

**课程设计报告**

( 2020--2021年度第1学期)

名 称：数据库原理课程设计

题 目： 物资管理信息系统

院 系： 计算机系

班 级： 信安1802

学 号： 201809000118

学生姓名： 王楚璇

指导教师： 廖尔崇

设计周数： 1

成 绩：

日期：2020年 12月 25日

**《数据库原理课程设计》课程设计**

**任 务 书**

1. **目的与要求**
2. 本实验是为信息安全等专业的学生在学习数据库原理后，为培养更好的解决问题和实际动手能力而设置的实践环节。通过这个环节，使学生具备应用数据库原理对数据库系统进行设计的能力。为后继课程和毕业设计打下良好基础。
3. 通过该实验，培养学生在建立数据库系统过程中使用关系数据理论的能力。
4. 通过对一个数据库系统的设计，培养学生对数据库需求分析、数据库方案设计、系统编码、界面设计和软件调试等各方面的能力。是一门考查学生数据库原理、面向对象设计方法、软件工程和信息系统分析与设计等课程的综合实验。
5. **主要内容**

针对一个具有实际应用场景的中小型系统（见题目附录）进行数据库设计，重点分析系统涉及的实体、实体之间的联系，实现增加、删除、更新、查询数据记录等基本操作。大致分为如下步骤：

1. 理解系统的数据库需求，分析实体及实体间联系，画出E-R图：

1. 分析确定实体的属性和码，完成对该实体的实体完整性、用户自定义完整性的定义。
2. 设计实体之间的联系，包括联系类型和联系的属性。最后画出完整的E-R图。

2．根据设计好的E-R图及关系数据库理论知识设计数据库模式：

1. 把E-R图转换为逻辑模式；
2. 规范化设计。使用关系范式理论证明所设计的关系至少属于3NF并写出证明过程；如果不属于3NF则进行模式分解，直到该关系满足3NF为止，要求写出分解过程。
3. 设计关系模式间的参照完整性，要求实现级联删除和级联更新。
4. 用SQL语言完成数据库内模式的设计。

3．数据库权限的设计：

1. 根据系统分析，完成授权操作；
2. 了解学习收回权限的操作。

4．完成用户界面的设计，对重要数据进行加密。

5．连接数据库，用宿主语言实现系统所需的各种操作：

1. 实现数据记录的录入、删除、查询和修改。
2. 以视图的形式完成复杂查询，比如多表、多条件等。

**三、 进度计划**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **设计(实验)内容** | **完成时间** | **备注** |
| 1 | 根据任务书完成信息模型（概念模型、逻辑模型、完整性、规范化）的设计，并基于选用的DBMS实现该信息模型，然后录入初始数据 | 2020.12.21 |  |
| 2 | 根据任务书完成各种数据定义和数据操作，并保留所有SQL语句。 | 2020.12.21 |  |
| 3 | 数据库权限设计，用户界面设计 | 2020.12.22 |  |
| 4 | 用可视化开发工具环境开发学生选定的信息系统（C/S或者B/S模式） | 2020.12.22 |  |
| 5 | 系统的完善与验收 | 2020.12.23 |  |

**四、 设计（实验）成果要求**

1. 在DBMS（如oracle，SQL Server 2005/2008，DB2等）上完成完整的数据库的设计；
2. 使用可视化开发平台完成信息系统，要求可以正确运行；
3. 完成实验报告。

**五、 考核方式**

1. 在微机上检查数据库模式的设计、三大完整性的设计、关系属于几范式等；
2. 在微机上检查系统的运行结果，要求学生阐述使用的相关技术；
3. 实验报告的检查。

**六、 题目附录**

* 1. 图书管理信息系统
  2. 物资管理信息系统
  3. 汽车销售管理信息系统
  4. 超市管理信息系统
  5. 通讯录管理信息系统
  6. 工资管理信息系统
  7. 酒店管理信息系统
  8. 小区物业管理信息系统

学生姓名：

指导教师：

2020年 12月 21日

**一、课程设计(综合实验)的目的与要求**

1.通过课程设计，学生具备应用数据库原理对数据库系统进行设计的能力，为后继课程和毕业设计打下良好基础。

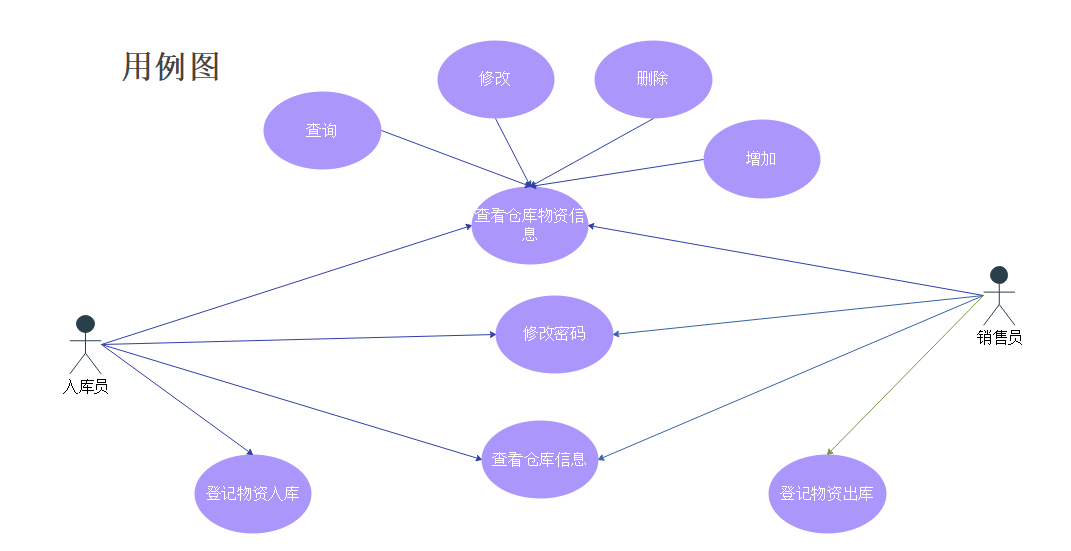
2.通过该实验，培养学生在建立数据库系统过程中使用关系数据理论的能力。

3.通过对一个数据库系统的设计，培养学生对数据库需求分析、数据库方案设计、系统编码、界面设计和软件调试等各方面的能力。是一门考查学生数据库原理、面向对象设计方法、软件工程和信息系统分析与设计等课程的综合实验。

**二、设计（实验）正文**

1.需求分析

将登录对象分为“入库员”和“销售员”，分别行使系统的“入库”与“出库”功能。两种对象都能查看自己所在仓库的信息，与增删改查对应仓库的基本物资表，修改个人信息，而“入库员”能对物资进行入库登记，记录在表中，“出库员”对物资进行出库操作，记录在对应的表中。



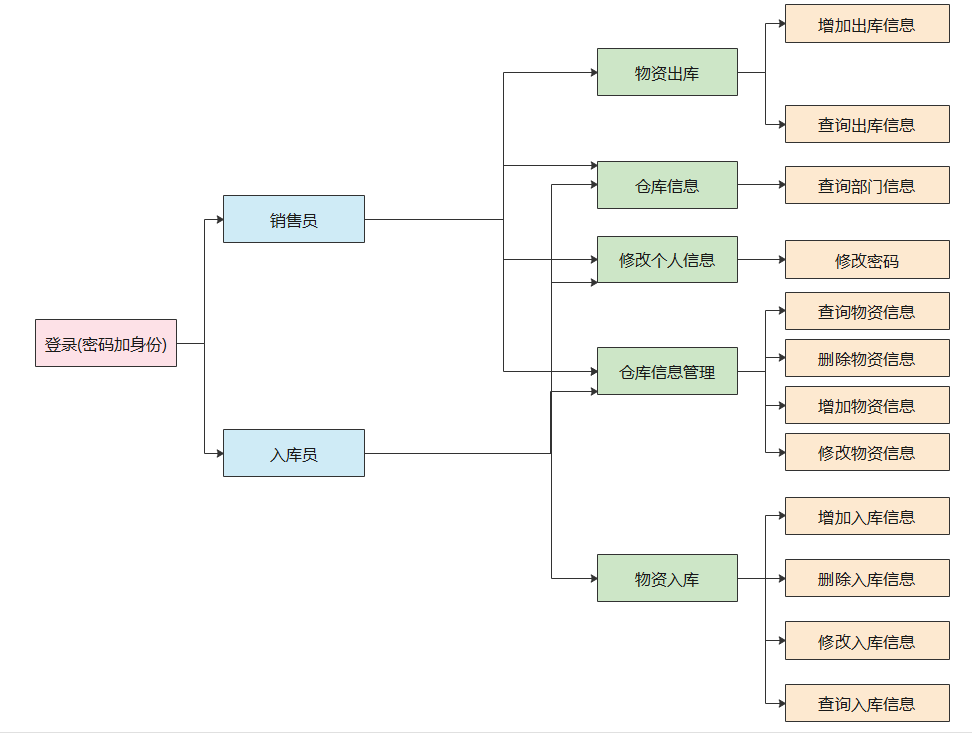
2.系统功能设计

2.1系统描述

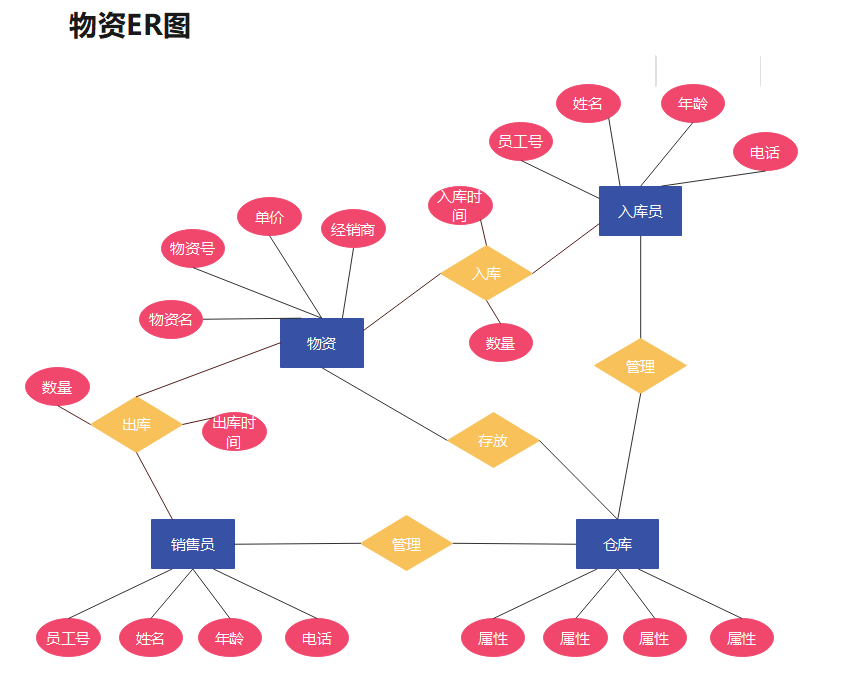
物资管理信息系统是企业与生产单位之间的桥梁、纽带，是企业正常生产的物资保障。

本系统利用信息技术提高物资管理的水平，主要功能是物资管理与出入库的信息登记。物资管理可以查看自己所在仓库的信息，增删改查对应仓库的基本物资表，还可以进行排序，而出入库则是根据身份的不同（入库员或销售员）进行入库或出库操作，登记信息，形成入库信息表和出库信息表，两个表都能进行查找物资号或物资名得到想要的信息。此外还有修改个人的密码等一些功能。

2.2系统功能结构图



3.数据库概念设计



1. 数据库逻辑设计

Store:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 仓库表 | | | | |
| 列名 | 数据类型 | 长度 | 约束 | 描述 |
| sId | char | 8 | Primary key | 仓库号 |
| sName | varchar | 20 |  | 仓库名 |
| sPlace | varchar | 20 |  | 地址 |
| sPhone | char | 8 |  | 电话号码 |
| ps | varchar | 40 |  | 备注 |

范式分析：

F={sId->sName, sId->sPlace, sId->sPhone, sId->ps}，此关系模式所有属性都是不可分割的基本数据项，属于1NF。并且每一个非主属性都完全函数依赖于码，没有非主属性对码的部分函数依赖，属于2NF。并且不存在非主属性对码的传递函数依赖，属于3NF。

Cman:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 库管员表 | | | | |
| 列名 | 数据类型 | 长度 | 约束 | 描述 |
| cId | char | 8 | Primary key | 员工号 |
| cName | varchar | 20 | Not null | 姓名 |
| cAge | int |  |  | 年龄 |
| cRank | varchar | 10 |  | 职位 |
| storeId | char | 10 |  | 工作仓库号 |
| cPhone | char | 11 |  | 电话号码 |
| password | varchar | 20 | Not null | 密码 |

范式分析：

F={cId->cName, cId->cAge, cId->cRank, cId->storeId, cId->cPhone, cId->password}，此关系模式所有属性都是不可分割的基本数据项，属于1NF。并且每一个非主属性都完全函数依赖于码，没有非主属性对码的部分函数依赖，属于2NF。并且不存在非主属性对码的传递函数依赖，属于3NF。

Goods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 物资存放表 | | | | |
| 列名 | 数据类型 | 长度 | 约束 | 描述 |
| goodsId | char | 8 | Primary key | 物资id |
| goodsName | varchar | 20 | Not null | 物资名 |
| number | int |  | Not null | 数量 |
| vaule | int |  |  | 单价 |
| shop | varchar | 20 |  | 经销商 |
| storeId | char | 10 | Not null | 仓库号 |
| ps | varchar | 40 |  | 备注 |

范式分析：

F={goodsId->goodsName ,goodsId->number ,goodsId->value ,goodsId->shop ,goodsId->storeId ,goodsId->ps}，此关系模式所有属性都是不可分割的基本数据项，属于1NF。并且每一个非主属性都完全函数依赖于码，没有非主属性对码的部分函数依赖，属于2NF。并且不存在非主属性对码的传递函数依赖，属于3NF。

Table\_in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入库表 | | | | |
| 列名 | 数据类型 | 长度 | 约束 | 描述 |
| goodsId | char | 8 | Primary key | 物资id |
| goodsName | varchar | 20 | Not null | 物资名 |
| number | int |  | Not null | 数量 |
| value | int |  |  | 单价 |
| inTime | datetime |  | Not null | 入库时间 |
| storeId | char | 10 | Not null | 仓库号 |
| shop | varchar | 20 |  | 经销商 |
| ps | varchar | 40 |  | 备注 |

范式分析：

F={ goodsId->goodsName ,goodsId->number ,goodsId->value ,goodsId->shop ,goodsId->storeId ,goodsId->ps }，此关系模式所有属性都是不可分割的基本数据项，属于1NF。并且每一个非主属性都完全函数依赖于码，没有非主属性对码的部分函数依赖，属于2NF。并且不存在非主属性对码的传递函数依赖，属于3NF。

出库表与入库表有不同，goodsId不能单独成为主码，而（goodsId,outTime）作为码又会产生部分函数依赖，故将出库表分解成：

Table\_out:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 出库表 | | | | |
| 列名 | 数据类型 | 长度 | 约束 | 描述 |
| goodsId | char | 8 | Primary key | 物资id |
| number | int |  | Not null | 出库数量 |
| inTime | datetime |  | Not null | 入库时间 |

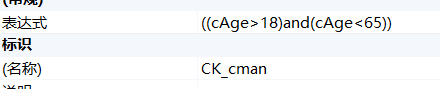
而当展示信息的时候，将通过与goods表拼接展示完整信息。

设计时要注意各关系模式的函数依赖是否能化成3NF甚至BCNF（每一个决定属性集都包含码），此处仅讨论是否为3NF。

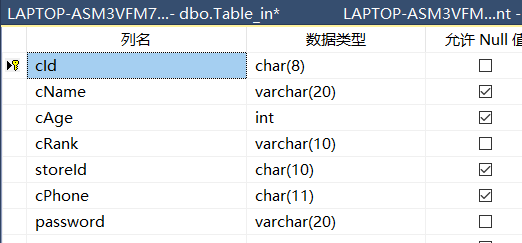
1. 数据库物理设计

Cman表：

Primary key---cId

用户自定义完整性：

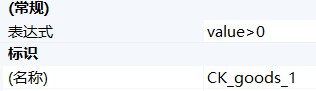
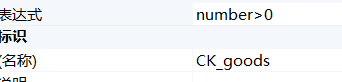
参照完整性：Foreign key(storeId) references store(sId)

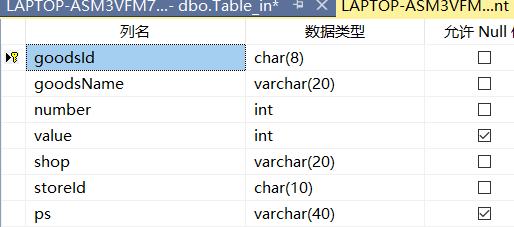


Goods表：

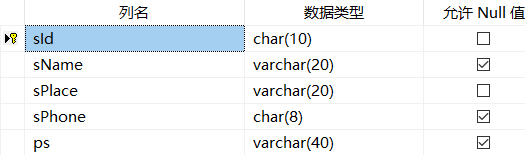
primary key--goodsId

用户自定义完整性：





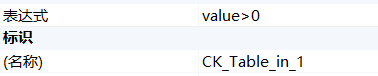
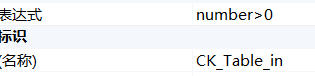
Store表：primary key--storeId



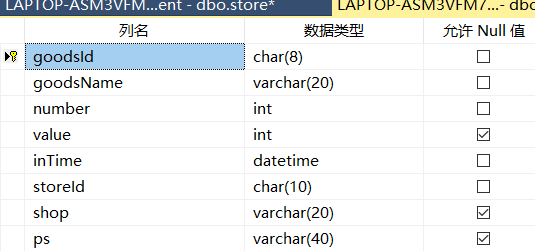
Table\_in表：

primary key--goodsId

用户自定义完整性：



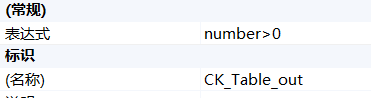
参照完整性：Foreign key(storeId) references store(sId)



Table\_out:

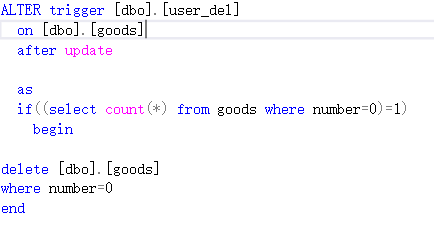
primary key—(goodsId,outTime)

用户自定义完整性：





使用了触发器，当物资出库数量减少为零时，将该物资从物资表中删除。



1. 系统编码与实现

6.1部分代码

private void insertg()//插入{

DBHelper dBHelper = new DBHelper();

try{

string sql = string.Format("insert " +

"into [goods]"+ "values('{0}','{1}',{2},{3},'{4}','{5}','{6}')",textBox1.Text.Trim(),textBox2.Text.Trim(),Convert.ToInt32(textBox3.Text),Convert.ToInt32(textBox4.Text),textBox5.Text.Trim(),textBox6.Text.Trim(),textBox7.Text.Trim());

SqlCommand cmd = new SqlCommand(sql, dBHelper.Connection);

dBHelper.openConnection();

int result = cmd.ExecuteNonQuery();//受影响行数

if(result==1)

{

MessageBox.Show("添加成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

}

}

catch(Exception){

MessageBox.Show("数据库操作失败", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally{

dBHelper.CloseConnection();

}

}

private void gaingoods()//获取原始数据{

DBHelper dBHelper = new DBHelper();

try{

string sql = string.Format("select \*" +

"from [goods]" +

"where goodsId='{0}'",goodsID);

SqlCommand cmd = new SqlCommand(sql, dBHelper.Connection);

dBHelper.openConnection();

SqlDataReader reader = cmd.ExecuteReader();

if(reader.Read())//可读{

this.textBox1.Text = reader["goodsId"].ToString();

this.textBox2.Text = reader["goodsName"].ToString();

this.textBox3.Text = reader["number"].ToString();

this.textBox4.Text = reader["value"].ToString();

this.textBox5.Text = reader["shop"].ToString();

this.textBox6.Text = reader["storeId"].ToString();

this.textBox7.Text = reader["ps"].ToString();

}

reader.Close();

}

catch (Exception){

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally{

dBHelper.CloseConnection();

}

}

private void updateg(){

DBHelper dBHelper = new DBHelper();

try{

string sql = string.Format(@"update [goods]" +

"set goodsId='{0}',goodsName='{1}',number={2},value={3},shop='{4}',storeId='{5}',ps='{6}'" +

"where goodsId='{7}'", textBox1.Text.Trim(), textBox2.Text.Trim(), Convert.ToInt32(textBox3.Text), Convert.ToInt32(textBox4.Text), textBox5.Text.Trim(), textBox6.Text.Trim(), textBox7.Text.Trim(),goodsID);

SqlCommand cmd = new SqlCommand(sql.ToString(), dBHelper.Connection);

dBHelper.openConnection();

int result = cmd.ExecuteNonQuery();

if(result==1){

MessageBox.Show("修改成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();//关闭，方便刷新

}

}

catch (Exception){

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally{

dBHelper.CloseConnection();

}

}

6.2功能截图

登录：



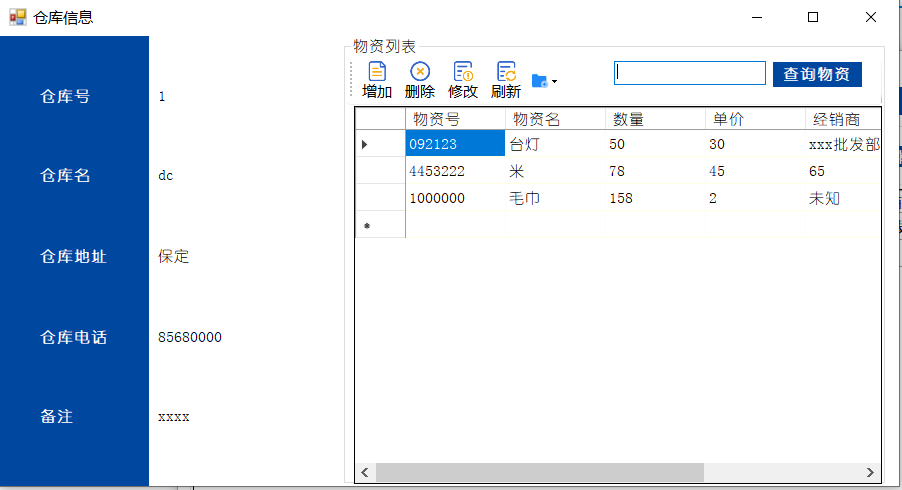
进货：由于该职员工作于1号仓库，故增加时“仓库号”输入框不可更改，但是当选择修改时可以进行修改。

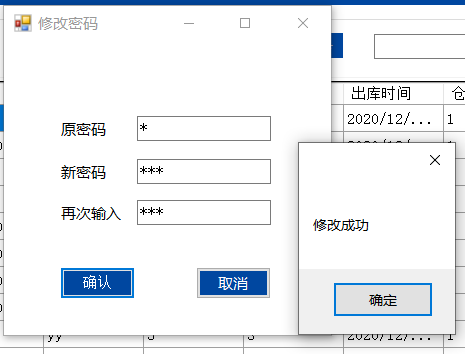


进货表查询：



仓库信息：(分别能进行增、删、改、刷新、排序、查询)





**三、课程设计（综合实验）总结或结论**

这个实验是我第一次用c#写代码，在一开始有一定挑战性，我在网上查阅了资料，参考了老师提供的视频文档，完成了一个基本符合要求的系统。

在一开始我设计数据表的时候遇见了问题，在出库表，我本用outTime作为主码，在老师的提醒下我发现当多个用户使用系统的时候就不能保证他的唯一性，所以我修改了表，把goodsId和outTime同时作为联合主码，其余信息通过表的连接得到。

在编写过程中，关于用户自定义完整性，我一开始是写进c#代码里，但是经过老师的指导，我在数据库中进行了添加完整性，提高了效率。

我还改进了界面，使界面更加美观，一体化，还对登录窗口通过控件达到渐变消失的效果。

在实验过程中，我最大的感受是自己调试代码的能力还有待完善，途中遇见了一个问题，始终得不到解决，设断点也找不到出路，后来在老师的指导下，我先把Sql语句进行输出，然后把Sql语句在数据库中进行执行，终于发现是自己的时间没有用引号括起来，这真是一个小错误，可是又那么致命，在以后的实验中我会更加仔细，每一个小地方都要认真检查。争取尽最少的时间完成工作。最后感谢老师的帮助，在今后的道路中我会不断提高自己的能力，培养不骄不躁的心态。

**四、同学间相互协作情况**

1 帮助他人（名称+学号）的内容；

2 他人（名称+学号）帮助我的内容。

**五、参考文献**

[1]王国胜, 张石磊. c#基础与案例开发详解. 清华大学出版社,2014年3月

**附录（设计流程图、程序、表格、数据等）**

namespace sjk物资管理信息系统//与数据库相连的Class

{

public class DBHelper

{

private string cString = @"Data Source=.;Initial Catalog=goodsManagement;Integrated Security=true";

private SqlConnection connection;

/// <summary>

/// 连接对象

/// </summary>

public SqlConnection Connection

{

get

{

if(connection==null)

{

connection = new SqlConnection(cString);

}

return connection;

}

}

/// <summary>

/// 打开数据库连接

/// </summary>

public void openConnection()

{

if(Connection.State==ConnectionState.Closed)

{

Connection.Open();

}

else if(Connection.State == ConnectionState.Broken)

{

Connection.Close();

Connection.Open();

}

}

/// <summary>

/// 关闭数据库

/// </summary>

public void CloseConnection()

{

if(Connection.State==ConnectionState.Open|| Connection.State == ConnectionState.Broken)

{

Connection.Close();

}

}

}

}

注：根据课程设计、综合实验的内容将标题任选其一。