



# **数据库课程设计**

课程名称 数据库系统

题目名称 图书管理系统

专业班级 15级网络工程4班

学 号 3115005534

学生姓名 李楚豪

指导教师 周俭铁

2018 年 1 月

**目录**

[**数据库课程设计** 1](#_Toc503274958)

[**一、** **相关技术介绍** 1](#_Toc503274959)

[**二、** **系统需求分析** 1](#_Toc503274960)

[a) 业务流程 1](#_Toc503274961)

[b) 功能需求 1](#_Toc503274962)

[c) 信息需求 1](#_Toc503274963)

[d) 安全性与完整性需求 1](#_Toc503274964)

[e) 数据字典 2](#_Toc503274965)

[f) 数据流图 3](#_Toc503274966)

[**三、** **概念模型设计** 4](#_Toc503274967)

[a) 分E-R图 4](#_Toc503274968)

[b) 总E-R图 6](#_Toc503274969)

[**四、** **逻辑结构设计** 7](#_Toc503274970)

[a) E-R图向关系模型的转换： 7](#_Toc503274971)

[b) 数据模型的优化 11](#_Toc503274972)

[c) 设计用户子模式 12](#_Toc503274973)

[**五、** **物理结构设计** 13](#_Toc503274974)

[a) 建立索引 13](#_Toc503274975)

[b) 数据存放 14](#_Toc503274976)

[c) 系统配置 14](#_Toc503274977)

[**六、** **数据库连接与应用** 14](#_Toc503274978)

[a) VC++中使用MFC通过ADO连接数据库 14](#_Toc503274979)

[b) 数据库的运行与维护 15](#_Toc503274980)

[**七、** **收获与体会** 21](#_Toc503274981)

1. **相关技术介绍**

开发环境：Windows 8.1

数据库管理系统：MYSQL

开发工具：Eclipse Java

运行环境：Windows xp以及更高版本

系统结构：C/S

1. **系统需求分析**
   1. 业务流程

用户在本系统申请账号后便可用申请到的账号登陆到本系统，在本系统中可以将一些用户其它系统的账号的相关信息（包括认证信息，基本信息）存储并根据需要进行共享。

* 1. 功能需求

1. 用户在本系统中填写相关信息申请账号；
2. 用户用申请得到的账号来完成身份验证登陆到本系统；
3. 用户可以把自己的一些账户信息存储在本系统中而受到保密；
4. 用户可对存储的账户进行条件查询，已获得相应结果；
5. 用户可共享自己的账号给本系统的其它用户，获得相应的共享值；
6. 用户可用自身的共享值来换取其他用户所共享的账户；
   1. 信息需求
7. 各大登陆系统平台的账号特点，种类；
8. 用户拥有账号的大概分布情况；
9. 用户的登陆操作习惯。
   1. 安全性与完整性需求
10. 用户申请账号时不能跟系统中已有的账号重复；
11. 用户在登陆系统后只能查询得到自己存储的以及已共享获得的并还在共享期间中的账户信息；
12. 用户的可用共享值不能小于零；
13. 每个用户都有唯一标识；
14. 每个存储的账户信息都有唯一标识；
15. 用户不能共享已在共享期间中的账户；
16. 用户不可重复共享两个账号；

数据库上存储信息的安全性则由数据库管理系统进行保证，完整性则主要由编程语句来体现。

* 1. 数据字典

**表2.1 数据项-数据项是不可再分的数据单位**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名称 | 别名 | 类型 | 含义说明 |
| 账号编号 | Ano | Int | 存储账户的唯一标识码，系统自动生成 |
| 账号 | Aacco | Char(20) | 存储账号，可由20位不同的符号组成 |
| 密码 | Acode | Char(30) | 账户密码，可由30位不同的符号组成 |
| 所需共享值 | Ashar | Int | 共享使用该账号所需的共享值，不可共享则为NULL值 |
| 共享时长 | Atime | Int | 支付共享值后，所获得的账户共享时长，不可共享则为NULL值 |
| 邮箱地址 | Bemai | Char(30) | 账户的保密E-mail，可以是163,qq等 |
| 注册手机号 | Bphno | Char(11) | 账户注册时所绑定的手机号 |
| 注册证件号 | Bceno | Char(18) | 账户注册时所绑定的手机号 |
| 申请日期 | Badat | Date | 账户的申请日期 |
| 存储日期 | Bsdat | Date | 账户的存储日期 |
| 相关系统编号 | Rno | Int | 账户相关的系统的编号，系统自动生成 |
| 相关系统名字 | Rnam | Char(10) | 账户相关的系统的名字 |
| 相关系统官网 | Rsite | Char(20) | 账户相关的系统的官网地址 |
| 相关系统类型 | Rtype | Char(4) | 账户相关的系统的类型 |
| 相关系统描述 | Rdesr | Char(50) | 账户相关的系统的大概描述 |
| 系统用户账号 | Sacco | Char(20) | 此系统中用户的注册账号，唯一标识 |
| 系统用户密码 | Scode | Char(30) | 此系统中用户的账户密码 |
| 账户总共享值 | Stsha | Int | 此系统用户的总共享值，包括已使用的 |
| 账户可共享值 | Scsha | Int | 此系统用户的剩余可使用的共享值 |
| 所剩共享时长 | Cstim | Int | 本次共享的剩余时间 |
| 系统用户  手机号 | Uphno | Char(11) | 此系统账户绑定的手机号，由11位数字组成 |
| 系统用户  邮箱地址 | Uemai | Char(30) | 此系统账户绑定的邮箱地址，由30位字符组成 |
| 系统用户  证件号 | Uceno | Char(18) | 此系统账户绑定的证件号，由18位数字组成 |
| 系统用户姓名 | Uname | Char(8) | 此系统用户的姓名 |
| 系统用户年龄 | Uage | Int | 此系统用户的年龄，为正数 |
| 系统用户学历 | Ueduc | Char(4) | 此系统用户的学历，为小学，中学，大学中的一种 |

**表2.2数据结构-数据结构反映数据之间的组合关系**

|  |  |  |
| --- | --- | --- |
| 数据结构名称 | 含义说明 | 组成 |
| 系统账户 | 用户在本系统的账户信息 | 账号，密码，总共享值，可共享值 |
| 用户 | 用户自身的信息 | 姓名，年龄，学历，邮箱，证件号，手机号，系统账户 |
| 存储账户 | 用户在本系统存储的账户信息 | 编号，账号，密码，所需共享值，手机号，邮箱，证件号，相关系统信息 |
| 相关系统信息 | 用户存储账户的相关系统信息 | 编号，网站，类型，名称，描述 |
| 共享信息 | 用户共享账号的相关信息 | 存储账户编号，系统用户，  剩余共享时长 |

* 1. 数据流图



图2.1 0层流图

1. **概念模型设计**
   1. 分E-R图



图2.2 E-R分图



图2.3 E-R分图

* 1. 总E-R图

图2.4 E-R总图

1. **逻辑结构设计**
   1. E-R图向关系模型的转换：

系统账户（系统账号，系统密码，可用共享值，总共享值）

建立表如图4.1：

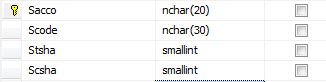


图4.1 系统帐户表

此为系统账户实体所对应的关系模式。

完整性：主键为“系统账号”；各属性均不可为空值。

用户（手机号，姓名，年龄，学历，证件号，用户邮箱地址，系统账号），建立表如图4.2：

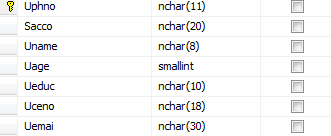


图4.2 用户表

此为用户实体所对应的关系模式，因为“注册”联系是1：1联系，所以将其与用户实体对应的关系模式合并，并加入“系统账户”关系模式中的码。

完整性：

主键为“手机号”；

外键为“系统账号”，并设置为级联更新和级联删除：

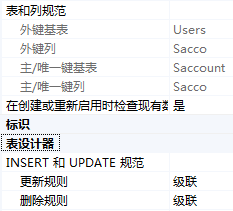


图4.3 系统帐户表

各属性均不可为空值。

账户相关系统（相关系统编号，系统名字，系统官网，系统类型，系统描述）

建立表如图4.4

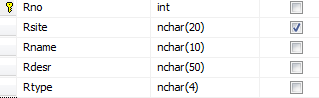
：

图4.4 帐户相关表

此为账户相关系统实体所对应的关系模式。

完整性：

主键为“系统编号”， 已被设置为自增长int类型：

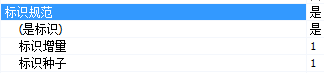


图4.5

除“系统官网”其它属性均不可为空值。

存储账户（账号编号，账号，密码，系统用户账号，注册手机号，注册证件号，邮箱地址，所需共享值，共享时长，存储日期，申请日期，相关系统编号）

（因进行优化，所以此关系模式中分为两张表，“数据模型的优化”将详细说明），建立表如图4.6

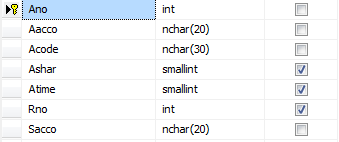




图4.6 存储帐户表

此为存储用户实体所对应的关系模式，因为“相关”联系是n：1联系，“存储账户”为n端，所以将其关系模式与“相关”联系对应的关系模式合并，并加入“账户相关系统”关系模式中的码；同理，因为“存储”联系是n：1联系，“存储账户”为n端，所以将其关系模式与”存储”联系对应的关系模式合并，并加入“系统账户”关系模式中的码

完整性：

主键为“账号编号”，已被设置为自增长int类型：

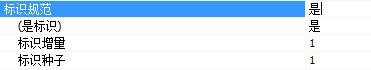


图4.7

外键为“相关系统编号”， 并设置为级联更新和级联删除：

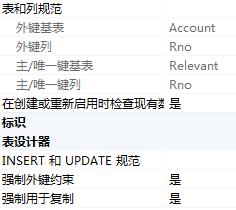


图4.8

外键为“系统用户账号”， 并设置为级联更新和级联删除：

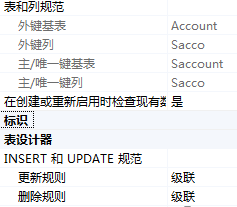


图4.9

除了属性“所需共享值”和“共享时长”可为空，其它属性值均不可为空。

账户共享（系统账号，账号编号，所剩共享时长），建立表如图4.10

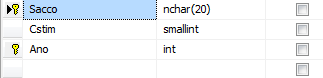


图4.10 帐户共享表

此为“共享”联系所对应的关系模式，由于联系为m：n关系，所以独立转换为一个关系模式，并加入“存储账户”与“系统用户”关系模式中各自的码。

完整性：

主键为“系统账号，账号编号”；

外键为“系统账号”和“账号编号”， 并设置为级联更新和级联删除：

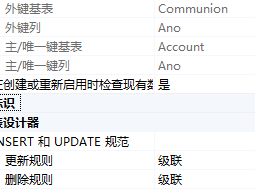


图4.11

所有属性均不能为空。

* 1. 数据模型的优化

确定各个关系中的数据依赖，对转换的关系模式逐一进行分析，不难发现以上各关系模式均属于BCNF范。

为提高数据操作的效率和存储空间的利用率，我们可根据实际使用情况对数据模式进行必要的分解。

关系模式 存储账户（账号编号…）中经常需要被查询使用的属性为（账号，密码，所需共享值，共享时长），而为保持无损连接性和保持函数依赖性，可对此关系模式进行垂直分解，分解为存储账户基本信息（账号编号，账号，密码，所需共享值，共享时长）与存储账户绑定信息（账号编号，申请日期，存储日期，注册证件号，注册手机号，邮箱地址）这两个子关系模式（分解后仍为BCNF），使其所有事物的总效率提高。

* 1. 设计用户子模式

为简化用户对本系统一般查询使用，将某些常用的较复杂的查询定义为视图，用户使用时直接对定义好的视图进行查询。

根据本系统的使用特点，可在

存储账户和账户相关系统两个关系上建立视图：

存储账号信息（账号，密码，系统名字，系统官网，系统类型…）：

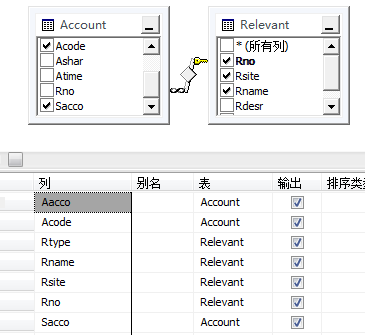


图4.12 表与表的联系

存储账户、账户相关系统与账户共享三个关系上建立视图：

共享账号信息（账号，密码，系统名字，系统官网，系统类型…）：

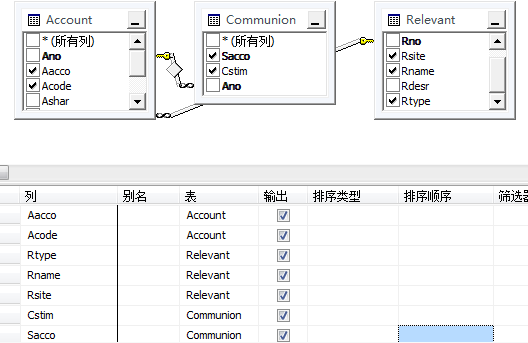


图4.13 相关表的联系

1. **物理结构设计**
   1. 建立索引

根据本系统的功能特点，可知

系统账户（系统账号，系统密码，可用共享值，总共享值）关系模式中的“系统账号”属性经常在连接操作的连接条件中出现，则可考虑在此属性上建立索引：



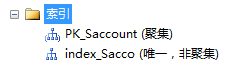


图5.1 建立索引

存储账户基本信息（账号编号，账号，密码，所需共享值，共享时长）中的“账号”经常在查询条件中出现，所以可考虑在此属性上建立索引：



图5.2

索引的建立一般可采取B+树索引方法，这里按照SQL Server默认设置。

* 1. 数据存放

根据系统实际使用情况将数据分为经常存储部分和存储频率较低部分并分开存放。将表和索引或者比较大的表存放在不同的磁盘上，从而提高物理I/O读写效率或加快存取速度，后者在多用户环境下特别有效。

* 1. 系统配置

系统按默认配置，在运行时再根据系统实际运行情况作进一步的调整，如限制同时接入数据库用户书，物理块的大小，锁的数目等。

1. **数据库连接与应用**
   1. VC++中使用MFC通过ADO连接数据库
2. 首先，要用#import语句来引用支持ADO的组件类型库(\*.tlb)：

直接在Stdafx.h文件中加入下面语句来实现：

#import "c:/program files/common files/system/ado/msado15.dll" no\_namespace rename("EOF", "adoEOF")

1. 在程序初始过程中需要初始化组件：AfxOleInit()；
2. 通过\_ConnectionPtr、\_RecordsetPtr和\_CommandPtr这三个只能指针直接使用ADO的操作：
3. \_ConnectionPtr接口返回一个记录集或一个空指针。通常使用它来创建一个数据连接或执行一条不返回任何结果的SQL语句，如一个存储过程。使用 \_ConnectionPtr接口返回一个记录集不是一个好的使用方法。对于要返回记录的操作通常用\_RecordserPtr来实现。而用 \_ConnectionPtr操作时要想得到记录条数得遍历所有记录，而用\_RecordserPtr时不需要。
4. \_CommandPtr接口返回一个记录集。它提供了一种简单的方法来执行返回记录集的存储过程和SQL语句。在使用\_CommandPtr接口时，可利用全局 \_ConnectionPtr接口或在\_CommandPtr接口里直接使用连接串。如果只执行一次或几次数据访问操作，后者是比较好的选择。但如果要频繁访问数据库，并要返回很多记录集，那么，应该使用全局\_ConnectionPtr接口创建一个数据连接，然后使用\_CommandPtr 接口执行存储过程和SQL语句。
5. \_RecordsetPtr是一个记录集对象。它不一定要使用一个已经创建的数据连接，可以用一个连接串代替连接指针赋给 \_RecordsetPtr的connection成员变量，让它自己创建数据连接。如果你要使用多个记录集，最好的方法是同Command对象一样使用已经创建了数据连接的全局\_ConnectionPtr接口，然后使用\_RecordsetPtr执行存储过程和SQL语句。
   1. 数据库的运行与维护

1）登陆页面

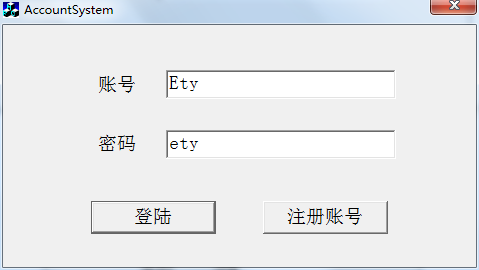


图6.1 登录界面

检查账号与密码是否与数据库中的相对应的SQL查询语句：

check.Format("

SELECT \*

FROM Saccount

WHERE Sacco='%s' AND Scode='%s'",

m\_lacco,m\_lcode);

pRst=pConn->Execute((class \_bstr\_t)check,NULL,adCmdText);

2）注册界面



图6.2 注册界面

检查填写的注册信息是否与数据库中存储的信息是否冲突的SQL以及程序判断语句：

check.Format("

SELECT \*

FROM Saccount

WHERE Sacco='%s'",m\_acco);

pRst=pConn->Execute((class \_bstr\_t)check,NULL,adCmdText);

账号检查无冲突后向数据库中添加新系统用户信息的SQL程序语句：

insert.Format("

INSERT INTO Saccount (Sacco,Scode,Stsha,Scsha)

VALUES ('%s','%s','%d','%d')",

m\_acco,m\_code,0,0);

pRst=pConn->Execute((class \_bstr\_t)insert,NULL,adCmdText);

insert.Format("

INSERT INTO Users(Uphno,Sacco,Uname,Uage,Ueduc,Uceno,Uemai)

VALUES ('%s','%s','%s','%d','%s','%s','%s')",

m\_phno,m\_acco,m\_name,m\_age,sedu,m\_ceno,m\_emai);

pRst=pConn->Execute((class \_bstr\_t)insert,NULL,adCmdText);

3）查询主界面



图6.3 查询主界面

查询存储的账号信息时的部分SQL查询编程语句：

query.Format("

SELECT \*

FROM ACC\_Rele

WHERE Sacco ='%s'",

p->m\_saccount);

（ACC\_Rele为定义好的“存储账号信息”视图）

删除存储的账号信息时的SQL相关编程语句：

dele.Format("

DELETE

FROM Relevant

WHERE Rno='%d'",

Rno);

pRst=pConn->Execute((class \_bstr\_t)dele,NULL,adCmdText);

4）添加存储账户信息界面



图6.4 添加界面

5）修改个人信息界面



图6.5 修改界面

修改个人信息资料时所对应的SQL相关语句：

alter.Format("

UPDATE Users

SET Uphno = '%s',Uname = '%s',Uage = '%d',Ueduc = '%s',Uceno = '%s',Uemai = '%s'

WHERE Sacco ='%s'",

m\_phno,m\_name,m\_age,m\_edcu,m\_ceno,m\_email,p->m\_saccount);

pRst=pConn->Execute((class \_bstr\_t)alter,NULL,adCmdText);

alter.Format("

UPDATE Saccount SET Scode = '%s'

WHERE Sacco ='%s'",

m\_code,p->m\_saccount);

pRst=pConn->Execute((class \_bstr\_t)alter,NULL,adCmdText);

6）统计各类账号信息界面

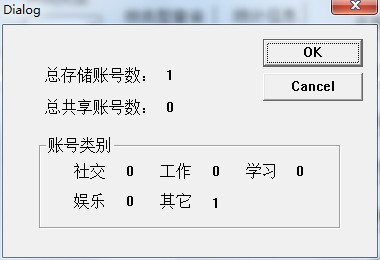


图6.6 统计界面

统计各类账号总数的相关SQL编程语句：

count.Format("

SELECT COUNT(\*)

FROM ACC\_Rele

WHERE Sacco ='%s'",

p->m\_saccount);

pRst=pConn->Execute((class \_bstr\_t)count,NULL,adCmdText);

count.Format("

SELECT COUNT(\*)

FROM ACC\_SHARE

WHERE Sacco ='%s'",

p->m\_saccount);

pRst=pConn->Execute((class \_bstr\_t)count,NULL,adCmdText);

count.Format("

SELECT COUNT(\*)

FROM ACC\_Rele

WHERE Sacco ='%s' AND Rtype ='%s'",

p->m\_saccount,"工作 ");

pRst=pConn->Execute((class \_bstr\_t)count,NULL,adCmdText);

1. **收获与体会**

在这次的数据库课程设计中，我充分运用在课堂上学到的数据库知识，真正地做到了学以致用。在数据库建立的每一个步骤我都仔细思考，争取把每一步都尽量做好，如在逻辑结构设计中，我尽量把每张表都进行优化处理，适当地提高所属的范式级别。而我对数据库的重要性有了更进一步的认识，明白到设计一个好的数据库与设计一个好的数据库应用系统是紧密相关的，一个好的数据库结构是应用系统的基础。我对SQL Server 2008的操作也熟悉了不少，一开始在实验课时接触此数据库管理系统时还很生疏，感觉不容易操作，但经过实验课以及这次的课程设计后，SQL Server 2008现在用得也很顺手了，里面的一些可视化操作也都能熟练运用；而在开发时编程实现各种查询也让我对于SQL语句也有了更深的掌握。

在课程设计刚开始时，我遇到了不少书上没有深入介绍的知识相关问题，例如数据库访问技术，数据库编程的一些相关问题，而在后来我通过自己的独立思考以及网上的学习资源，都能顺利找到了解决的方法，

这次的课程设计，我收获最大的一点是，之前没完全理解的书上的某些跟实际联系比较强的知识点，比如视图的作用与意义等，现在通过自己的具体实践，都对这些得到了更深一步的掌握。

感谢老师这一学期的理论指导与实验指导，让我学到了不少有关“数据库”的知识，并顺利完成了此次的课程设计。