**数据库设计文档**

# 1背景

## 1.1应用背景

为了本小组所制作的网上书城管理系统而设计的此数据库，目的是能够完成相应的后台功能，对管理员、用户及图书的信息进行管理维护，制定一套合理、有效，规范和实用的网上书城数据库系统，对图书资料进行集中统一的管理。

## 1.2技术背景

SQL Sever2016数据库

美国Microsoft公司推出的一种关系型数据库系统。SQL Server是一个可扩展的、高性能的、为分布式客户机/服务器计算所设计的[数据库管理系统](http://baike.baidu.com/view/68446.htm)，实现了与WindowsNT的有机结合，提供了基于[事务](http://baike.baidu.com/view/121511.htm)的企业级信息管理系统方案。

通过内置的突破式内存驻留技术，SQL server 2014 能为要求最高的数据库应用提供关键业务所需性能内存驻留技术，性能最高提升30倍。软件及硬件合作伙伴使用多样化的工作负载进行了全新的性能测试，证明了采用开创性内存计算技术的 SQL Server 2014 可以为那些对数据库有极高要求的应用程序提供符合需求的数据平台。注意：SQL server 2014产品只有olp开放式批量授权方式。olp批量许可是一个软件许可计划，由用户单独订阅，代理将用户信息上报给微软厂家，用户会收到厂家直发的邮件，再按照邮件指示的下载和安装等步骤进行操作即可。通常卖给这些企业客户，捆绑有五个或更多的数量的licenses。

## 1.3工具

本项目在SQL Sever2014中存储数据，利用IDEA开发平台实现项目的开发

# 2.概念结构设计

## 2.1E-R模型设计

1. 管理员实体

* 实体属性

管理员：{账号，密码，姓名，等级

* 实体ER图如图2-1



图2-1

1. 用户实体

* 实体属性

读者用户{账号，密码，姓名，性别，证件号码，电话号码，电子邮件，真实姓名，住址，登录次数，上次登录时间

* 实体ER图如图2-2



图2-2

1. 图书类型实体

* 实体属性

图书：图书类型ID,图书类型名

* 实体ER图如图2-3



图2-3

1. 图书信息实体

* 实体属性

图书：图书类型ID,图书类型名，图书名，图书ID，出版社，版本，作者，ISBN，页数，URL，出版日期，数量，价格

* 实体ER图如图2-4



图2-4

1. 图书购物车实体

* 实体属性

图书：读者ID，图书ID，订购数量

* 实体ER图如图2-5



图2-5

1. 图书订购实体

* 实体属性

图书：读者ID，订购ID，订购数量，邮寄方法，收获人，总价，订购状况，订购日期，完成时间

* 实体ER图如图2-6



图2-6

1. 图书详情实体

* 实体属性

图书：订购ID，图书ID，订购数量，完成时间，详情，订购详情ID

* 实体ER图如图2-7



图2-7

## 2.2总体E-R图



图2-8

# 3.逻辑结构设计

## 3.1命名规则

（1）字符集合

a-z, A-Z, 0-9和\_共63个

（2）所有字条以小写字母开始，所有名词采用单数，所以前缀都要小写如：uid

（3）数据库对象命名规范

数据库对象={表,试图,索引,关联,存储过程（参数查询），函数} 规则:对象名字由前缀和实际名字组成,他们之间用下划线连接,不要再对象名地字符之间留空格,长度不超过30个字符。

【对象名字】=【前缀】\_【实际名字】 前缀：使用表名的首字母 表 tb 试图 vi 索引 Idx 关联 rl 存储过程 sp 函数

fn

实际名字：实际名字尽量描述实体的内容，由单词或单词组合，每个单词均小写，不已数字和\_开头，单词和单词间不用下划线。

【实际名字】=【小写字母开头的单词】【大写字母开头的单词】···【大写字母开头的单词】

例子：tb\_user （4）数据库表命名规范

表名由前缀(tb)接\_接实际名字组成 【表名】=tb\_【实际名字】

## 3.2　关系模式优化

通过对用户需求和实体之间的关系的分析，结合范式考虑，本系统建立以下表

（1）Customer表

表3-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键或外键 | 是否为空 | 说明 |
| CustomerID | int | 主键 | 否 | 用户编号 |
| CustomerName | varchar(20) |  | 否 | 用户姓名 |
| CustomerPwd | varchar(64) |  | 否 | 用户密码 |
| CustomerTrueName | varchar(20) |  | 否 | 用户真实姓名 |
| CustomerSex | varchar(2) |  | 否 | 用户性别 |
| CustomerTel | varchar(20) |  | 否 | 用户电话 |
| CustomerEmail | varchar(20) |  | 否 | 用户邮箱 |
| CustomerAddr | varchar(20) |  | 否 | 用户地址 |
| CustomerLogTimes | Int |  | 否 | 用户登录次数 |
| CustomerLastLogTime | datetime |  | 否 | 上次登录时间 |

Check约束：

check (CustomerSex in ('男','女'))

（2）Admin表

表3-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键或外键 | 是否为空 | 说明 |
| AdminID | int | 主键 | 否 | 管理员ID |
| AdminName | varchar(20) |  | 否 | 管理员姓名 |
| AdminPwd | varchar(64) |  | 否 | 管理员密码 |
| AdminLevel | int |  | 否 | 管理员等级 |

（3）BookType表

表3-3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键或外键 | 是否为空 | 说明 |
| BookTypeID | int | 主键 | 否 | 图书类型ID |
| BookTypeName | varchar(20) |  | 否 | 图书类型名 |

（4）BookInfo表

表3-4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键或外键 | 是否为空 | 说明 |
| BookID | int | 主键 | 否 | 图书ID |
| BookName | varchar(20) |  | 否 | 图书名 |
| BookTypeID | int | 外键 | 否 | 图书类型ID |
| BookTypeName | varchar(20) |  | 否 | 图书类型名 |
| BookPubHouse | varchar(20) |  | 否 | 图书出版社 |
| BookPubDate | datetime |  | 否 | 图书出版日期 |
| BookVersion | varchar(10) |  | 否 | 图书版本 |
| BookAuthor | varchar(10) |  | 否 | 图书作者 |
| BookISBN | varchar(20) |  | 否 | 图书ISBN编号 |
| BookPrice | money |  | 否 | 图书价格 |
| BookPages | Int |  | 否 | 图书页数 |
| BookUrl | varchar(50) |  | 否 | 图书URL |
| BookOutline | varchar(200) |  | 否 | 图书摘要 |
| BookAmount | int |  | 否 | 图书数量 |

（5）BookCART表

表3-5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键或外键 | 是否为空 | 说明 |
| CustomerID | int | 主键、外键 | 否 | 用户ID |
| BookID | int | 主键、外键 | 否 | 图书ID |
| OrderAmount | int |  | 否 | 订购数量 |

（6）Orders表

表3-6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键或外键 | 是否为空 | 说明 |
| OrderID | int | 主键 | 否 | 订购ID |
| CustomerID | int | 外键 | 否 | 用户ID |
| OrderAmount | int |  | 否 | 订购数量 |
| PostMethod | varchar(100) |  | 否 | 邮寄方法 |
| ReceiverName | varchar(10) |  | 否 | 收货人 |
| TotalPrice | money |  | 否 | 总价 |
| OrderStatus | varchar(10) |  | 否 | 订购状态 |
| OrderDate | datetime |  | 否 | 订购日期 |
| FinishDate | datetime |  | 是 | 完成时间 |

Check约束：

check(OrderStatus in ('已完成','已取消','进行中'))

（7）OrderDetail表

表3-7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键或外键 | 是否为空 | 说明 |
| OrderDetailID | int | 主键 | 否 | 订购详细ID |
| OrderID | int | 外键 | 否 | 订购ID |
| BookID | int | 外键 | 否 | 图书ID |
| OrderAmount | int |  | 否 | 订购数量 |
| OrderDetailStatus | varchar(10) |  | 否 | 订购详细状况 |
| FinishDate | datetime |  | 否 | 完成时间 |

Check约束：

check(OrderDetailStatus in ('未发货','已发货','已收货','已取消'))

# 4.物理结构设计

## 4.1物理设计

实现该设计的环境为Windows 10+ MS SQL Server2014 或以上版本。

（1）用户表建立

create table Customer(

CustomerID int identity,

CustomerName varchar(20) unique,

CustomerPwd varchar(64) not null,

CustomerTrueName varchar(20) not null,

CustomerSex varchar(2) not null,

CustomerTel varchar(20) not null,

CustomerEmail varchar(20) not null,

CustomerAddr varchar(20) not null,

CustomerLogTimes int not null,

CustomerLastLogTime datetime not null,

primary key(CustomerId),

check (CustomerSex in ('男','女'))

)

（2）管理员表建立  
create table Admin(

AdminID int identity,

AdminName varchar(20) not null,

AdminPwd varchar(64) not null,

AdminLevel int not null,

primary key (AdminID)

)

（3）图书类型表建立  
create table BookType(

BookTypeID int identity,

BookTypeName varchar(20) not null,

primary KEY (BookTypeID)

)

（4）图书信息表建立  
create table BookInfo(

BookID int identity,

BookName varchar(30) not null,

BookTypeID int not null,

BookTypeName varchar(20) not null,

BookPubHouse varchar(20) not null,

BookPubDate datetime not null,

BookVersion varchar(10) not null,

BookAuthor varchar(20) not null,

BookISBN varchar(20) not null,

BookPrice money not null,

BookPages int not null,

BookUrl varchar(50) null,

BookOutline varchar(200) not null,

BookAmount int not null,

check(BookAmount>=0),

primary key (BookId),

foreign key (BookTypeID) references BookType(BookTypeID)

)  

（5）图书CART表建立  
create table BookCART(

CustomerID int not null,

BookID int not null,

OrderAmount int not null,

primary key (CustomerID,BookID),

foreign key (CustomerID) references Customer(CustomerID),

foreign key (BookID) references BookInfo(BookID),

)

（6）订购表建立  
create table Orders(

OrderID int identity,

CustomerID int not null,

OrderAmount int not null,

PostMethod varchar(100) not null,

ReceiverName varchar(10) not null,

TotalPrice money not null,

OrderStatus varchar(10) not null,

OrderDate datetime not null,

FinishDate datetime null,

check(OrderStatus in ('已完成','已取消','进行中')),

primary key (OrderId),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

)

（7）订购细节表建立  
create table OrderDetail(

OrderDetailID int identity,

OrderID int not null,

BookID int not null,

OrderAmount int not null,

OrderDetailStatus varchar(10) not null,

FinishDate datetime null,

check(OrderDetailStatus in ('未发货','已发货','已收货','已取消','未付款')),

primary key (OrderDetailID),

foreign key (OrderID) references Orders(OrderID),

foreign key (BookID) references BookInfo(BookID),

)

# 5.触发器设计

（1）在购物车BookCART里面使用了一个触发器，用于向购物车添加商品时判断是否存在

CREATE TRIGGER [dbo].[BookCART\_OrderPrice]  
   ON  [dbo].[BookCART]  
   INSTEAD OF INSERT  
AS   
BEGIN  
declare @BookPrice money  
declare @OrderAmount int  
declare @BookID int  
declare @CustomerID int  
select @BookID = BookID,@CustomerID=CustomerID,@OrderAmount=OrderAmount from inserted  
if (select COUNT(\*) from BookCART where BookID=@BookID and CustomerID=@CustomerID)=0  
BEGIN  
insert into BookCART values(@CustomerID,@BookID,@OrderAmount)  
END  
else  
BEGIN  
update BookCART set OrderAmount=OrderAmount+@OrderAmount where BookID=@BookID and CustomerID=@CustomerID  
END  
END