 |  |
| 邮箱 | user\_email | String | 20 |  |
| qq | user\_qq | String | 10 |  |
| 支付密码 | user\_pay\_pwd | String | 20 |  |
| 真实姓名 | user\_real\_name | String | 10 |  |
| 性别 | user\_sex | Char | 1 |  |
| 证件类型 | user\_certificate\_type | Char | 1 | 1:身份证2:军官证3:护照 |
| 证件号码 | user\_certificate | String | 20 |  |
| 头像 | user\_img | Text | 15 |  |
| 出生日期 | user\_birth | Date | 10 |  |
| 会员积分 | user\_point | Int | 10 |  |
| 会员级别 | user\_level | Short Integer |  | 1:普通会员2:青铜3:白银4:黄金5:钻石 |

表6：会员级别数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 会员级别id | level\_id | String | 10 |  |
| 会员级别名称 | level\_name | String | 10 |  |
| 该级别最低积分 | min\_point | Int | 10 |  |
| 该级别最高积分 | max\_point | Int | 10 |  |

表7：商城广告数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 广告id | adver\_id | String | 10 |  |
| 广告标题 | adver\_title | String | 20 |  |
| 广告链接 | adver\_url | String | 100 |  |
| 是否可关闭 | is\_close | Bool | 1 |  |
| 广告开始时间 | adver\_start\_time | Date | 10 |  |
| 广告结束时间 | adver\_end\_time | Date | 10 |  |
| 广告类型 | adver\_type | Int | 2 | 1:商城广告 2:店铺广告 |
| 店铺id | store\_id | String | 10 | 当广告类型为2时才有值，否则为null |

表8：商城促销活动数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 活动id | activity\_id | String | 10 |  |
| 活动名称 | activity\_title | String | 20 |  |
| 开始时间 | activity\_start\_time | Date | 10 |  |
| 结束时间 | activity\_end\_time | Date | 10 |  |
| 活动范围 | activity\_range | Int | 2 | 1:所有商品 2:部分 |
| 活动类型 | activity\_type | Int | 2 | 1:满减 2:直降 |
| 店铺id | store\_id | String | 10 | 当活动范围为1时该字段为0，表示所有店铺都有活动 |

表9：商城满减活动数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 满减id | activity\_id | String | 10 |  |
| 优惠金额 | discount\_price | Real | 10 |  |
| 满足金额 | meet\_price | Real | 10 |  |
| 商品id | goods\_id | String | 10 |  |

表10：商城直降活动数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 直降id | activity\_id | String | 10 |  |
| 折扣百分比 | discount\_percent | Real | 10 |  |
| 最大订货量 | max\_num | Int | 5 |  |
| 最小订货量 | min\_num | Int | 5 |  |
| 原销售价 | former\_price | Real | 10 |  |
| 促销价 | promotin\_price | Real | 10 |  |
| 商品id | goods\_id | String | 10 |  |

表11：商品SPU数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| spu\_id | spu\_id | String | 10 |  |
| 商品名称 | goods\_name | String | 10 |  |
| 价格 | goods\_price | Real | 10 |  |
| 详细信息 | goods\_detail\_info | Text | 512 |  |
| 上架时间 | goods\_start\_time | Date&Time | 20 |  |
| 下架时间 | goods\_end\_time | Date&Time | 20 |  |
| 商品简介 | goods\_bried\_info | Varchar | 256 |  |
| 商品评分 | goods\_point | Real | 10 |  |
| 可售状态 | goods­­\_statue | Bool | 1 |  |

表12：商品类型数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 类型id | type\_id | String | 10 |  |
| 类型名称 | type\_name | String | 30 |  |

表13：商品规格数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 规格id | spec\_id | String | 10 |  |
| 规格名称 | spec\_name | String | 30 |  |

表14：商品规格值数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 规格id | spec\_value\_id | String | 10 |  |
| 规格值 | spec\_value | String | 20 |  |

表15：品牌数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 品牌id | brand\_id | String | 10 |  |
| 品牌描述 | brand\_info | Varchar | 512 |  |
| 品牌状态 | brand\_status | Bool | 1 |  |
| 创建时间 | brand\_create\_time | Date&Time | 20 |  |
| 修改时间 | brand\_modified\_time | Date&Time | 20 |  |

表16：类目数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 类目id | category\_id | String | 10 |  |
| 父类目 | category\_parent | String | 10 |  |
| 类目名称 | category\_name | String | 10 |  |
| 类目状态 | category\_status | Bool | 1 |  |

表17：SKU数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 商品id | goods\_id | String | 10 |  |
| 价格 | goods\_price | Real | 10 |  |
| 库存量 | goods\_num | Int | 10 |  |
| 规格 | goods\_spec\_value | String | 20 |  |

表18：店铺数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 店铺id | store\_id | String | 10 |  |
| 店铺名称 | store\_name | String | 10 |  |
| 店铺成立时间 | store\_create\_time | Date&Time | 20 |  |
| 店铺评分 | store\_point | Int | 10 |  |
| 关注人数 | store\_follower | Int | 10 |  |
| 店铺联系方式 | store\_phone | String | 15 |  |
| 店铺类型 | store\_type | String | 10 |  |
| 店铺状态 | store\_status | Bool | 1 |  |

表19：会员评价商品数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 评价id | comment\_id | String | 10 |  |
| 订单id | order\_id | String | 10 |  |
| spu\_id | spu\_id | String | 10 |  |
| 商品id | goods\_id | String | 10 |  |
| 评价内容 | comment | Varchar | 512 |  |
| 用户id | user\_id | String | 10 |  |
| 评分 | comment\_point | Int | 2 | 评分为1-5分 |
| 评价时间 | comment\_date | Date | 10 |  |
| 审核状态 | comment\_status | Bool | 1 |  |

表20：客服数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 客服id | server\_id | String | 10 |  |
| 客服昵称 | server\_name | String | 10 |  |
| 客服是否在线 | server\_status | Bool | 1 |  |
| 客服信息 | server\_info | Varchar | 512 |  |

表21：订单主表数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 订单id | order\_id | String | 10 |  |
| 会员id | user\_id | String | 10 |  |
| 店铺id | store\_id | String | 10 |  |
| 支付时间 | order\_start\_time | Date&Time | 20 |  |
| 支付方式 | pay\_type | Char | 1 | 1：微信支付 2：银行卡支付 3：本地钱包支付 |
| 快递单号 | delivery\_id | String | 10 |  |
| 订单总额 | order\_money | Real | 10 |  |
| 优惠金额 | district\_money | Real | 10 |  |
| 订单积分 | order\_point | Int | 10 |  |
| 运费险 | dilivery\_assurance | Bool | 1 |  |
| 订单状态 | order\_status | Char | 1 | 1：待付款 2：待发货 3：待评价 4：退换/售后 |
| 发货时间 | order\_start\_time | Date&Time | 20 |  |
| 收货时间 | order\_end\_time | Date&Time | 20 |  |

表22：订单商品数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 订单编号 | order\_id | String | 10 |  |
| spu\_id | spu\_id | String | 10 |  |
| 商品id | goods\_id | String | 10 |  |
| 店铺id | store\_id | String | 10 |  |
| 商品数量 | goods\_num | Int | 10 |  |
| 商品单价 | goods\_price | Real | 10 |  |

表23：物流信息数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 快递单号 | delivery\_id | String | 10 |  |
| 物流公司id | company\_id | String | 10 |  |
| 发货时间 | delivery\_start\_time | Date&Time | 20 |  |
| 收货时间 | delivery\_end\_time | Date&Time | 20 |  |
| 收货地址id | add\_id | String | 10 |  |
| 物流状态 | delivery­­\_status | Real | 10 | 1：待发货 2：已发货 3：运输中 4：派件中 5：已收货 |
| 配送员姓名 | distributor\_name | String | 10 |  |
| 配送员联系方式 | distributor\_phone | String | 15 |  |

表24：物流公司数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 物流公司id | company\_id | String | 10 |  |
| 物流公司名称 | company\_name | String | 10 |  |
| 物流公司联系人 | company\_contact\_name | String | 10 |  |
| 物流公司电话 | company\_contact\_phone | String | 10 |  |
| 运费金额 | company\_monye | Real | 10 |  |
| 配送员姓名 | distributor\_name | String | 10 |  |
| 配送员联系方式 | distributor\_phone | String | 15 |  |

表25：退货数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 退货订单编号 | return\_id | String | 10 |  |
| 申请时间 | return\_time | Date&Time | 20 |  |
| 退货原因 | return\_reason | Varchar | 512 |  |
| 退货运费 | return\_money | Real | 10 |  |
| 物流公司id | company\_id | company\_id | 10 |  |
| 买家地址id | user\_add\_id | String | 10 |  |
| 商家地址 | store\_add\_id | String | 10 |  |
| 审核状态 | return\_status | Char | 1 |  |

表26：购物车数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 购物车id | cart\_id | String | 10 |  |
| 用户id | user\_id | String | 10 |  |
| 商品id | goods\_id | String | 10 |  |
| 购买数量 | goods\_num | Int | 10 |  |
| 商品价格 | good\_price | Real | 10 |  |
| 商品状态 | good\_status | Bool | 1 | 下单之后商品就不在购物车了 |
| 商品是否选中 | goods\_selected | Bool | 1 | 表示是否选中购买该商品 |

2概念结构设计

2.1会员管理子系统

根据需求分析中画出的会员管理子系统的第一层数据流图，可以看出在会员管理子系统中一切活动都是以会员为核心，各种行为都是由会员主动去完成的，如注册、登录、搜索商品、加入购物车、购买商品、支付、查看订单信息、评价商品、退换货等等。在整个数据流图中会员处于核心地位，为了更加完整的得到会员管理子系统的E-R图，根据实际需求继续对会员管理子系统进行划分，可以划分为4个子系统，从这4个子系统设计出4个概念模式，为下一步的概念结构设计打好基础，这4个子系统是根据时间先后顺序进行划分，该4个子系统分别为：会员注册登录，会员选择商品，会员购买商品，会员购后行为子系统。

会员管理子系统

会员登陆注册

会员购买商品

会员选择商品

会员购后行为

图9 会员管理子系统概念结构图

以会员管理子系统的第一层数据流图为基础，接下来对于各个稍微复杂的处理过程进行细化，画出其具体数据流图，从而抽象出E-R图，为进一步的逻辑设计打下基础。以下为4个处理过程的具体数据流图分析。

2.1.1数据流图分析

（1）会员登录注册模块

本子系统是用于刚刚进入该系统的用户而用，对于没有进行注册的用户，只能查看本系统展示的初始界面，如需购买商品等需要进行登录，如果用户不拥有该系统的账号，则需要进行注册，注册需要填写会员相关信息，登录同样也需要填写相关信息，如账号、密码，登录成功或失败都返回商城主页面，同时登录成功即可购买相关商品，但支付需要身份证等实名认证信息。

选择商品子系统

成功

商城主页面

失败

管理员审核

注册

**会员**

登录

成功

失败

完善个人信息

图9 登陆注册模块数据流图

（2）选择商品模块

会员在注册登录商城后，完善个人相关信息后，即可选择购买相关商品，购买商品之前，会员可以选择将商品加入购物车，同时遇到心仪的商品或者店铺，会员可以收藏商品或者店铺。

**会员**

查询商品

收藏商品

收藏店铺

加入购物车

选择商品子系统

图10 商品选择数据流图

（3）购买商品模块

会员在选择了足够的商品后，既可以进行购买，购买可以通过选中购物车的部分商品进行购买，同时也可以选中直接在店铺的某个商品进行购买，注意购买之后生成的订单是根据店铺划分的，一个店铺的商品生成一个订单，购买商品还可以使用优惠券，通过商城活动获得的优惠券可以减少会员的实付金额，同时会员购买商品实付金额的10%作为会员积分，会员积分达到一定数量会员等级会进行升级，不同的会员等级所拥有的权限不同，购买商品的限制以及优惠也不同。

**会员**

填写订单信息

是否使用优惠券

支付金额

提交订单

购后行为子系统

计算金额

选定商品

图11 商品购买数据流图

（4）购后行为模块

会员提交订单之后，只需要等待物流将商品送达目的地即可，此时，会员可以查看订单信息，物流信息，当商品送达目的地，会员可以选择拿取，同时确认收货，收货之后可以进行评价；会员也可以对商品表示不满意，与商家沟通后进行退货，退货的运费根据实际规则计算，退货也是使用物流将商品送还给商家。

**会员**

商品到达目的地

查看物流信息

查看订单信息

退货物流

评价订单

退货

确认收货

图12 购后行为数据流图

2.1.2 E-R模型设计

对应各个功能模块的数据流图，设计得到的E-R模型如图13——图16所示。

（1）会员登录注册模块

会员等级

对应2

**N**

**1**

会员基本信息

会员详细信息

会员收货地址

拥有1

对应1

**N**

**1**

**1**

**1**

图13 会员登录注册模块E-R图

（2）商品选择模块

会员基本信息

拥有2

拥有3

3

会员收藏夹

购物车

**N**

**N**

**1**

**1**

图14 商品选择模块E-R图

（3）商品购买模块

会员基本信息

拥有4

拥有5

优惠券

订单主信息

**N**

**N**

**1**

物流信息

对应3

**1**

**1**

**M**

图15 商品购买模块E-R图

（4）购后行为模块

会员基本信息

退货

评价

评价商品

退货信息

**N**

**N**

**1**

**1**

图16 商品购买模块E-R图

各子功能模块的分E-R图设计完成后，结合会员管理子系统的第一层数据流图设计出总体E-R图如图17所示。由于篇幅的问题，不再画出各个实体与其属性的E-R分图，各个实体及其属性可以参看下面列出的说明。

**M**

**1**

**N**

**N**

**N**

**N**

**N**

**1**

**1**

**N**

**N**

会员基本信息

拥有4

会员等级

会员详细信息

对应1

等级

拥有1

优惠券

**1**

评价

拥有2

拥有3

退货

**1**

**1**

**1**

会员收货地址

拥有5

**1**

评价商品

退货信息

会员收藏夹

购物车

商家子系统接口

物流信息

对应2

订单主信息

**1**

**1**

**N**

拥有6

**1**

**1**

**N**

图17 会员管理子系统总体E-R图

各实体的属性如下：

会员基本信息（会员id，用户名，密码，旧密码，注册时间，会员类型，是否绑定邮箱，是否绑定qq，手机号码，审核状态）

会员收货地址（地址id，收货人姓名，手机号码，省，市，区/县，街道/镇，详细地址，是否默认）

会员详细信息（邮箱，qq，支付密码，真实姓名，性别，证件类型，证件号码，出生日期，会员积分）

会员等级（会员级别id，会员级别名称，该级别最低积分，该级别最高积分）

购物车（购物车id，购买数量，商品价格，商品状态）

会员收藏夹（收藏id，商品或店铺id，创建时间，收藏类型，状态）

订单主信息（订单id，支付时间，支付方式，快递单号，订单总额，优惠金额，订单积分，运费险，订单状态，发货时间，收货时间）

物流信息（快递单号，发货时间，收货时间，物流状态）

优惠券（优惠券编号，优惠券开始时间，优惠券结束时间，优惠券金额，优惠券类型，优惠券使用范围，优惠券使用条件，优惠券使用范围id）

退货信息（退货订单id，申请时间，退货原因，退货运费，，审核状态）

评价商品（评价id，评价内容，评分，评价时间，审核状态）

各联系的属性说明如下：

对应1、对应2都是1：1联系，只需将一方主码加入另一方作为外码即可，不需要属性。

拥有2、拥有3、拥有4、拥有5、拥有6、等级、退货、评价都是一对多的联系，只需在多的那一方加入一的那一方的主码作为外码即可，联系也不需要多余的属性。

拥有1为多对多的联系，需要生成一个新的关系，该联系需要含有两个属性，分别为领取时间、数量。

解释一：物流信息是指从卖家的订单出发后的物流信息，其物流状态与订单的订单状态是不同的。

解释二：关于会员收藏夹，为了使得用户收藏的店铺和商品能够统一，为会员收藏夹新增加了一个属性收藏类型，该属性用于表明收藏的是商品还是店铺，因此对于该关系而言，收藏类型和商品或店铺id两个属性可以组成一个候选码。

解释三：对于会员积分而言，该项作为会员等级的升级指标，当达到一定的会员积分，会员可以升级，不同等级的会员的权限以及优惠程度不同，同时也为了后期扩展新功能，建立良好的电商社区等新功能提供了一定的基础，同时会员积分的获取方式为：用户购买订单的实付金额的10%。

解释四：订单信息中的优惠金额有两种优惠方式，一种是使用优惠券，当达到了优惠券的使用条件即可使用优惠券从而有所优惠；第二种就是利用商城活动，商城活动又分为满减活动和直降活动，可以根据实际活动的相关规则从而得到优惠，用户实付金额为商品总额减去优惠金额。

2.2商家管理子系统

根据需求分析中画出的商家管理子系统的数据流图，可以看出在商家管理子系统中一切活动都是以商家为核心，而各种处理也是由商家主动去完成，如完善店铺信息、管理店铺、管理订单、设置退换货运费、添加/下架商品等。商家管理子系统能够让商家更方便地管理自己的商业信息，包括对店铺的处理及自己商品的处理，能对提高自己销售量起到帮助。在整个数据流图中商家处于核心地位。

商家管理子系统

申请入驻模块

订单管理模块

商品管理模块

图18 商家管理子系统概念结构图

以商家管理子系统数据流图为基础，接下来对于各个稍微复杂的处理过程进行细化，画出其具体数据流图，从而抽象出E-R图，为进一步的逻辑设计打下基础。而在本子系统的数据流图中，客服管理与店铺管理处理的数据比较简单，可以很简单的找出其关系表，活动管理在商城子系统中已有具体分析，这三个不再进行细化分析。另外3个处理过程的具体数据流图分析如下。

2.2.1数据流图分析

（1）申请入驻模块

本子系统是用于商家入驻之用，需要成为商家的基础是已经通过注册成为了本系统的会员，拥有自己的账号，在此基础上，提供相关资料供管理员审核，只有在相关资料符合审核条件通过后，才能正式入驻成为商家。而对于相关资质或有不正当行为的商家，管理员有权限将已入驻的商家状态设置为无效，不能继续售卖商品。在审核通过后，系统会将审核结果反馈给商家。

会员

完善相关资料

审核不通过

审核通过

开通店铺

提交相关资料

成为商家

退回申请

申请入驻成为商家

店铺信息管理

设置店铺客服

图19 申请入驻模块数据流图

（2）商品管理模块

本子系统是用于商家管理售卖商品的作用，以商家为主体对数据进行处理，商家可自由选择上架、下架商品或者编辑已经上架的商品信息，例如大小、尺寸、颜色、品牌、价格、库存等。有时会出现商品库存量与现实不符，若继续售卖会引起库存量无法满足订单销售量，影响购买会员与店铺的信誉，此时商品管理模块可以提供修改信息的功能来解决类似的问题。

下架商品

商家

选择商品

修改商品信息

管理商品

编辑商品

上架商品

填写商品信息

图20 商品管理模块数据流图

（3）订单管理模块

本子系统是用于商家处理订单，当会员购买该店铺的商品后，商家即需要联系物流公司进行商品运送，将商品交给物流公司进行运送后，订单状态需要实时更新；会员收到商品后，如果满意可以进行评价，商家可以管理评价，如在一定情况下可以删除评价，修改评价等；

如果会员不满意，需要进行退换货，商家此时则需要根据实际情况是否允许会员进行退换货，确定退换货的运费。

商家确定运费

满意

不满意

退换货

会员评价

是否满意

会员购买商品

商家

反馈

联系

运送商品

物流公司

会员收货

图21 订单管理模块数据流图

2.2.2 E-R模型设计

对应各个功能模块的数据流图，设计得到的E-R模型如下所示。

（1）商家入驻模块

**1**

**1**

会员基本信息

对应

店铺

**1**

拥有

**N**

客服

图22 商品入驻模块E-R图

（2）商品管理模块

**N**

**1**

**1**

**N**

**1**

店铺

类目表

SKU

包含9

包含8

包含6

包含5

商品SPU

**N**

品牌

包含4

商品类型

**1**

包含1

**1**

包含7

**1**

商品规格

包含3

商品规格值

**1**

包含2

**N**

**N**

**N**

**N**

**1**

**1**

**1**

**N**

图23 商品管理模块E-R图

（3）订单管理模块

店铺

包含10

订单主表

**N**

**1**

图24 订单管理模块E-R图

以上三个模块的E-R图设计完成后，结合商家管理子系统第一层数据流图设计总体的E-R图。由于篇幅问题，不在画出各个实体与其属性的E-R分图，各个实体及其属性可以参考下面列出的说明。

**1**

**1**

对应

**1**

**1**

**1**

店铺

**N**

商品SPU

**N**

包含6

包含8

包含9

SKU

包含4

品牌

包含5

**N**

**N**

**N**

商品类型

包含1

**1**

**1**

商品规格

包含2

包含3

商品规格值

包含7

类目表

**1**

**1**

**N**

**N**

包含10

订单主表

会员基本信息

客服

**1**

**N**

拥有

**1**

**1**

**1**

**N**

图25 商家管理子系统总E-R图

各实体的属性如下：

1.店铺（店铺id，商家姓名，电话，店铺成立时间，店铺评分，关注人数，店铺联系方式，店铺类型，店铺状态）

2.订单主表（订单编号，支付时间，支付方式，订单总额，优惠金额，订单状态，订单积分，运费险）

3.客服（客服编号，客服昵称，客服信息，客服在线状态）

4.类目（类目编号，父类目，类目名称，类目状态）

5.商品SPU：（SPU编号，商品名称，价格，详细信息，上架时间，下架时间，商品简介，商品评分，可售状态）

6.商品SKU（商品编号，价格，库存量，规格）

7.品牌（品牌编号，品牌描述，品牌状态，创建时间，修改时间）

8.会员评价（评价编号，订单编号，spu编号，商品编号，评价内容，用户编号，评价时间，审核状态）

9.商品类型（类型id，类型名称）

10.商品规格（规格id，规格名称）

11.商品规格值（规格id，规格值）

各联系的属性说明如下：

包含1、包含2、包含3、包含4、包含5、包含6、包含7、包含8、包含9、包含10、对应、拥有均为1对多联系，只需在转换为关系时按照概念模式与关系模式转换的一般规则转换即可，以上联系均不需要多余的属性。

解释一：为了保证京东数据库系统支持多种商品，多类商品，本次设计采用了SKU和SPU的基本设计理念，将商品类型和商品具体规格值等分离开，可以极大保证减少数据冗余度。

2.3订单管理子系统

订单管理子系统是用户在下订单时所需要的子系统。在需求部分已经总结得出了总的数据流图，在此将进一步细化，有利于设计出对应的E-R图。

经过详细的业务分析，得到订单管理子系统的具体业务流程如下所示：首先订单管理子系统接收订单，但此时的订单可能是有效订单也可能是无效订单，当用户某些信息填写错误导致不能直接下单的时候，就有可能出现无效订单，而无效订单用户就会进行取消订单操作。当订单是有效订单的时候，用户即可确认订单，商家接收到这个确认订单的消息后，就会进行相应的发货并分配相应的物流公司根据用户所在的地址进行商品运输，这个期间会存在一个信息反馈的过程，这是因为需要保证商品在运输的过程中商家和买家都能够查看到相关商品的运输过程。当商品到达买家手里的时候，买家可以选择直接确认收货，收货完成之后，买家还可以选择对商品进行订单评价，该笔订单交易就完成了，也可以选择进行退货或者是换货操作，再通过物流公司将商品运回相应的商家手里，也算完成交易。

运输

买家

确认收货

订单评价

退换货

运输

信息反馈

接收订单

物流公司接货

有效订单

无效订单

取消订单

库存管理

确认订单

商家

信息反馈

图26 订单管理子系统总数据流图

根据需求分析画出订单管理子系统的第一层数据流图，可以看出在订单管理子系统中，是以订单和物流两个模块为核心的，各种处理也是围绕这两个模块去完成的，所以接下来的数据流图分析也将围绕这两个模块展开详细的介绍。

2.3.1 数据流图分析

（1）订单模块

本模块是用于用户从选购商品一直到确认订单的所有操作进行分析。其中需要的信息很多，从选购商品开始就可能出现商品已下架和未下架两种情况，在未下架商品中，也可能存在商品的库存不足等问题，这就需要及时的能够提醒补货的功能，而接下来就是用户的信息填写，其中包含很多相关信息的填写，比如选择配送方式、选择收货地址、选择支付方式、是否使用优惠券等等，把信息填写完毕后订单的相关内容就会呈现给买家，买家确认相关信息正确无误后就可以提交订单了。订单模块的数据流图如下图所示：

信息填写

优惠券使用

选择配送方式

选择支付方式

选择收货地址

不使用

提交订单

商家接单

不使用

京东配送

商家指定

本地钱包

微信

银行卡

用户

未下架

选择商品

已下架

库存充足

库存不足

提醒补货

图27 订单子系统数据流图

（2）物流模块

本模块是用于用户确认订单之后，商家处理订单一直到买家收到货确认收货或者退货的所有操作进行分析。商家处理订单之后将货物交给相应的物流公司或者是商家指定的物流公司，接下来就是由物流公司负责将相关货物运输到买家手上与此同时在运输途中必须保持商家和买家都能掌握到最新的物流动态。在退货时，还要考虑到该商品是否包含运费险，如果有运费险，买家在退货时需先行垫付运费而后商家收到货后在进行赔付，若没有运费险，则退货运费需买家自行承担物流模块的数据流图如下图所示：

退货处理

含运费险

运费险

确认收货

不含运费险

结束交易

物流模块

商家自付

买家自付

商家收货

买家收货

物流模块

图28 物流子系统数据流图

* + 1. E-R模型设计

对应各个功能模块的数据流图，设计得到E-R模型如图所示。

（1）订单模块

订单主表

物流信息表

拥有

对应

**N**

**1**

**1**

**N**

会员基本信息

订单商品表

店铺

商品

包含2

包含3

包含1

**1**

**1**

**1**

**1**

**N**

**N**

图29 订单模块E-R图

（2）物流模块

会员基本信息

退货表

退货

**N**

**1**

**N**

物流信息表

物流公司表

包含4

**1**

包含5

1

**N**

拥有

物流配送员

**1**

**N**

图30 物流模块E-R图

订单模块和物流模块的E-R图设计完成后，结合订单管理子系统第一层数据流图设计总体的E-R图。由于篇幅问题，不在画出各个实体与其属性的E-R分图，各个实体及其属性可以参考下面列出的说明。

订单商品

商品

**N**

**1**

**M**

包含3

店铺

**N**

**1**

**1**

会员基本信息

对应

拥有

退货

退货表

订单主表

**1**

**1**

**N**

**N**

物流信息表

物流公司表

包含5

包含4

**1**

N

**1**

**1**

**N**

图31 订单管理子系统总E-R图

各实体属性如下：

会员基本信息（会员id，用户名，密码，旧密码，注册时间，会员类型，是否绑定邮箱，是否绑定qq，手机号码）

订单主表（订单id，支付时间，支付方式，订单总额，优惠金额，订单积分，运费险，订单状态）

物流信息表（快递单号，发货时间，收获时间，物流状态）

物流配送员（配送员id，配送员姓名，配送员联系方式）

物流公司表（物流公司id，物流公司名称，物流公司联系人，物流公司电话，运费金额）

商品（spu\_id，商品名称，价格，详细信息，上架时间，下架时间，商品简介，商品评分，可售状态）

退货表（退货订单编号，申请时间，退货原因，退货运费，审核状态）

店铺（店铺id，店铺名称，店铺成立时间，店铺评分，关注人数，店铺联系方式，店铺类型，店铺动态）

各联系的属性说明如下：

以上联系为一对多的联系，只需在转换为关系时按照概念模式与关系模式转换的一般规则转换即可，一对多联系均不需要多余的属性。

注意订单主表和商品SKU表有多对多联系，此联系需要额外属性，如下：

订单商品表（订单价格，订单购买数量）

解释一：对于会员基本信息而言，由于含有会员类型字段，因此该关系不仅可以用来表示会员，还可以表示商家、管理员，因此需要注意。

解释二：退货表同时也跟会员收货地址是多对一的关系，在本图中没有画出，因为在会员管理子系统已经描述，因此在此不再赘述。

解释三：注意退货表只与物流公司有联系，与物流信息表没有联系，即在本系统中，如果会员进行退货，无法查看退货信息物流，只需当商家收到商品后按照退货规则返还商品金额即可。

2.4商城管理子系统

该子系统的局部E-R模型如下图所示

**1**

会员基本信息

商城活动

商城广告

广告

活动

**1**

**N**

**N**

包括3

店铺

**1**

**N**

对应1

商品

店铺

满减活动

**1**

**N**

**N**

**M**

**M**

直降活动

s

**N**

图18 商城管理子系统总体E-R图

对应的实体及实体属性如下所示：

商城广告（广告id，广告标题，广告链接，是否可关闭，广告开始时间，广告结束时间，广告类型，店铺id）

商城活动（活动id，活动名称，开始时间，结束时间，活动范围，活动类型，店铺id）

商城满减活动（优惠金额，满足金额，商品id）

商城直降活动（折扣百分比，最大订货量，最小订货量，原销售价，促销价，商品id）

对应的联系及属性如下：

以上总体E-R图中一对多的关系均不需多余属性，只需在转化为ldm时在多的一方加入一的主码作为外码即可。

满减活动（优惠金额，满足金额）

直降活动（折扣百分比，最大订货量，最小订货量，原销售价，促销价）

解释一：注意以上所有行为均为商城管理员进行操作，即商城管理员管理商城广告和活动，同时对于本系统而言，商城管理员，商家，会员均由会员基本信息表确定，因为在会员基本信息表中加入了一个字段，作为区分商城管理员，商家，会员的手段，该字段名为会员类型。

3.逻辑结构设计

在概念结构设计的基础上，根据设计得到系统总体E-R图，按照概念模式与关系模式转换的一般规则，结合实际需求进行逻辑结构设计，E-R图中的实体、实体的属性和实体之间的联系转化为关系模式，最后生成的关系及关系表如下（同时附优化说明）

3.1关系模式设计及优化说明

3.1.1会员管理子系统

（1）．会员基本信息（会员id，会员级别id，店铺id，用户名，密码，旧密码，注册时间，会员类型，是否绑定邮箱，是否绑定qq，手机号码，注册审核状态）

说明：会员基本信息与会员级别、店铺是一对多联系，因此在该表加入会员级别id，店铺id。

（2）．会员详细信息（会员id，邮箱，qq，支付密码，真实姓名，性别，证件类型，证件号码，出生日期，会员积分）

说明：因为会员详细信息与会员基本信息属于一对一联系，而该关系没有主码，因此在该关系中加入会员id作为外码且为非空。

（3）．会员等级（会员级别id，会员级别名称，该级别最低积分，该级别最高积分）

（4）．会员收获地址（地址id，会员id，收货人姓名，收货人手机号码，省，市，区/县，街道/镇，详细地址，是否默认）

说明：该关系与会员属于一对多联系，因此在该关系加入会员id作为外码且非空。

（5）．购物车（购物车id，会员id，商品id，购物车商品状态，购物车购买数量，购物车商品价格）

说明：该关系与会员基本信息，商品SKU为一对多关系，因此加入他们的主码作为外码，且都为非空。

（6）．会员收藏夹（收藏id，会员id，收藏类型，状态，商品/店铺id，收藏夹创建时间）

说明：需要加入会员id作为外码，同时非空。

（7）．订单主信息（订单id，会员id，店铺id，支付时间，支付方式，订单总额，订单积分，运费险，订单状态，订单优惠金额）

说明：加入会员id，店铺id作为外码，同时非空。

（8）．物流信息（快递单号，订单id，物流公司id，地址id，发货时间，物流状态，收货时间）

说明：加入订单id，物流公司id，地址id作为外码且非空。

（9）．优惠券（优惠券id，优惠券开始时间，优惠券结束时间，优惠券金额，优惠券类型，优惠券使用范围，优惠券使用条件，优惠券使用范围id）

（10）．退货信息（退货订单，会员id，物流公司id，买家地址id，卖家地址id，申请时间，退货原因，退货运费，退货审核状态）

说明：会员id，物流公司id，买家地址id，卖家地址id均为外码且非空。

（11）．商品评价（评价id，会员id，订单id，评价内容，评分，评价时间，评价审核状态）

说明：会员id，订单id均为外码且非空。

（12）．会员关联优惠券（优惠券id，会员id，领取时间，数量）

说明：会员与优惠券之间的多对多联系使得该关系主码为两个：优惠券id，会员id。

3.1.2商家管理子系统

（13）．商品SPU（SPUid，类型id，类目id，品牌id，店铺id，商品名称，价格，详细信息，上架时间，下架时间，商品简介，商品评分，可售状态）

说明：由实体型生成的关系模式，spu和类型，类目，品牌，店铺是多对一的关系，因此在该表中加入类型id，类目id，品牌id，和店铺id作为外码。

（14）．商品sku（商品id，spu\_id,价格，库存）

说明：spu和sku是一对多的关系，因此在该表中加入spu\_id作为外码且为非空。

（15）．商品品牌（品牌id，类型id，品牌描述，品牌状态，修改时间，品牌创建时间）

说明：品牌和类型是多对一的关系，因此在该表中加入类型id作为外码且为非空。

（16）．商品类目（类目id，类型id，父类目，类目名称，类目状态）

说明：类型和类目是一对多的关系，因此在该表中加入类型id作为外码且为非空。

（17）．商品规格（规格id，类型id，规格名称）

说明：规格和类型是多对一的关系，因此在该表中加入类型id作为外码且为非空。

（18）．商品规格值（规格值id，规格id，规格值）

说明：规格和规格值是一对多的关系，因此在该表中加入规格id作为外码且为非空。

（19）．客服（客服id，店铺id。客服昵称，客服是否在线，客服信息）

说明：店铺和客服是一对多的关系，因此在该表中加入店铺id作为外码且为非空。

（20）．店铺（店铺id，会员id，广告id，店铺名称，店铺成立时间，店铺评分，关注人数，店铺联系方式，店铺类型，店铺状态）

说明及优化：店铺和会员，广告都是一对多的关系，因此在该表中加入会员id和广告id作为外码且都为非空，同时该关系由商家管理，将会员id改为商家id即可。

（21）．商品类型（类型id，类型名称）

3.1.3订单管理子系统

（22）．物流配送员（配送员id，物流公司id，配送员姓名，配送员联系方式）

说明：物流配送员和物流公司是多对一的关系，因此在该表中加入物流公司id作为外码且不为空。

（23）．物流公司（物流公司id，物流公司名称，物流公司联系人，物流公司电话，运费金额）

（24）．订单-商品（订单id，商品id，订单价格，订单购买数量）

由于订单-商品是订单和商品SKU之间的多对多关系而成，因此有两个主码。

3.1.4商城管理子系统

（25）．商城广告（广告id，管理员id，广告标题，广告链接，是否可关闭，广告开始时间，广告结束时间，广告类型）

说明及优化：管理员id为会员基本信息的会员id而来，由于会员基本信息可的用户可以是会员、商家、管理员，因此将本关系中该属性改为管理员id。

（26）．商城活动（活动id，管理员id，店铺id，活动名称，开始时间，结束时间，活动范围，活动类型）

说明及优化：管理员id来自会员基本信息关系，因该活动为管理员管理，因此将属性改为管理员id且非空，店铺id来自店铺关系。

（27）．满减活动（活动id，spu\_id，满减满足金额，满减优惠金额）

说明：由于多对多联系，生成了两个作为主码，即活动id和spu\_id。

（28）．直降活动（活动id，spu\_id，折扣百分比，最大订货量，最小订货量，原销售价，促销价）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 会员id | 主 | Unique  Not null |  | Varchar(10) |  |
| 会员级别id | 外 | Unique  Not null |  | Varchar(10) |  |
| 店铺id | 外 | Unique  Not null |  | Varchar(10) |  |
| 用户名 |  | Not null |  | Varchar(20) |  |
| 密码 |  | Not null |  | Varchar(20) |  |
| 旧密码 |  |  |  | Varchar(20) |  |
| 注册时间 |  |  |  | Date |  |
| 会员类型 |  | Not null |  | Char(2) | 1/2/3 |
| 是否绑定邮箱 |  |  |  | Boolean |  |
| 是否绑定qq |  |  |  | Boolean |  |
| 手机号码 |  |  |  | Varchar(15) |  |
| 注册审核状态 |  |  |  | Boolean |  |

说明：由于多对多凉席，生成了两个作为主码，即活动id和spu\_id。

3.2关系表总体设计说明

由于本系统在进行概念结构设计时，各个系统的负责人之间曾进行过多次交流，因此在合成的时候没有太多的冲突。各个部分也能比较好地结合到一起。

表27：会员基本信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 会员id | 主、外 | Unique  Not null |  | Varchar(10) |  |
| 邮箱 |  |  |  | Varchar(20) |  |
| qq |  |  |  | Varchar(15) |  |
| 支付密码 |  | Not null |  | Char(6) |  |
| 真实姓名 |  | Not null |  | Varchar(20) |  |
| 性别 |  |  | 男 | Char(6) |  |
| 证件类型 |  | Not null | 1 | Char(2) | 1/2/3 |
| 证件号码 |  | Not null |  | Varchar(20) |  |
| 出生日期 |  |  |  | Date |  |
| 会员积分 |  |  |  | Integer |  |

表28：会员详细信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 收藏id | 主 | Unique  Not null |  | Varchar(10) |  |
| 会员id | 外 | Unique  Not null |  | Varchar(10) |  |
| 收藏类型 |  | Not null | 1 | Char(2) | 1/2 |
| 状态 |  | Not null | 0 | Boolean |  |
| 商品/店铺id |  | Not null |  | Varchar(10) |  |
| 收藏夹创建时间 |  |  |  | Date |  |

表29：会员收藏夹

表30：会员收货地址

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 地址id | 主 | Unique  Not null |  | Varchar(10) |  |
| 会员id | 外 | Unique  Not null |  | Varchar(10) |  |
| 收货人姓名 |  | Not null |  | Varchar(20) |  |
| 收货人手机号码 |  | Not null |  | Varchar(15) |  |
| 省 |  | Not null |  | Varchar(10) |  |
| 市 |  | Not null |  | Varchar(10) |  |
| 区/县 |  | Not null |  | Varchar(10) |  |
| 街道/镇 |  | Not null |  | Varchar(10) |  |
| 详细地址 |  | Not null |  | Varchar(256) |  |
| 是否默认 |  | Not null | 1 | Boolean |  |

表31：会员等级

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 会员级别id | 主 | Unique  Not null |  | Varchar(10) |  |
| 级别名称 |  | Not null | 普通会员 | Varchar(20) |  |
| 该级最高积分 |  | Not null |  | Integer |  |
| 该级最低积分 |  | Not null |  | Integer |  |

表32：购物车

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 购物车id | 主 | Unique  Not null |  | Varchar(10) |  |
| 会员id | 外 | Unique  Not null |  | Varchar(10) |  |
| 商品id | 外 | Unique  Not null |  | Varchar(10) |  |
| 购物车商品状态 |  | Not null | 1 | Boolean |  |
| 购物车购买数量 |  | Not null |  | Integer |  |
| 购物车商品价格 |  |  |  | Decimal(8,2) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 订单id | 主 | Unique  Not null |  | Varchar(10) |  |
| 会员id | 外 | Unique  Not null |  | Varchar(10) |  |
| 店铺id | 外 | Unique  Not null |  | Varchar(10) |  |
| 支付时间 |  |  |  | Date&Time |  |
| 支付方式 |  | Not null | 1 | Char(2) | 1/2/3 |
| 订单总额 |  | Not null |  | Decimal(8,2) |  |
| 订单积分 |  | Not null |  | Integer |  |
| 运费险 |  | Not null | 0 | Boolean |  |
| 订单状态 |  |  |  | Char(2) |  |
| 订单优惠金额 |  | Not null |  | Decimal(8,2) |  |

表33：订单主表

表34：物流信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 快递单号 | 主 | U/Not null |  | Varchar(10) |  |
| 订单id | 外 | U/Not null |  | Varchar(10) |  |
| 物流公司id | 外 | U/Not null |  | Varchar(10) |  |
| 地址id | 外 | U/Not null |  | Varchar(10) |  |
| 物流状态 |  | Not null |  | Char(2) |  |
| 收货时间 |  |  |  | Date&Time |  |
| 发货时间 |  |  |  | Date&Time |  |

表35：优惠券

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 优惠券编号 | 主 | Not null |  | Varchar(10) |  |
| 优惠券开始时间 |  |  |  | Date&time |  |
| 优惠券结束时间 |  |  |  | Date&time |  |
| 优惠券金额 |  |  |  | Decimal(8,2) |  |
| 优惠券类型 |  |  |  | Boolean |  |
| 优惠券使用范围 |  |  |  | Integer |  |
| 优惠券使用条件 |  |  |  | Decimal(8,2) |  |
| 优惠券使用范围id |  |  |  | Varchar(10)d |  |

表36：退货

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 退货订单id | 主 | U/Not null |  | Varchar(10) |  |
| 会员id | 外 | U/Not null |  | Varchar(10) |  |
| 申请时间 |  |  |  | Date&Time |  |
| 退货运费 |  | Not null |  | Decimal(8,2) |  |
| 物流公司id | 外 | U/Not null |  | Varchar(10) |  |
| 买家地址id | 外 | U/Not null |  | Varchar(10) |  |
| 商家地址id | 外 | U/Not null |  | Varchar(10) |  |
| 审核状态 |  | Not null |  | Boolean |  |
| 退货原因 |  |  |  | Varchar(512) |  |

表37：商品评价

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 评价id： | 主 | U/Not null |  | Varchar（10） |  |
| 会员id | 外 | U/Not null |  | Varchar（10） |  |
| 订单id | 外 | U/Not null |  | Varchar（10） |  |
| 评价内容 |  |  |  | Varchar（1024） |  |
| 评分 |  | Not null |  | Integer | 1-5分 |
| 评价时间 |  |  |  | Date |  |
| 评价审核状态 |  | Not null |  | Boolean |  |

表37：会员关联优惠券

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 优惠券id | 主 | U/Not null |  | Varchar(10) |  |
| 会员id | 朱 | U/Not null |  | Varchar(10) |  |
| 领取时间 |  |  |  | Date&Time |  |
| 数量 |  |  |  | Integer |  |

表38：商品SPU

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| Spu\_id | 主 | Unique |  | Varchar（10） |  |
| 类型id | 外 | Unique |  | Varchar（10） |  |
| 类目id | 外 | Unique |  | Varchar（10） |  |
| 品牌id | 外 | Unique |  | Varchar（10） |  |
| 店铺id | 外 | U/Not null |  | Varchar（10） |  |
| 商品名称 |  | Not null |  | Varchar（20） |  |
| 商品价格 |  | Not null |  | Decimal（8，2） |  |
| 详细信息 |  |  |  | Varchar（1024） |  |
| 上架时间 |  |  |  | Date |  |
| 下架时间 |  |  |  | Date |  |
| 商品简介 |  |  |  | Varchar（512） |  |
| 商品评分 |  |  |  | Float |  |
| 可售状态 |  | Not null |  | Boolean |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 类型id | 主 | U/Not null |  | Varchar（10） |  |
| 类型名称 |  |  |  | Varchar（20） |  |

表39：商品类型

表40：商品规格

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 规格id | 主 | U/Not null |  | Varchar（10） |  |
| 类型id | 外 | U/Not null |  | Varchar（20） |  |
| 规格名称 |  | Not null |  | Varchar（20） |  |

表41：商品规格值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 规格值id | 主 | U/Not null |  | Varchar（10） |  |
| 规格id | 外 | U/Not null |  | Varchar（20） |  |
| 规格值 |  |  |  | Varchar（20） |  |

表42：商品品牌

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 品牌id： | 主 | U/Not null |  | Varchar（10） |  |
| 类型id | 外 | U/Not null |  | Varchar（20） |  |
| 品牌描述 |  |  |  | Varchar（512） |  |
| 品牌状态 |  | Not null |  | Boolean |  |
| 修改时间 |  |  |  | Date |  |
| 品牌创建时间 |  |  |  | Date |  |

表43：类目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 类目id | 主 | U/Not null |  | Varchar（10） |  |
| 父类目 |  |  |  | Varchar（10） |  |
| 类目名称 |  | Not null |  | Varchar（20） |  |
| 类型id | 外 | U/Not null |  | Varchar（10） |  |
| 类目状态 |  | Not null |  | Boolean |  |

表44：商品SKU

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 商品id | 主 | U/Not null |  | Varchar（10） |  |
| spu\_id | 外 | U/Not null |  | Varchar（10） |  |
| 价格 |  | Not null |  | Decimal（8,2） |  |
| 库存 |  | Not null |  | Integer |  |

表45：店铺

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 店铺id： | 主 | Unique |  | Varchar（10） |  |
| 会员id | 外 | Unique |  | Varchar（10） |  |
| 广告id | 外 | Unique |  | Varchar（10） |  |
| 店铺名称 |  | Not null |  | Varchar（20） |  |
| 店铺成立时间 |  |  |  | Date |  |
| 店铺评分 |  |  |  | Integer |  |
| 关注人数 |  |  |  | Integer |  |
| 店铺联系方式 |  | Not null |  | Varchar（15） |  |
| 店铺类型 |  |  |  | Varchar（20） |  |
| 店铺状态 |  | Not null |  | Boolean |  |

表46：客服

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 客服id | 主 | U/Not null |  | Varchar（10） |  |
| 店铺id | 外 | U/Not null |  | Varchar（10） |  |
| 客服昵称 |  | Not null |  | Varchar（20） |  |
| 客服是否在线 |  | Not null |  | Boolean |  |
| 客服信息 |  |  |  | Varchar（512） |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 订单id | 主/外 | U/Not null |  | Varchar(10) |  |
| 商品id | 主/外 | U/Not null |  | Varchar(10) |  |
| 订单价格 |  | Not null |  | Integer |  |
| 订单购买数量 |  | Not null |  | Decimal(8,2) |  |

表47：订单-商品

表48：物流公司

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 物流公司id | 主 | Not null |  | Varchar(10) |  |
| 物流公司联系人 |  |  |  | Varchar(20) |  |
| 物流公司名称 |  | Not null |  | Varchar(20) |  |
| 物流公司电话 |  |  |  | Varchar(15) |  |
| 运费金额 |  | Not null |  | Decimal(8,2) |  |

表49：物流配送员

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 配送员id | 主 | U/Not null |  | Varchar(10) |  |
| 物流公司id | 外 | U/Not null |  | Varchar(10) |  |
| 配送员姓名 |  | Not null |  | Varchar(20) |  |
| 配送员联系方式 |  | Not null |  | Varchar(15) |  |

表49：商城广告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 广告id | 主 | U/Not null |  | Varchar(10) |  |
| 管理员id | 外 | U/Not null |  | Varchar(10) |  |
| 广告标题 |  |  |  | Varchar(20) |  |
| 广告链接 |  | Not null |  | Varchar(256) |  |
| 是否可关闭 |  | Not null |  | Boolean |  |
| 广告开始时间 |  |  |  | Date |  |
| 广告结束时间 |  |  |  | Date |  |
| 广告类型 |  | Not null |  | Char(2) |  |

表50:商城活动

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 活动id | 主 | U/t null |  | Varchar(10) |  |
| 管理员id | 外 | U/Not null |  | Varchar(10) |  |
| 店铺id | 外 | U/Not null |  | Varchar(10) |  |
| 活动名称 |  |  |  | Varchar(20) |  |
| 开始时间 |  | Not null |  | Date |  |
| 结束时间 |  | Not null |  | Date |  |
| 活动范围 |  | Not null | 1 | Char (2) | 1/2 |
| 活动类型 |  | Not null | 1 | Char (2) | 1/2 |

表51商城满减活动

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| 活动id | 主/外 | U/Not null |  | Varchar(10) |  |
| spu\_id | 主/外 | U/Not null |  | Varchar(10) |  |
| 满足金额 |  | Not null |  | Decimal(8,2) |  |
| 满减优惠金额 |  | Not null |  | Decimal(8,2) |  |

表52商城直降活动

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key Type | Nulls/Unique | Default | Data Type | Sample Data |
| spu\_id | 主/外 | U/Not null |  | Varchar(10) |  |
| 活动id | 主/外 | U/Not null |  | Varchar(10) |  |
| 最大订货量 |  | Not null |  | Integer |  |
| 最小订货量 |  | Not null |  | Integer |  |
| 原销售量 |  | Not null |  | Decimal(8,2) |  |
| 促销价 |  | Not null |  | Decimal(8,2) |  |

3.3设计用户子模式

在将概念模型转化为全局逻辑模型后，根据医院系统的局部应用需求，接下来设计用户子模式。

考虑以下几方面的需求：

（1）会员查看商品、店铺、商城相关信息

（2）商家管理店铺、查看报表统计等相关信息

（3）管理员管理会员、商家、商城以及查阅对应信息

本系统定义了如下具有不同操作权限的用户级别：

对于会员、商家、管理员的级别定义如下：

（1）会员：对会员管理子系统所有关系的查询、修改权限，对商家管理子系统中部分关系的查询权限，对商城部分信息的查看权限；

（2）商家：对自身相关信息的查询、修改权限，对商家管理子系统的所有关系的查询、修改权限，对商城部分信息的查看权限；

（3）管理员：对所有关系的查看和修改权限。

制作查询子系统

根据分析需求，我们设计了以下几个查询系统：会员查询子系统、商家查询子系统、管理员查询子系统。

（1）管理员查询子系统

本系统是根据会员基本信息关系中会员类型的划分来区分会员、商家和管理员的，因此，首先需要创建三个试图用于区分三种不同的用户：

会员基本信息（会员id，会员级别id，用户名，密码，旧密码，注册时间，是否绑定邮箱，是否绑定qq，手机号码）

商家基本信息（商家id，店铺id，用户名，密码，旧密码，注册时间，是否绑定邮箱，是否绑定qq，手机号码）

管理员基本信息（管理员id，用户名，密码，旧密码，注册时间，是否绑定邮箱，是否绑定qq，手机号码）

注：以上三个视图分别通过会员类型字段来判断，当会员类型字段为1时表示为会员，2表示商家，3表示管理员

为了便于对该系统进行数据统计，建立了两个报表视图：

会员报表（会员id，创建时间，下单总数，购买商品总额，购买店铺数）

注：对于同一个会员，可能有多个报表，可以根据创建时间来判断不同的数据的不同，创建时间即在该时间统计会员在该时间之前所有的购买商品总额和购买店铺数。

商家报表（店铺id，订单数量，销量，客户数，spu\_id）

注：该报表用于后期对于商城的扩展，可以对商家店铺进行销量排名，客户数排名，订单数量排名等，注意销量对应的是一个spu\_id的销量。

（2）会员查询子系统

本系统用于会员的相关操作，对于会员而言，本系统有一个关系为会员收藏夹，该字段用于记录会员收藏的店铺和商品，有一个收藏类型字段用于区分收藏的是商品还是店铺，因此为了便于统计，可以建立两个试图用于区分会员商品收藏和会员店铺收藏。

会员商品收藏（收藏id，sku\_id，会员id，创建时间，状态）

会员店铺收藏（收藏id，店铺id，会员id，创建时间，状态）

注：当收藏类型为1时表示收藏商品，收藏类型为2时表示收藏店铺。

（3）商家查询子系统

该系统用于商家进行相关信息的查询，包括店铺商品评分，店铺收入等，为了便于统计，可以建立两个个试图来分别统计店铺商品评分、店铺收入。

店铺商品评分（店铺id，spu\_id，评分）

注：评分是根据spu\_id来评分的，即对于店铺的一个商品类评分，同时该视图还可以便于统计店铺的评分。

店铺收入（店铺id，店铺收入）

注：本视图可以根据订单数来判断，订单有一个订单金额，一个店铺有多个订单，从而计算出店铺的总收入。

4.物理结构设计

京东电商系统是多用户共享的系统，对同一个关系要建立多条存储路径，才能满足多用户的多种应用要求。在数据库设计中，可供选用的存储方法主要有三种：Ｂ＋树索引方法、聚簇方法、ＨＡＳＨ方法。其中，Ｂ＋数索引方法发展较为完善，也是目前大型数据库使用的经典存取方法，使用最为普遍，并且符合本系统的要求，因此，我们采用Ｂ＋树索引方法作为系统的存取方法。

4.1建立索引

所谓选择索引存取方法实际上就是根据应用要求确定对关系的哪些属性列建立索引、哪些属性列建立组合索引、哪些索引要设计为唯一索引等。根据本信息系统的具体情况，建立索引的依据如下：

一个属性经常在查询条件下出现。

一个属性经常作为最大值或最小值等聚集函数的参数。

一个属性经常在连接操作的连接条件出现。

需要建立索引的属性列表如下。

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **对应实体** | **属性名** | **对应实体** |
| spu\_id | 商品SPU | 类目id | 商品SPU |
| 类型id | 商品SPU | 品牌id | 商品SPU |
| 店铺id | 商品SPU | 品牌id | 商品品牌 |
| 类型id | 商品品牌 | 类目id | 商品类目 |
| 类型id | 商品类目 | 规格id | 商品规格 |
| 类型id | 商品规格 | 规格值id | 商品规格值 |
| 规格id | 商品规格值 | 类型id | 商品类型 |
| 评价id | 评价商品 | 会员id | 评价商品 |
| 订单id | 评价商品 | 优惠券id | 优惠券 |
| 物流公司id | 物流公司 | 快递单号 | 物流信息 |
| 订单id | 物流信息 | 物流公司id | 物流信息 |
| 地址id | 物流信息 | spu\_id | 直降活动 |
| 活动id | 直降活动 | 配送员id | 物流配送员 |
| 物流公司id | 物流配送员 | spu\_id | 满减活动 |
| 活动id | 满减活动 | 活动id，spu\_id | 满减活动 |
| goods\_PK | 商品SKU | SPU\_SKU\_FK | 商品SKU |
| 活动id | 商城活动 | 管理员id | 商城活动 |
| 店铺id | 商城活动 | 广告id | 商城广告 |
| 管理员id | 商城广告 | 订单id，商品id | 订单-商品 |
| 订单id | 订单-商品 | 商品id | 订单-商品 |
| 订单id | 订单主信息 | 店铺id | 订单主信息 |
| 会员id | 订单主信息 | 退货订单 | 退货信息 |
| 会员id | 退货信息 | 物流公司id | 退货信息 |
| 买家地址id | 退货信息 | 卖家地址id | 退货信息 |
| 客服id | 客服 | 店铺id | 客服 |
| 购物车id | 购物车 | 会员id | 购物车 |
| 商品id | 购物车 | 店铺id | 店铺 |
| 会员id | 店铺 | 广告id | 店铺 |
| 地址id | 会员收货地址 | 会员id | 会员收货地址 |
| 会员id | 会员基本信息 | 会员级别id | 会员基本信息 |
| 店铺id | 会员基本信息 | 优惠券id，会员id | 会员关联优惠券 |
| 优惠券id | 会员关联优惠券 | 会员id | 会员关联优惠券 |
| 会员id | 会员详细信息 | 收藏夹id | 会员收藏夹 |
| 会员id | 会员收藏夹 | 会员级别id | 会员等级 |

4.2确定数据库的存储结构

本部分主要是确定数据库物理结构，即确定数据的存放位置和存储结构，包括确定关系、索引、聚簇、日志、备份等存储安排和存储结构，确定系统配置等。

在综合考虑存取时间、存储空间利用率和维护代价三方面的因素下，经过权衡，确定下数据的存放位置和存储结构。

（1）确定数据的存放位置

为了提高系统的性能，根据应用情况将数据的易变部分与稳定部分、经常存取部分与存取效率较低部分分开存放。

考虑到目前许多计算机都有多个磁盘，因此将表（如商家管理子系统表）和索引放在不同的磁盘上，在查询时，两个磁盘驱动器并行工作，可以大大提高I/O读写的效率；而对于较大的表（如商家管理子系统的表、用户管理子系统的表）则分放在两个磁盘上，这样可以加快存取速度，在多用户环境下的作用更有效；为了改进系统的性能，将日志文件与数据库对象（表、索引等）放在了不同的盘上。

（2）确定系统配置

在数据库设计初始阶段，为了对以后的数据库进行物理优化，提供了一些系统配置变量、存储分配系数，并对这些变量赋予了合理的缺省值。但是这些值不一定适合每一种应用环境，在进行物理设计时，对这些变量重新赋值，以改善系统的性能。

对系统配置的变量，例如：同时使用数据库的用户数，同时打开的数据库对象数，内存分配参数，缓冲区分配参数（使用的缓冲区长度、个数），存储分配参数，物理块的大小，物理块装填因子，时间片大小，数据库的大小，锁的数目等。这些参数值直接影响存取时间和存储空间的分配，在物理设计时应根据应用环境确定这些参数值，以使系统性能最佳。

5.数据库实现

drop index store\_SPU\_FK;

drop index SPU\_brand\_SPU\_FK;

drop index SPU\_category\_SPU\_FK;

drop index SPU\_type\_SPU\_FK;

drop index SPU\_PK;

drop table SPU;

drop index SPU\_type\_brand\_FK;

drop index SPU\_brand\_PK;

drop table SPU\_brand;

drop index SPU\_type\_category\_FK;

drop index SPU\_category\_PK;

drop table SPU\_category;

drop index SPU\_type\_spec\_FK;

drop index SPU\_spec\_PK;

drop table SPU\_spec;

drop index SPU\_spec\_spec\_value\_FK;

drop index SPU\_spec\_value\_PK;

drop table SPU\_spec\_value;

drop index SPU\_type\_PK;

drop table SPU\_type;

drop index order\_main\_info\_comment\_goods\_FK;

drop index user\_basic\_info\_comment\_goods\_FK;

drop index comment\_goods\_PK;

drop table comment\_goods;

drop index coupon\_PK;

drop table coupon;

drop index delivery\_company\_PK;

drop table delivery\_company;

drop index user\_add\_delivery\_info\_FK;

drop index delivery\_company\_info\_FK;

drop index order\_main\_info\_delivery\_info\_FK;

drop index delivery\_info\_PK;

drop table delivery\_info;

drop index discending\_SPU\_FK;

drop index activity\_discending\_FK;

drop table discending\_SPU;

drop index delivery\_company\_distributor\_FK;

drop index distributor\_PK;

drop table distributor;

drop index full\_reduction\_SPU\_FK;

drop index activity\_full\_reduction\_FK;

drop index full\_reduction\_SPU\_PK;

drop table full\_reduction\_SPU;

drop index SPU\_SKU\_FK;

drop index goods\_PK;

drop table goods;

drop index activity\_store\_FK;

drop index administractor\_activity\_FK;

drop index mall\_activity\_PK;

drop table mall\_activity;

drop index administractor\_adver\_FK;

drop index mall\_ad\_PK;

drop table mall\_ad;

drop index order\_goods\_main\_info\_FK;

drop index SKU\_order\_goods\_FK;

drop index order\_goods\_PK;

drop table order\_goods;

drop index store\_order\_main\_info\_FK;

drop index user\_basic\_info\_order\_main\_info\_FK;

drop index order\_main\_info\_PK;

drop table order\_main\_info;

drop index business\_add\_return\_goods\_FK;

drop index buyer\_add\_return\_goods\_FK;

drop index delivery\_company\_return\_goods\_FK;

drop index user\_basic\_info\_return\_goods\_FK;

drop index return\_goods\_PK;

drop table return\_goods;

drop index store\_server\_FK;

drop index server\_PK;

drop table server;

drop index SKU\_shopping\_cart\_FK;

drop index user\_basic\_info\_shopping\_cart\_FK;

drop index shopping\_cart\_PK;

drop table shopping\_cart;

drop index adver\_store\_FK;

drop index user\_store2\_FK;

drop index store\_PK;

drop table store;

drop index user\_basic\_info\_add\_FK;

drop index user\_add\_PK;

drop table user\_add;

drop index user\_store\_FK;

drop index upgrade\_FK;

drop index user\_basic\_info\_PK;

drop table user\_basic\_info;

drop index coupon\_is\_have\_FK;

drop index user\_have\_coupon\_FK;

drop index user\_basic\_info\_coupon\_PK;

drop table user\_basic\_info\_coupon;

drop index user\_detail\_info\_PK;

drop table user\_detail\_info;

drop index user\_basic\_info\_favorite\_FK;

drop index user\_favorite\_PK;

drop table user\_favorite;

drop index user\_rank\_PK;

drop table user\_rank;

/\*==============================================================\*/

/\* Table: SPU \*/

/\*==============================================================\*/

create table SPU (

spu\_id VARCHAR(10) not null,

type\_id VARCHAR(10) null,

category\_id VARCHAR(10) null,

brand\_id VARCHAR(10) null,

store\_id VARCHAR(10) not null,

spu\_name VARCHAR(20) not null,

cart\_spu\_price DECIMAL(8,2) not null,

spu\_detail\_info VARCHAR(1024) null,

spu\_start\_time DATE null,

spu\_end\_time DATE null,

spu\_brief\_info VARCHAR(512) null,

spu\_point FLOAT8 null,

spu\_status BOOL not null,

constraint PK\_SPU primary key (spu\_id)

);

/\*==============================================================\*/

/\* Index: SPU\_PK \*/

/\*==============================================================\*/

create unique index SPU\_PK on SPU (

spu\_id

);

/\*==============================================================\*/

/\* Index: SPU\_type\_SPU\_FK \*/

/\*==============================================================\*/

create index SPU\_type\_SPU\_FK on SPU (

type\_id

);

/\*==============================================================\*/

/\* Index: SPU\_category\_SPU\_FK \*/

/\*==============================================================\*/

create index SPU\_category\_SPU\_FK on SPU (

category\_id

);

/\*==============================================================\*/

/\* Index: SPU\_brand\_SPU\_FK \*/

/\*==============================================================\*/

create index SPU\_brand\_SPU\_FK on SPU (

brand\_id

);

/\*==============================================================\*/

/\* Index: store\_SPU\_FK \*/

/\*==============================================================\*/

create index store\_SPU\_FK on SPU (

store\_id

);

/\*==============================================================\*/

/\* Table: SPU\_brand \*/

/\*==============================================================\*/

create table SPU\_brand (

brand\_id VARCHAR(10) not null,

type\_id VARCHAR(10) not null,

brand\_info VARCHAR(512) null,

brand\_status BOOL not null,

brand\_modified\_time DATE null,

brand\_create\_time DATE null,

constraint PK\_SPU\_BRAND primary key (brand\_id)

);

/\*==============================================================\*/

/\* Index: SPU\_brand\_PK \*/

/\*==============================================================\*/

create unique index SPU\_brand\_PK on SPU\_brand (

brand\_id

);

/\*==============================================================\*/

/\* Index: SPU\_type\_brand\_FK \*/

/\*==============================================================\*/

create index SPU\_type\_brand\_FK on SPU\_brand (

type\_id

);

/\*==============================================================\*/

/\* Table: SPU\_category \*/

/\*==============================================================\*/

create table SPU\_category (

category\_id VARCHAR(10) not null,

type\_id VARCHAR(10) not null,

category\_parent VARCHAR(10) null,

category\_name VARCHAR(20) not null,

category\_status BOOL not null,

constraint PK\_SPU\_CATEGORY primary key (category\_id)

);

/\*==============================================================\*/

/\* Index: SPU\_category\_PK \*/

/\*==============================================================\*/

create unique index SPU\_category\_PK on SPU\_category (

category\_id

);

/\*==============================================================\*/

/\* Index: SPU\_type\_category\_FK \*/

/\*==============================================================\*/

create index SPU\_type\_category\_FK on SPU\_category (

type\_id

);

/\*==============================================================\*/

/\* Table: SPU\_spec \*/

/\*==============================================================\*/

create table SPU\_spec (

spe\_id VARCHAR(10) not null,

type\_id VARCHAR(10) not null,

spec\_name VARCHAR(20) not null,

constraint PK\_SPU\_SPEC primary key (spe\_id)

);

/\*==============================================================\*/

/\* Index: SPU\_spec\_PK \*/

/\*==============================================================\*/

create unique index SPU\_spec\_PK on SPU\_spec (

spe\_id

);

/\*==============================================================\*/

/\* Index: SPU\_type\_spec\_FK \*/

/\*==============================================================\*/

create index SPU\_type\_spec\_FK on SPU\_spec (

type\_id

);

/\*==============================================================\*/

/\* Table: SPU\_spec\_value \*/

/\*==============================================================\*/

create table SPU\_spec\_value (

spec\_value\_id VARCHAR(10) not null,

spe\_id VARCHAR(10) not null,

spec\_value VARCHAR(20) null,

constraint PK\_SPU\_SPEC\_VALUE primary key (spec\_value\_id)

);

/\*==============================================================\*/

/\* Index: SPU\_spec\_value\_PK \*/

/\*==============================================================\*/

create unique index SPU\_spec\_value\_PK on SPU\_spec\_value (

spec\_value\_id

);

/\*==============================================================\*/

/\* Index: SPU\_spec\_spec\_value\_FK \*/

/\*==============================================================\*/

create index SPU\_spec\_spec\_value\_FK on SPU\_spec\_value (

spe\_id

);

/\*==============================================================\*/

/\* Table: SPU\_type \*/

/\*==============================================================\*/

create table SPU\_type (

type\_id VARCHAR(10) not null,

type\_name VARCHAR(20) null,

constraint PK\_SPU\_TYPE primary key (type\_id)

);

/\*==============================================================\*/

/\* Index: SPU\_type\_PK \*/

/\*==============================================================\*/

create unique index SPU\_type\_PK on SPU\_type (

type\_id

);

/\*==============================================================\*/

/\* Table: comment\_goods \*/

/\*==============================================================\*/

create table comment\_goods (

comment\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

order\_id VARCHAR(10) not null,

comment VARCHAR(1024) null,

comment\_point INT4 not null default 1

constraint CKC\_COMMENT\_POINT\_COMMENT\_ check (comment\_point in (1,2,3,4,5)),

comment\_date DATE null,

comment\_goods\_status BOOL not null,

constraint PK\_COMMENT\_GOODS primary key (comment\_id)

);

/\*==============================================================\*/

/\* Index: comment\_goods\_PK \*/

/\*==============================================================\*/

create unique index comment\_goods\_PK on comment\_goods (

comment\_id

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_comment\_goods\_FK \*/

/\*==============================================================\*/

create index user\_basic\_info\_comment\_goods\_FK on comment\_goods (

user\_id

);

/\*==============================================================\*/

/\* Index: order\_main\_info\_comment\_goods\_FK \*/

/\*==============================================================\*/

create index order\_main\_info\_comment\_goods\_FK on comment\_goods (

order\_id

);

/\*==============================================================\*/

/\* Table: coupon \*/

/\*==============================================================\*/

create table coupon (

coupon\_id VARCHAR(10) not null,

coupon\_start\_time DATE null,

coupon\_end\_time DATE null,

coupon\_money DECIMAL(8,2) null,

coupon\_type INT4 null default 1

constraint CKC\_COUPON\_TYPE\_COUPON check (coupon\_type is null or (coupon\_type in (1,2,3))),

coupon\_use\_range INT4 null default 1

constraint CKC\_COUPON\_USE\_RANGE\_COUPON check (coupon\_use\_range is null or (coupon\_use\_range in (1,2,3))),

coupon\_condition DECIMAL(8,2) null,

coupon\_use\_id VARCHAR(10) null,

constraint PK\_COUPON primary key (coupon\_id)

);

/\*==============================================================\*/

/\* Index: coupon\_PK \*/

/\*==============================================================\*/

create unique index coupon\_PK on coupon (

coupon\_id

);

/\*==============================================================\*/

/\* Table: delivery\_company \*/

/\*==============================================================\*/

create table delivery\_company (

company\_id VARCHAR(10) not null,

company\_name VARCHAR(20) not null,

company\_contact\_name VARCHAR(20) null,

company\_contact\_phone VARCHAR(15) null,

company\_money DECIMAL(8,2) not null,

constraint PK\_DELIVERY\_COMPANY primary key (company\_id)

);

/\*==============================================================\*/

/\* Index: delivery\_company\_PK \*/

/\*==============================================================\*/

create unique index delivery\_company\_PK on delivery\_company (

company\_id

);

/\*==============================================================\*/

/\* Table: delivery\_info \*/

/\*==============================================================\*/

create table delivery\_info (

delivery\_id VARCHAR(10) not null,

order\_id VARCHAR(10) not null,

company\_id VARCHAR(10) not null,

add\_id VARCHAR(10) not null,

delivery\_start\_time DATE null,

delivery\_status CHAR(2) not null default '1'

constraint CKC\_DELIVERY\_STATUS\_DELIVERY check (delivery\_status in ('1','2','3','4','5')),

delivery\_end\_time DATE null,

constraint PK\_DELIVERY\_INFO primary key (delivery\_id)

);

/\*==============================================================\*/

/\* Index: delivery\_info\_PK \*/

/\*==============================================================\*/

create unique index delivery\_info\_PK on delivery\_info (

delivery\_id

);

/\*==============================================================\*/

/\* Index: order\_main\_info\_delivery\_info\_FK \*/

/\*==============================================================\*/

create index order\_main\_info\_delivery\_info\_FK on delivery\_info (

order\_id

);

/\*==============================================================\*/

/\* Index: delivery\_company\_info\_FK \*/

/\*==============================================================\*/

create index delivery\_company\_info\_FK on delivery\_info (

company\_id

);

/\*==============================================================\*/

/\* Index: user\_add\_delivery\_info\_FK \*/

/\*==============================================================\*/

create index user\_add\_delivery\_info\_FK on delivery\_info (

add\_id

);

/\*==============================================================\*/

/\* Table: discending\_SPU \*/

/\*==============================================================\*/

create table discending\_SPU (

spu\_id VARCHAR(10) not null,

activity\_id VARCHAR(10) not null,

discending\_discount\_percent DECIMAL(8,2) not null,

discending\_max\_num INT4 not null,

discending\_min\_num INT4 not null,

discending\_former\_price DECIMAL(8,2) not null,

discending\_discount\_price DECIMAL(8,2) not null

);

/\*==============================================================\*/

/\* Index: activity\_discending\_FK \*/

/\*==============================================================\*/

create index activity\_discending\_FK on discending\_SPU (

activity\_id

);

/\*==============================================================\*/

/\* Index: discending\_SPU\_FK \*/

/\*==============================================================\*/

create index discending\_SPU\_FK on discending\_SPU (

spu\_id

);

/\*==============================================================\*/

/\* Table: distributor \*/

/\*==============================================================\*/

create table distributor (

distributor\_id VARCHAR(10) not null,

company\_id VARCHAR(10) not null,

distributor\_name VARCHAR(20) not null,

distributor\_phone VARCHAR(15) not null,

constraint PK\_DISTRIBUTOR primary key (distributor\_id)

);

/\*==============================================================\*/

/\* Index: distributor\_PK \*/

/\*==============================================================\*/

create unique index distributor\_PK on distributor (

distributor\_id

);

/\*==============================================================\*/

/\* Index: delivery\_company\_distributor\_FK \*/

/\*==============================================================\*/

create index delivery\_company\_distributor\_FK on distributor (

company\_id

);

/\*==============================================================\*/

/\* Table: full\_reduction\_SPU \*/

/\*==============================================================\*/

create table full\_reduction\_SPU (

spu\_id VARCHAR(10) not null,

activity\_id VARCHAR(10) not null,

reduction\_meet\_price DECIMAL(8,2) not null,

reduction\_discount\_price DECIMAL(8,2) not null,

constraint PK\_FULL\_REDUCTION\_SPU primary key (spu\_id, activity\_id)

);

/\*==============================================================\*/

/\* Index: full\_reduction\_SPU\_PK \*/

/\*==============================================================\*/

create unique index full\_reduction\_SPU\_PK on full\_reduction\_SPU (

spu\_id,

activity\_id

);

/\*==============================================================\*/

/\* Index: activity\_full\_reduction\_FK \*/

/\*==============================================================\*/

create index activity\_full\_reduction\_FK on full\_reduction\_SPU (

activity\_id

);

/\*==============================================================\*/

/\* Index: full\_reduction\_SPU\_FK \*/

/\*==============================================================\*/

create index full\_reduction\_SPU\_FK on full\_reduction\_SPU (

spu\_id

);

/\*==============================================================\*/

/\* Table: goods \*/

/\*==============================================================\*/

create table goods (

goods\_id VARCHAR(10) not null,

spu\_id VARCHAR(10) not null,

sku\_goods\_price DECIMAL(8,2) not null,

sku\_goods\_num INT4 not null,

constraint PK\_GOODS primary key (goods\_id)

);

/\*==============================================================\*/

/\* Index: goods\_PK \*/

/\*==============================================================\*/

create unique index goods\_PK on goods (

goods\_id

);

/\*==============================================================\*/

/\* Index: SPU\_SKU\_FK \*/

/\*==============================================================\*/

create index SPU\_SKU\_FK on goods (

spu\_id

);

/\*==============================================================\*/

/\* Table: mall\_activity \*/

/\*==============================================================\*/

create table mall\_activity (

activity\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

store\_id VARCHAR(10) not null,

activity\_name VARCHAR(20) null,

activity\_start\_time DATE null,

activity\_end\_time DATE null,

activity\_range CHAR(2) null default '1'

constraint CKC\_ACTIVITY\_RANGE\_MALL\_ACT check (activity\_range is null or (activity\_range in ('1','2'))),

activity\_type CHAR(2) null default '1'

constraint CKC\_ACTIVITY\_TYPE\_MALL\_ACT check (activity\_type is null or (activity\_type in ('1','2'))),

constraint PK\_MALL\_ACTIVITY primary key (activity\_id)

);

/\*==============================================================\*/

/\* Index: mall\_activity\_PK \*/

/\*==============================================================\*/

create unique index mall\_activity\_PK on mall\_activity (

activity\_id

);

/\*==============================================================\*/

/\* Index: administractor\_activity\_FK \*/

/\*==============================================================\*/

create index administractor\_activity\_FK on mall\_activity (

user\_id

);

/\*==============================================================\*/

/\* Index: activity\_store\_FK \*/

/\*==============================================================\*/

create index activity\_store\_FK on mall\_activity (

store\_id

);

/\*==============================================================\*/

/\* Table: mall\_ad \*/

/\*==============================================================\*/

create table mall\_ad (

adver\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

adver\_title VARCHAR(20) not null,

adver\_url VARCHAR(256) not null,

adver\_is\_closed BOOL not null,

adver\_start\_time DATE null,

adver\_end\_time DATE null,

adver\_type CHAR(2) not null default '1'

constraint CKC\_ADVER\_TYPE\_MALL\_AD check (adver\_type in ('1','2')),

constraint PK\_MALL\_AD primary key (adver\_id)

);

/\*==============================================================\*/

/\* Index: mall\_ad\_PK \*/

/\*==============================================================\*/

create unique index mall\_ad\_PK on mall\_ad (

adver\_id

);

/\*==============================================================\*/

/\* Index: administractor\_adver\_FK \*/

/\*==============================================================\*/

create index administractor\_adver\_FK on mall\_ad (

user\_id

);

/\*==============================================================\*/

/\* Table: order\_goods \*/

/\*==============================================================\*/

create table order\_goods (

order\_id VARCHAR(10) not null,

goods\_id VARCHAR(10) not null,

order\_goods\_price DECIMAL(8,2) not null,

order\_goods\_num INT4 not null,

constraint PK\_ORDER\_GOODS primary key (order\_id, goods\_id)

);

/\*==============================================================\*/

/\* Index: order\_goods\_PK \*/

/\*==============================================================\*/

create unique index order\_goods\_PK on order\_goods (

order\_id,

goods\_id

);

/\*==============================================================\*/

/\* Index: SKU\_order\_goods\_FK \*/

/\*==============================================================\*/

create index SKU\_order\_goods\_FK on order\_goods (

goods\_id

);

/\*==============================================================\*/

/\* Index: order\_goods\_main\_info\_FK \*/

/\*==============================================================\*/

create index order\_goods\_main\_info\_FK on order\_goods (

order\_id

);

/\*==============================================================\*/

/\* Table: order\_main\_info \*/

/\*==============================================================\*/

create table order\_main\_info (

order\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

store\_id VARCHAR(10) not null,

order\_pay\_time DATE null,

order\_type CHAR(2) not null default '1'

constraint CKC\_ORDER\_TYPE\_ORDER\_MA check (order\_type in ('1','2','3')),

order\_money DECIMAL(8,2) not null,

order\_point INT4 not null,

order\_assurance BOOL not null default false,

order\_status CHAR(2) null default '1'

constraint CKC\_ORDER\_STATUS\_ORDER\_MA check (order\_status is null or (order\_status in ('1','2','3','4'))),

oder\_discount DECIMAL(8,2) not null,

constraint PK\_ORDER\_MAIN\_INFO primary key (order\_id)

);

/\*==============================================================\*/

/\* Index: order\_main\_info\_PK \*/

/\*==============================================================\*/

create unique index order\_main\_info\_PK on order\_main\_info (

order\_id

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_order\_main\_info\_FK \*/

/\*==============================================================\*/

create index user\_basic\_info\_order\_main\_info\_FK on order\_main\_info (

user\_id

);

/\*==============================================================\*/

/\* Index: store\_order\_main\_info\_FK \*/

/\*==============================================================\*/

create index store\_order\_main\_info\_FK on order\_main\_info (

store\_id

);

/\*==============================================================\*/

/\* Table: return\_goods \*/

/\*==============================================================\*/

create table return\_goods (

return\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

company\_id VARCHAR(10) not null,

buyer\_add\_id VARCHAR(10) not null,

business\_add\_id VARCHAR(10) not null,

return\_time DATE null,

return\_reason VARCHAR(512) null,

return\_money DECIMAL(8,2) not null,

return\_goods\_status BOOL not null,

constraint PK\_RETURN\_GOODS primary key (return\_id)

);

/\*==============================================================\*/

/\* Index: return\_goods\_PK \*/

/\*==============================================================\*/

create unique index return\_goods\_PK on return\_goods (

return\_id

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_return\_goods\_FK \*/

/\*==============================================================\*/

create index user\_basic\_info\_return\_goods\_FK on return\_goods (

user\_id

);

/\*==============================================================\*/

/\* Index: delivery\_company\_return\_goods\_FK \*/

/\*==============================================================\*/

create index delivery\_company\_return\_goods\_FK on return\_goods (

company\_id

);

/\*==============================================================\*/

/\* Index: buyer\_add\_return\_goods\_FK \*/

/\*==============================================================\*/

create index buyer\_add\_return\_goods\_FK on return\_goods (

buyer\_add\_id

);

/\*==============================================================\*/

/\* Index: business\_add\_return\_goods\_FK \*/

/\*==============================================================\*/

create index business\_add\_return\_goods\_FK on return\_goods (

business\_add\_id

);

/\*==============================================================\*/

/\* Table: server \*/

/\*==============================================================\*/

create table server (

server\_id VARCHAR(10) not null,

store\_id VARCHAR(10) not null,

server\_name VARCHAR(20) null,

server\_status BOOL null,

server\_info VARCHAR(512) null,

constraint PK\_SERVER primary key (server\_id)

);

/\*==============================================================\*/

/\* Index: server\_PK \*/

/\*==============================================================\*/

create unique index server\_PK on server (

server\_id

);

/\*==============================================================\*/

/\* Index: store\_server\_FK \*/

/\*==============================================================\*/

create index store\_server\_FK on server (

store\_id

);

/\*==============================================================\*/

/\* Table: shopping\_cart \*/

/\*==============================================================\*/

create table shopping\_cart (

cart\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

goods\_id VARCHAR(10) not null,

cart\_goods\_status BOOL not null default true,

cart\_goods\_num INT4 not null,

cart\_goods\_price DECIMAL(8,2) null,

constraint PK\_SHOPPING\_CART primary key (cart\_id)

);

/\*==============================================================\*/

/\* Index: shopping\_cart\_PK \*/

/\*==============================================================\*/

create unique index shopping\_cart\_PK on shopping\_cart (

cart\_id

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_shopping\_cart\_FK \*/

/\*==============================================================\*/

create index user\_basic\_info\_shopping\_cart\_FK on shopping\_cart (

user\_id

);

/\*==============================================================\*/

/\* Index: SKU\_shopping\_cart\_FK \*/

/\*==============================================================\*/

create index SKU\_shopping\_cart\_FK on shopping\_cart (

goods\_id

);

/\*==============================================================\*/

/\* Table: store \*/

/\*==============================================================\*/

create table store (

store\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

adver\_id VARCHAR(10) null,

store\_name VARCHAR(20) not null,

store\_create\_time DATE null,

store\_point INT4 null,

store\_follower INT4 null,

store\_phone VARCHAR(15) not null,

store\_type VARCHAR(20) null,

store\_status BOOL not null,

constraint PK\_STORE primary key (store\_id)

);

/\*==============================================================\*/

/\* Index: store\_PK \*/

/\*==============================================================\*/

create unique index store\_PK on store (

store\_id

);

/\*==============================================================\*/

/\* Index: user\_store2\_FK \*/

/\*==============================================================\*/

create index user\_store2\_FK on store (

user\_id

);

/\*==============================================================\*/

/\* Index: adver\_store\_FK \*/

/\*==============================================================\*/

create index adver\_store\_FK on store (

adver\_id

);

/\*==============================================================\*/

/\* Table: user\_add \*/

/\*==============================================================\*/

create table user\_add (

add\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

receiver\_name VARCHAR(20) null,

receiver\_phone VARCHAR(15) null,

province VARCHAR(10) null,

city VARCHAR(10) null,

county VARCHAR(10) null,

town VARCHAR(10) null,

detail\_add VARCHAR(256) null,

is\_default BOOL null default true,

constraint PK\_USER\_ADD primary key (add\_id)

);

/\*==============================================================\*/

/\* Index: user\_add\_PK \*/

/\*==============================================================\*/

create unique index user\_add\_PK on user\_add (

add\_id

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_add\_FK \*/

/\*==============================================================\*/

create index user\_basic\_info\_add\_FK on user\_add (

user\_id

);

/\*==============================================================\*/

/\* Table: user\_basic\_info \*/

/\*==============================================================\*/

create table user\_basic\_info (

user\_id VARCHAR(10) not null,

level\_id VARCHAR(10) null,

store\_id VARCHAR(10) null,

user\_name VARCHAR(20) not null,

user\_pwd VARCHAR(20) not null,

user\_former\_pwd VARCHAR(20) null,

register\_time DATE null,

user\_type CHAR(2) not null

constraint CKC\_USER\_TYPE\_USER\_BAS check (user\_type in ('1','2','3')),

is\_bind\_email BOOL null,

is\_bind\_qq BOOL null,

user\_phone VARCHAR(15) null,

register\_status BOOL null,

constraint PK\_USER\_BASIC\_INFO primary key (user\_id)

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_PK \*/

/\*==============================================================\*/

create unique index user\_basic\_info\_PK on user\_basic\_info (

user\_id

);

/\*==============================================================\*/

/\* Index: upgrade\_FK \*/

/\*==============================================================\*/

create index upgrade\_FK on user\_basic\_info (

level\_id

);

/\*==============================================================\*/

/\* Index: user\_store\_FK \*/

/\*==============================================================\*/

create index user\_store\_FK on user\_basic\_info (

store\_id

);

/\*==============================================================\*/

/\* Table: user\_basic\_info\_coupon \*/

/\*==============================================================\*/

create table user\_basic\_info\_coupon (

coupon\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

get\_time DATE null,

coupon\_num INT4 null,

constraint PK\_USER\_BASIC\_INFO\_COUPON primary key (coupon\_id, user\_id)

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_coupon\_PK \*/

/\*==============================================================\*/

create unique index user\_basic\_info\_coupon\_PK on user\_basic\_info\_coupon (

coupon\_id,

user\_id

);

/\*==============================================================\*/

/\* Index: user\_have\_coupon\_FK \*/

/\*==============================================================\*/

create index user\_have\_coupon\_FK on user\_basic\_info\_coupon (

user\_id

);

/\*==============================================================\*/

/\* Index: coupon\_is\_have\_FK \*/

/\*==============================================================\*/

create index coupon\_is\_have\_FK on user\_basic\_info\_coupon (

coupon\_id

);

/\*==============================================================\*/

/\* Table: user\_detail\_info \*/

/\*==============================================================\*/

create table user\_detail\_info (

user\_id VARCHAR(10) not null,

user\_email VARCHAR(20) null,

user\_qq VARCHAR(15) null,

user\_pay\_pwd CHAR(6) not null,

user\_real\_name VARCHAR(20) not null,

user\_sex CHAR(6) null default '男'

constraint CKC\_USER\_SEX\_USER\_DET check (user\_sex is null or (user\_sex in ('男','女'))),

user\_certificate\_type CHAR(2) not null default '1'

constraint CKC\_USER\_CERTIFICATE\_\_USER\_DET check (user\_certificate\_type in ('1','2','3')),

user\_certificate VARCHAR(20) not null,

user\_birth DATE null,

user\_point INT4 null default 0

constraint CKC\_USER\_POINT\_USER\_DET check (user\_point is null or (user\_point >= 0)),

constraint PK\_USER\_DETAIL\_INFO primary key (user\_id)

);

/\*==============================================================\*/

/\* Index: user\_detail\_info\_PK \*/

/\*==============================================================\*/

create unique index user\_detail\_info\_PK on user\_detail\_info (

user\_id

);

/\*==============================================================\*/

/\* Table: user\_favorite \*/

/\*==============================================================\*/

create table user\_favorite (

favorite\_id VARCHAR(10) not null,

user\_id VARCHAR(10) not null,

id\_type CHAR(2) not null default '1'

constraint CKC\_ID\_TYPE\_USER\_FAV check (id\_type in ('1','2')),

status BOOL not null default false,

favorite\_goods\_id VARCHAR(10) not null,

favorite\_add\_time DATE null,

constraint PK\_USER\_FAVORITE primary key (favorite\_id)

);

/\*==============================================================\*/

/\* Index: user\_favorite\_PK \*/

/\*==============================================================\*/

create unique index user\_favorite\_PK on user\_favorite (

favorite\_id

);

/\*==============================================================\*/

/\* Index: user\_basic\_info\_favorite\_FK \*/

/\*==============================================================\*/

create index user\_basic\_info\_favorite\_FK on user\_favorite (

user\_id

);

/\*==============================================================\*/

/\* Table: user\_rank \*/

/\*==============================================================\*/

create table user\_rank (

level\_id VARCHAR(10) not null,

level\_name VARCHAR(20) not null default '普通会员'

constraint CKC\_LEVEL\_NAME\_USER\_RAN check (level\_name in ('普通会员','青铜会员','白银会员','黄金会员','钻石会员')),

min\_point INT4 not null,

max\_point INT4 not null,

constraint PK\_USER\_RANK primary key (level\_id)

);

/\*==============================================================\*/

/\* Index: user\_rank\_PK \*/

/\*==============================================================\*/

create unique index user\_rank\_PK on user\_rank (

level\_id

);

alter table SPU

add constraint FK\_SPU\_SPU\_BRAND\_SPU\_BRAN foreign key (brand\_id)

references SPU\_brand (brand\_id)

on delete restrict on update restrict;

alter table SPU

add constraint FK\_SPU\_SPU\_CATEG\_SPU\_CATE foreign key (category\_id)

references SPU\_category (category\_id)

on delete restrict on update restrict;

alter table SPU

add constraint FK\_SPU\_SPU\_TYPE\_\_SPU\_TYPE foreign key (type\_id)

references SPU\_type (type\_id)

on delete restrict on update restrict;

alter table SPU

add constraint FK\_SPU\_STORE\_SPU\_STORE foreign key (store\_id)

references store (store\_id)

on delete restrict on update restrict;

alter table SPU\_brand

add constraint FK\_SPU\_BRAN\_SPU\_TYPE\_\_SPU\_TYPE foreign key (type\_id)

references SPU\_type (type\_id)

on delete restrict on update restrict;

alter table SPU\_category

add constraint FK\_SPU\_CATE\_SPU\_TYPE\_\_SPU\_TYPE foreign key (type\_id)

references SPU\_type (type\_id)

on delete restrict on update restrict;

alter table SPU\_spec

add constraint FK\_SPU\_SPEC\_SPU\_TYPE\_\_SPU\_TYPE foreign key (type\_id)

references SPU\_type (type\_id)

on delete restrict on update restrict;

alter table SPU\_spec\_value

add constraint FK\_SPU\_SPEC\_SPU\_SPEC\_\_SPU\_SPEC foreign key (spe\_id)

references SPU\_spec (spe\_id)

on delete restrict on update restrict;

alter table comment\_goods

add constraint FK\_COMMENT\_\_ORDER\_MAI\_ORDER\_MA foreign key (order\_id)

references order\_main\_info (order\_id)

on delete restrict on update restrict;

alter table comment\_goods

add constraint FK\_COMMENT\_\_USER\_BASI\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table delivery\_info

add constraint FK\_DELIVERY\_DELIVERY\_\_DELIVERY foreign key (company\_id)

references delivery\_company (company\_id)

on delete restrict on update restrict;

alter table delivery\_info

add constraint FK\_DELIVERY\_ORDER\_MAI\_ORDER\_MA foreign key (order\_id)

references order\_main\_info (order\_id)

on delete restrict on update restrict;

alter table delivery\_info

add constraint FK\_DELIVERY\_USER\_ADD\_\_USER\_ADD foreign key (add\_id)

references user\_add (add\_id)

on delete restrict on update restrict;

alter table discending\_SPU

add constraint FK\_DISCENDI\_ACTIVITY\_\_MALL\_ACT foreign key (activity\_id)

references mall\_activity (activity\_id)

on delete restrict on update restrict;

alter table discending\_SPU

add constraint FK\_DISCENDI\_DISCENDIN\_SPU foreign key (spu\_id)

references SPU (spu\_id)

on delete restrict on update restrict;

alter table distributor

add constraint FK\_DISTRIBU\_DELIVERY\_\_DELIVERY foreign key (company\_id)

references delivery\_company (company\_id)

on delete restrict on update restrict;

alter table full\_reduction\_SPU

add constraint FK\_FULL\_RED\_ACTIVITY\_\_MALL\_ACT foreign key (activity\_id)

references mall\_activity (activity\_id)

on delete restrict on update restrict;

alter table full\_reduction\_SPU

add constraint FK\_FULL\_RED\_FULL\_REDU\_SPU foreign key (spu\_id)

references SPU (spu\_id)

on delete restrict on update restrict;

alter table goods

add constraint FK\_GOODS\_SPU\_SKU\_SPU foreign key (spu\_id)

references SPU (spu\_id)

on delete restrict on update restrict;

alter table mall\_activity

add constraint FK\_MALL\_ACT\_ACTIVITY\_\_STORE foreign key (store\_id)

references store (store\_id)

on delete restrict on update restrict;

alter table mall\_activity

add constraint FK\_MALL\_ACT\_ADMINISTR\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table mall\_ad

add constraint FK\_MALL\_AD\_ADMINISTR\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table order\_goods

add constraint FK\_ORDER\_GO\_SKU\_ORDER\_GOODS foreign key (goods\_id)

references goods (goods\_id)

on delete restrict on update restrict;

alter table order\_goods

add constraint FK\_ORDER\_GO\_ORDER\_GOO\_ORDER\_MA foreign key (order\_id)

references order\_main\_info (order\_id)

on delete restrict on update restrict;

alter table order\_main\_info

add constraint FK\_ORDER\_MA\_STORE\_ORD\_STORE foreign key (store\_id)

references store (store\_id)

on delete restrict on update restrict;

alter table order\_main\_info

add constraint FK\_ORDER\_MA\_USER\_BASI\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table return\_goods

add constraint FK\_RETURN\_G\_BUSINESS\_\_USER\_ADD foreign key (business\_add\_id)

references user\_add (add\_id)

on delete restrict on update restrict;

alter table return\_goods

add constraint FK\_RETURN\_G\_BUYER\_ADD\_USER\_ADD foreign key (buyer\_add\_id)

references user\_add (add\_id)

on delete restrict on update restrict;

alter table return\_goods

add constraint FK\_RETURN\_G\_DELIVERY\_\_DELIVERY foreign key (company\_id)

references delivery\_company (company\_id)

on delete restrict on update restrict;

alter table return\_goods

add constraint FK\_RETURN\_G\_USER\_BASI\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table server

add constraint FK\_SERVER\_STORE\_SER\_STORE foreign key (store\_id)

references store (store\_id)

on delete restrict on update restrict;

alter table shopping\_cart

add constraint FK\_SHOPPING\_SKU\_SHOPP\_GOODS foreign key (goods\_id)

references goods (goods\_id)

on delete restrict on update restrict;

alter table shopping\_cart

add constraint FK\_SHOPPING\_USER\_BASI\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table store

add constraint FK\_STORE\_ADVER\_STO\_MALL\_AD foreign key (adver\_id)

references mall\_ad (adver\_id)

on delete restrict on update restrict;

alter table store

add constraint FK\_STORE\_USER\_STOR\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table user\_add

add constraint FK\_USER\_ADD\_USER\_BASI\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table user\_basic\_info

add constraint FK\_USER\_BAS\_UPGRADE\_USER\_RAN foreign key (level\_id)

references user\_rank (level\_id)

on delete restrict on update restrict;

alter table user\_basic\_info

add constraint FK\_USER\_BAS\_USER\_STOR\_STORE foreign key (store\_id)

references store (store\_id)

on delete restrict on update restrict;

alter table user\_basic\_info\_coupon

add constraint FK\_USER\_BAS\_COUPON\_IS\_COUPON foreign key (coupon\_id)

references coupon (coupon\_id)

on delete restrict on update restrict;

alter table user\_basic\_info\_coupon

add constraint FK\_USER\_BAS\_USER\_HAVE\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table user\_detail\_info

add constraint FK\_USER\_DET\_USER\_BASI\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;

alter table user\_favorite

add constraint FK\_USER\_FAV\_USER\_BASI\_USER\_BAS foreign key (user\_id)

references user\_basic\_info (user\_id)

on delete restrict on update restrict;