

# 电子商务网站数据库系统实践

段敏萱

(北京大学元培学院, 北京 100871);

E-mail: dminxuan@gmail.com

阮思凯

(北京大学信息科学技术学院, 北京 100871);

E-mail: 同学的邮件

**摘要:** 描述本课程设计的主要工作, 特色, 完成这个项目的收获

## 1 引言

随着互联网的迅速发展, 电子商务与网络购物正逐步走向平民化。调查机构 AC 尼尔森最新研究显示, 有 63% 的中国网民有过通过互联网购物的经历。2019 年双十一购物节, 仅天猫一家的订单峰值也达到 54.4 万笔/秒, 如此高强度的交易处理量对电子商务网站的数据库系统建设提出了新的挑战。

在这个课程设计中, 我们使用 Microsoft SQL Server / pymssql 实现了

本项目处理现场数据建模过程。我们使用 homedepot.com 并创建了一个示例数据库来管理商店操作。在前期设计的部分概念模型的基础上, 以网站产品和部门为参考, 对数据库中的对象进行适当的属性、约束和关系的设计。

此外, 我们还编写了一些 DCL 语句来管理数据库的安全性。

我们使用复杂的 SQL 查询执行一些存储事务和管理分析。

描述为什么选择这个题目, 其意义是什么

描述一下课程设计实现的功能

个人的收获

本文的组织: 在第二部分中, 我们介绍了本课程设计需要的背景知识, 包括。。。; 第三部分描述了课程设计的系统框架; 第四部分给出相关的关系数据库模式和相应的操作; 第五部分总结了全文, 并且描述了完成本课程设计的收获。

## 2 相关工作

以淘宝网的交易系统为例, 我们研究了电商平台的相关技术: 淘宝网交易系统其由各种不同功能的模块组成, 其中有负责存储大量小文件的淘宝文件系统 (TFS), 负责对高频使用的数据进行内存缓存功能的淘宝 KV 缓存系统 (Tair), 对图像进行分发以降低网络流量的内容分发网络 (CDN), 负责内部通讯调度的高性能服务框架 (HSF), 负责存储海量数据的分布式数据库 (OceanBase), 负责负载均衡的 LVS 服务器, 负责可靠信息传递的消息中间件系统, 负责网站内容展示的网站应用框架 (WEBX)。另外还有高性能低消耗的基于 X86 架构的服务器。在这些技术中, OceanBase 分布式数据库是技术集大成者, 最关键和最成功的系统之一。

介绍一下完成这个项目的关键技术, 不用太多

### 3 课程设计的系统框架

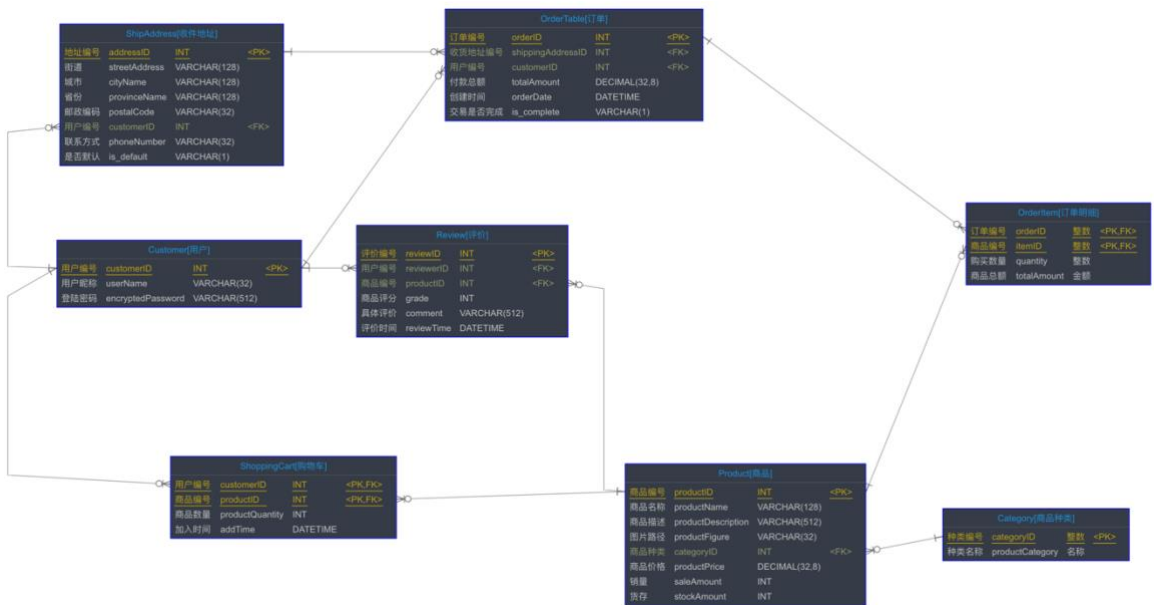
利用图描述整个系统的框架，可以看作是需求分析的进一步描述  
同时利用文字给出粗略给出每个模块的功能介绍（可能不止一个图）  
如画图建议使用 Visio

### 4 课程设计对应的关系数据库模式

#### 4.1 ER图

概念模式设计，并给出一定的解释，这个如果用 powerdesigner，可以直接从 powerdesign 拷贝那个设计图

#### 4.2 关系数据库模式描述



上图描述了关系数据库的物理模式设计，表清单具体如下：

名称	解释
Customer	用户
Product	商品
OrderTable	订单
ShoppingCart	购物车
ShipAddress	收件地址
Category	商品种类
Review	评价

OrderItem	订单明细
-----------	------

#### 4.2.1 Customer 【用户】

代码	名称	数据类型	备注
customerID	用户编号	INT	主键，自增
userName	用户昵称	VARCHAR(32)	唯一值
encryptedPassword	登陆密码	VARCHAR(512)	hash 加密

Customer 表存储了用户的登录信息，主键是 customerID。每个用户在注册后自增地生成一个 customerID，并向 Customer 表插入其用户昵称 userName 和登录密码 encryptedPassword。昵称不可以重复，因此为主属性。为了防止黑客盗取信息，我们并不是明文存储密码，而是通过 hashlib 库对密码进行 hash 加密。之后每次用户登录的时候，从数据库取出已经加密的密码，并将用户输入经过 hash 后的字符串进行对比。如果 hash 值相同，用户获得登入授权，否则，会被告知输入了无效的登录信息。

Customer 关系中每一分量都不可再分；每个非主属性完全依赖于码，且不存在对码的传递依赖；因为主属性只有一个，因此不存在主属性之间的部分和传递函数依赖；不存在不包含码的非平凡多值依赖。因此该表的设计达到了 4NF 的要求。

#### 4.2.2 Product 【商品】

代码	名称	数据类型	备注
productID	商品编号	INT	主键，自增
productName	商品名称	VARCHAR(128)	
productDescription	商品描述	VARCHAR(512)	可以为空
productFigure	图片路径	VARCHAR(32)	可以为空
categoryID	商品种类	INT	外键(Category)
productPrice	商品价格	DECIMAL(32,8)	
saleAmount	销量	INT	
stockAmount	货存	INT	

Product 表存储了所有在架商品的信息，主键是 productID。productID 在实际应用中可以选用商品二维码来表示，但在这里为了方便管理，我们选用系统内部自增编号来表示。Product 表存储了商品的名称、图文描述、分类、价格、销量及货存，其中关于商品的属性（如颜色、尺码、规格等）都统一记录在商品描述字段中，每个商品仅配一张图片，图片路径格式统一为 static/images/id.jpg。

Product 关系中每一分量在使用时都不可再分；且不存在主属性之间的部分和传递函数依赖；不存在不包含码的非平凡多值依赖。因此该表的设计达到了 4NF 的要求。

#### 4.2.3 OrderTable 【订单】

代码	名称	数据类型	备注
orderID	订单编号	INT	主键，自增
shippingAddressID	收货地址编号	INT	外键(ShipAddress)
customerID	用户编号	INT	外键(Customer)
totalAmount	付款总额	DECIMAL(32,8)	
orderDate	创建时间	DATETIME	
is_complete	交易是否完成	VARCHAR(1)	1=完成，0=取消，NULL=进行中

OrderTable 表存储了订单的基本信息，主键是 orderID。用户下单后，在 OrderTable 表中插入一条新的记录，包含了用户之前填写并存储在 ShipAddress 表中的收货地址、用户编号、创建时间，并计算订单内全部商品的价格总额。记录中的 is\_complete 字段的 default 值为 NULL，表示交易进行中，订单完成后更改为 1，若交易被取消则改为 0。订单中的商品明细是订单的多值属性，我们将单独建立 OrderItem 表。

类似前文的讨论，OrderTable 关系的设计达到了 4NF 的要求。

#### 4.2.4 ShoppingCart【购物车】

代码	名称	数据类型	备注
customerID	用户编号	INT	主键，外键
productID	商品编号	INT	主键，外键
productQuantity	商品数量	INT	至少为 1
addTime	加入时间	DATETIME	

ShoppingCart 表存储了购物车的明细，主键是 customer ID 和 productID，即每个用户对每个商品想要购买但尚未下单的操作在购物车里对应一条记录，这条记录还包含了想要购买的商品数量和加入购物车的时间。

类似前文的讨论，ShoppingCart 关系的设计达到了 4NF 的要求。

#### 4.2.5 ShipAddress【收件地址】

代码	名称	数据类型	备注
addressID	地址编号	INT	主键，自增
streetAddress	街道	VARCHAR(128)	
cityName	城市	VARCHAR(128)	
provinceName	省份	VARCHAR(128)	
postalCode	邮政编码	VARCHAR(32)	可以为空

customerID	用户编号	INT	外键(Customer)
phoneNumber	联系方式	VARCHAR(32)	
is_default	是否默认	VARCHAR(1)	1=是, 0=否

ShipAddress 表存储了用户添加的收件地址，主键是 address ID。每个用户可以创建多个地址，并且可以将其中一个设置为默认地址，default 值为否。用户每次填写订单时，页面会自动填好默认地址。每个用户最多有一个默认地址。每个记录里还存储了街道、城市、省份、联系方式等必要信息，邮政编码可以为空。

ShipAddress 关系中每一分量在使用时都不可再分，且每个非主属性完全依赖于码。虽然好像存在城市决定邮政编码这一非主属性对码的传递依赖，但是由于设置了邮政编码可以为 NULL 值，因此该表的设计还是达到了 4NF 的要求。

#### 4.2.6 Category 【商品种类】

代码	名称	数据类型	备注
categoryID	种类编号	INT	主键，自增
productCategory	种类名称	VARCHAR(128)	

Category 表中存储了所有的商品种类。Category 表格足够简单，关系中每一分量在使用时都不可再分，也不存在不包含码的非平凡多值依赖，因此该表的设计达到了 4NF 的要求。

#### 4.2.7 Review 【评价】

代码	名称	数据类型	备注
reviewID	评价编号	INT	主键，自增
reviewerID	用户编号	INT	外键(Customer)
productID	商品编号	INT	外键(Product)
grade	商品评分	INT	1-5 的整数
comment	具体评价	VARCHAR(512)	可以为空
reviewTime	评价时间	DATETIME	

Review 表中存储了用户在购买完成后对商品的评价，主键是 reviewID。每条记录里存储了对应的用户和商品编号、用户对商品的评分（1-5 的整数颗星）以及具体评价内容（可以为空）和评价时间。

类似前文的讨论，Review 的设计达到了 4NF 的要求。

#### 4.2.8 OrderItem 【订单明细】

代码	名称	数据类型	备注
orderID	订单编号	INT	主键，外键
itemID	商品编号	INT	主键，外键

quantity	购买数量	INT	至少为 1
totalAmount	商品总额	DECIMAL(32,8)	

OrderItem 表中存储了订单中的每件商品、购买数量并计算其商品总额，主键是 orderID 和 itemID。

OrderItem 关系中每一分量在使用时都不可再分，且不存在主属性之间的部分和传递函数依赖。这里似乎存在商品编号和购买数量决定了商品总额，但需要注意的是商品价格的浮动不会影响历史订单的商品价格，因此该决定关系并不成立，OrderItem 表的设计依旧达到了 4NF 的要求。

还包括视图设计、索引设计，说明是否采用了采用 Stored procedure, Trigger 等操作，

#### 4.3 关系数据库模式之上的操作

针对我们在第三部分中提到的各种功能要求，利用关系数据库的 SQL 表达相关的核心实现

### 5 总结

项目总结和个人收获

#### References:

- [1] Ashish Kumar Gupta, Dan Suciu: Stream Processing of XPath Queries with Predicates. In: Alon Y. Halevy, Zachary G. Ives, AnHai Doan., eds Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, ACM, 2003, 419-430

字数要求在 3000 字以上