**《数据库系统原理》**

**课程设计报告**

　　　　　　　　　　（2019—2020学年第一学期）

题 目 仓库管理系统

学生姓名 赵 猛

专业班级 计实验17-1

学生学号 311709000714

教师姓名 吴 岩

成 绩**：**

评 语：

教师签名：

日期：

摘要：仓库管理系统对于一个企业来说是必不可少的组成部分。目前好多企业还停留在管理人员手工记录数据的最初阶段，手工记录对于规模小的企业来说还勉强可以接受，但对于信息量比较庞大，需要记录存档的数据比较多的高校来说，人工记录是相当麻烦的。而且当查找某条记录时，由于数据量庞大，还只能靠人工去一条一条的查找，这样不但麻烦还浪费了许多时间，效率也比较低。当今社会是飞速进步的世界，原始的记录方式已经被社会所淘汰了，计算机化管理正是适应时代的产物。信息世界永远不会是一个平静的世界，当一种技术不能满足需求时，就会有新的技术诞生并取代旧技术。21世纪的今天,信息社会占着主流地位，计算机在各行各业中的运用已经得到普及，自动化、信息化的管理越来越广泛应用于各个领域。我们针对如此，设计了一套仓库管理系统。仓库管理系统采用的是计算机化管理，系统做的尽量人性化，使用者会感到操作非常方便，管理人员需要做的就是将数据输入到系统的数据库中去。由于数据库存储容量相当大，而且比较稳定，适合较长时间的保存，也不容易丢失。这无疑是为信息存储量比较大的企业提供了一个方便、快捷的操作方式。本系统具有运行速度快、安全性高、稳定性好的优点，并且具备修改功能，能够快速的查询企业所需的信息。

目录

[1 系统开发概述 1](#_Toc28875176)

[1.1 系统开发背景 1](#_Toc28875177)

[1.2 系统功能需求分析 1](#_Toc28875178)

[1.2.1 系统总体设计图 2](#_Toc28875179)

[1.3 系统数据流程分析 2](#_Toc28875180)

[2 数据库概念结构设计 3](#_Toc28875181)

[2.1 实体分析 3](#_Toc28875182)

[2.2 各实体E-R图 3](#_Toc28875183)

[2.2.1 仓库管理员 3](#_Toc28875184)

[2.2.2 货物 4](#_Toc28875185)

[2.2.3 供应商 4](#_Toc28875186)

[2.2.4 客户 5](#_Toc28875187)

[2.3 全局E-R图 5](#_Toc28875188)

[3 数据库逻辑结构设计 6](#_Toc28875189)

[4 数据库物理设计与实现 7](#_Toc28875190)

[4.1 数据库的创建 7](#_Toc28875191)

[4.2 数据表的创建 7](#_Toc28875192)

[4.2.1 仓库管理员表 7](#_Toc28875193)

[4.2.2 供应商表 7](#_Toc28875194)

[4.2.3 客户表 8](#_Toc28875195)

[4.2.4 进仓表 8](#_Toc28875196)

[4.2.5 出仓表 8](#_Toc28875197)

[4.3 数据完整性设计 8](#_Toc28875198)

[4.3.1 主键约束的创建 8](#_Toc28875199)

[4.3.2 外键约束的创建 9](#_Toc28875200)

[4.3.3 UNIQUE约束的创建 9](#_Toc28875201)

[4.3.4 DEFAULT约束的创建 10](#_Toc28875202)

[4.3.5 CHECK约束的创建 11](#_Toc28875203)

[4.4 索引的创建 11](#_Toc28875204)

[4.5 视图的创建 12](#_Toc28875205)

[4.6 数据操纵 13](#_Toc28875206)

[4.7 数据查询 14](#_Toc28875207)

[4.8 触发器的创建 15](#_Toc28875208)

[4.9 存储过程的创建 16](#_Toc28875209)

[5 应用系统功能实现 17](#_Toc28875210)

[5.1 C#介绍 17](#_Toc28875211)

[5.2 效果 17](#_Toc28875212)

[5.2.1 登录界面 17](#_Toc28875213)

[5.2.2 菜单界面 18](#_Toc28875214)

[5.2.3 货物查询 19](#_Toc28875215)

[5.2.4 入仓信息管理 20](#_Toc28875216)

[5.2.5 出仓信息管理 21](#_Toc28875217)

[5.2.6 密码修改 22](#_Toc28875218)

[5.3 实现过程 23](#_Toc28875219)

[5.3.1 登录界面代码 23](#_Toc28875220)

[5.3.2 主界面代码 25](#_Toc28875221)

[5.3.3 货物查询代码 26](#_Toc28875222)

[5.3.4 入仓信息管理代码 29](#_Toc28875223)

[5.3.5 出仓信息管理代码 33](#_Toc28875224)

[5.3.6 密码修改代码 37](#_Toc28875225)

[6 总结 40](#_Toc28875226)

# 系统开发概述

## 系统开发背景

对于一个生产商品或者经营商品的单位来说，建立一个仓库是十分必要的，各种商城、超市需要使用仓库存放物资，企业和工厂需要使用仓库存放生产原料和成品，由于仓库占据着举足轻重的地位，因此仓库管理必须高效，一个高效的仓库管理系统，可以为生产和经营提供强有力的支持，而效率低下甚至混乱不堪的仓库会成为企业健康发展的约束。相较于普通的人力管理仓库，使用计算机系统性的管理可以大幅度的减少不必要的人力和物力。

## 系统功能需求分析

仓库管理系统主要是实现库存货物的管理，对货物的出库和入库的管理，和对仓库系统维护的功能。

功能包括：

* 货物入库管理

入库单包括单号、供应商信息、货物编号、交易额和数量，入库单的添加。

* 货物出库管理

出库单包括单号、客户信息、货物编号、交易额和数量，出库单的添加。

* 货物查询

输入货物的编号即可查询仓库中该货物的所有的信息：名称、进价、售价、剩余数量。

* 用户账户的管理

用户可以修改登录密码。

### 系统总体设计图



图1-1. 系统总体设计图

## 系统数据流程分析



图1-2. 系统数据流程图

# 数据库概念结构设计

## 实体分析

* 仓库管理员

属性：编号、姓名、登录密码、联系方式

* 货物

属性：编号、数量、货物名称、入价、售价

* 供应商

属性：编号、名称、联系方式、地址

* 客户

## 各实体E-R图

### 仓库管理员



图2-1仓库管理员E-R图

### 货物



图2-2. 货物E-R图

### 供应商



图2-3 供应商E-R图

### 客户



图2-4 客户E-R图

## 全局E-R图



图2-5 全局E-R图

# 数据库逻辑结构设计

E-R图转换为对应的关系模式：

仓库管理员：{**仓库管理员编号**，姓名，登陆密码，联系方式}；

货物：{**货物编号**，货物名称，单价，进价，售价}；

供应商：{**供应商编号**, 供应商名称，联系方式，地址}；

客户：{**客户编号**, 客户名称，联系方式，地址}；

供应：{**供应商编号，货物编号**，货物数量，交易金额，交易日期}；

订购：{**客户编号，货物编号**，货物数量，交易金额，交易日期}；

# 数据库物理设计与实现

## 数据库的创建

使用Microsoft SQL Server Management Studio创建数据库，数据库名称为CKGL,设置初始大小为8MB，增量为64MB，最大文件大小无限制，数据库文件和日志文件都存放在C:\Database中。



图4-1. 数据库的创建

## 数据表的创建

### 仓库管理员表

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 允许NULL值 |
| Manager\_no | Nchar(10) | No |
| Manager\_name | Nvarchar(10) | No |
| Manager\_psd | Nvarchar(16) | No |
| Manager\_phone | char(11) | No |

### 供应商表

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 允许NULL值 |
| supplier\_no | nchar(10) | No |
| supplier\_name | nvarchar(20) | No |
| supplier\_phone | nchar(11) | No |
| supplier\_address | nvarchar(20) | No |

### 客户表

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 允许NULL值 |
| client\_no | nchar(10) | No |
| client\_name | nvarchar(20) | No |
| client\_phone | nchar(11) | No |
| client\_address | nvarchar(20) | No |

### 进仓表

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 允许NULL值 |
| in\_no | nchar(10) | No |
| supplier\_no | nchar(10) | No |
| hw\_no | nchar(10) | No |
| in\_num | int | No |
| in\_money | int | No |
| in\_date | time(7) | No |

### 出仓表

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 允许NULL值 |
| out\_no | nchar(10) | No |
| client\_no | nchar(10) | No |
| hw\_no | nchar(10) | No |
| out\_num | int | No |
| out\_money | int | No |
| out\_date | time(7) | No |

## 数据完整性设计

### 主键约束的创建

对每个表设置主键约束，仓库管理员表的主键为Manger\_no，货物表的主键为Hw\_no，供应商表的主键为supplier\_no，客户表的主键为client\_no，进仓表的主键为in\_no，出仓表的主键为out\_no。

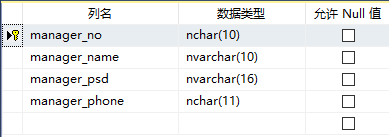


图4-2. manager表主键约束的创建

### 外键约束的创建

在对象资源管理器中右键CKGL中的数据库关系图，点击新建数据库关系图，选中suppliers、in、HW三个表，将in表中的supplier\_no、hw\_no、分别拖动到其他三个表上，设置为外键。

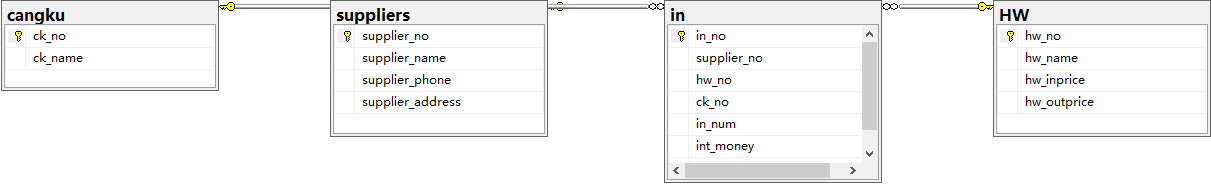


图4-3. in表外键约束的创建

同样的，对out表也进行相似的处理，为out表设置外键约束。

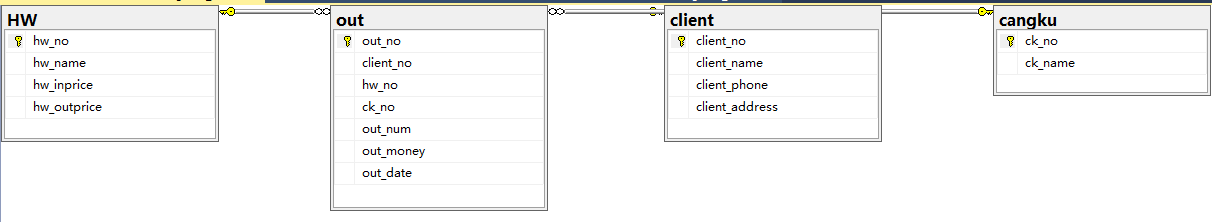


图4-4. out表外键约束的创建

### UNIQUE约束的创建

为了确保列中的值是唯一的，需要设置唯一性约束，用户可以对一列或者多列定义唯一性约束。

右键单击Manager\_no表设计器，在弹出的快捷菜单中选择“索引/键”命令，弹出“索引/键”对话框，在弹出的“索引/键”对话框中单击添加按钮，添加新的主/唯一键或索引；在“常规”栏上午“类型”右边选择“唯一键”。选择列名和排序规则ASC。

设置完成后，单击关闭保存完成UNIQUE约束的创建。

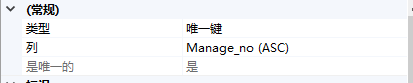


图4-5. UNIQUE约束的创建

直接使用SQL语句也可创建UNIQUE约束：

USE CKGL

GO

CREATE TABLE HW

(

hw\_no nchar(10) PRIMATY KEY,

hw\_name nvarchar(20) CONSTRAINTUN\_ClassName UNIQUE,

hw\_inprice int,

hw\_outprice int

)

### DEFAULT约束的创建

给in表的in\_num创建default约束，默认值为10。

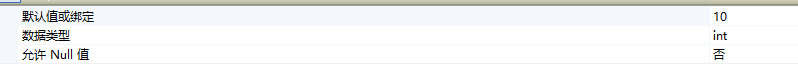


图4-6. DEFAULT约束的创建

创建HW表，同时HW表的hw\_no初始化为0，SQL语句：

USE CKGL

GO

CREATE TABLE HW

(

hw\_no nchar(10) PRIMARY KEY,

hw\_name nvarchar(20),

hw\_inprice int,

hw\_outprice int,

hw\_num int DEFAULT 0

)

效果：



图4-7. DEFAULT值为0

### CHECK约束的创建

给Managers表中的Manager\_psd设置长度为6-16的限制。

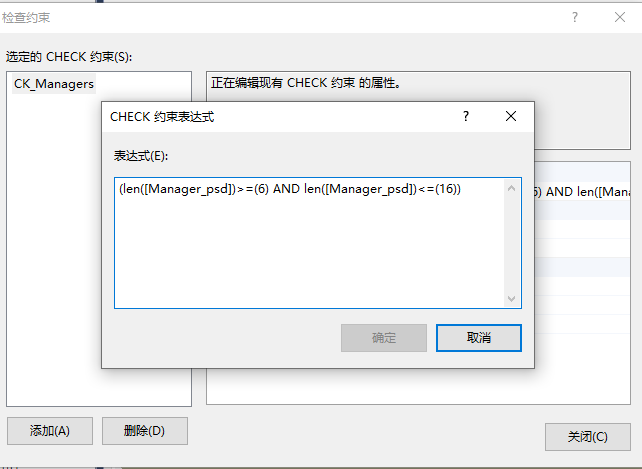


图4-8. CHECK约束的创建

## 索引的创建

使用MSSMS给CKGL数据库中的in表中in\_no列创建索引，右键in\_no列，点击“索引/键”，右键单击添加，增加一个索引，然后设置索引对应的字段为in\_no，排序方式为ASC。

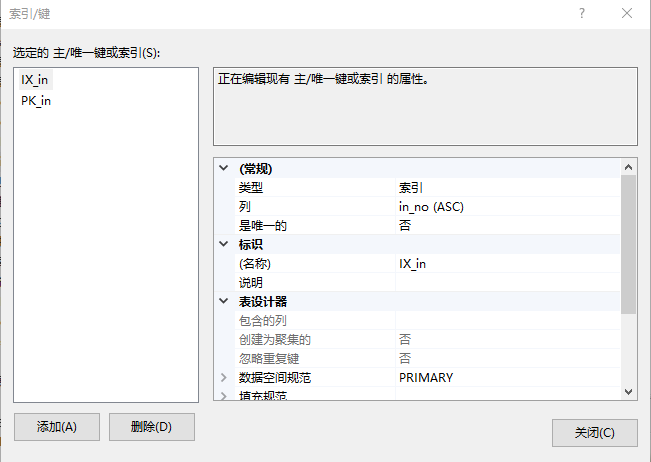


图4-9． 索引的创建

## 视图的创建

右键CKGL数据库中的视图选项，选择创建视图，选择HW、in、suppliers四个表，定义视图的名字为view1。

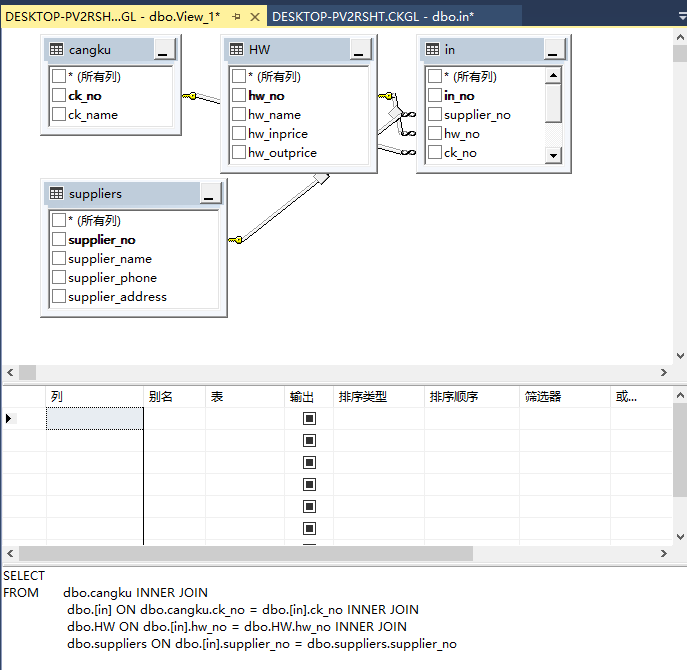


图4-9．View\_1的创建

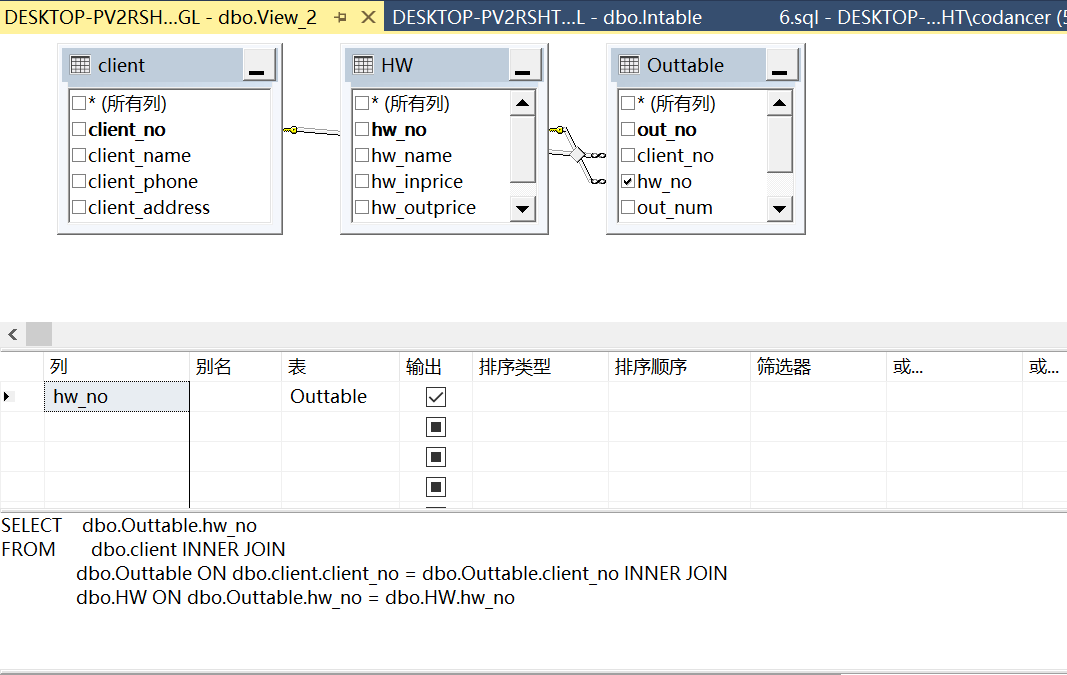


图4-10. View\_2的创建

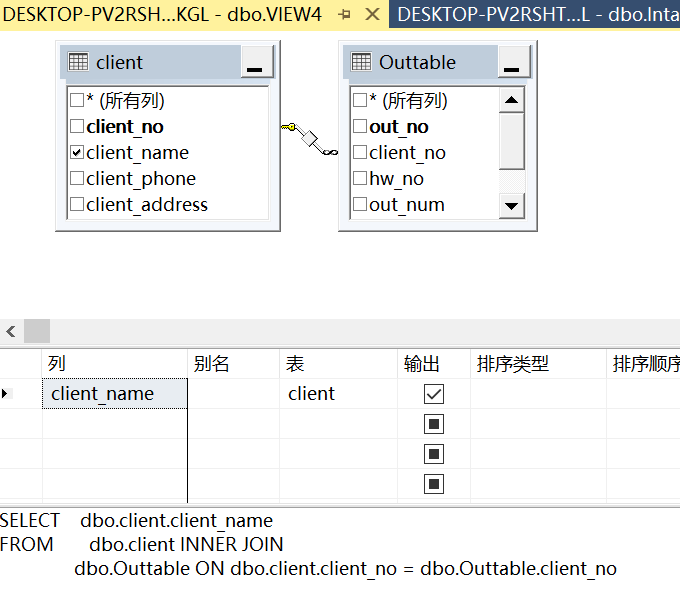


图4-11. VIEW3的创建

## 数据操纵

向CKGL中的每个表中输入数据，选中表之后点击编辑前200行，直接输入数据即可。

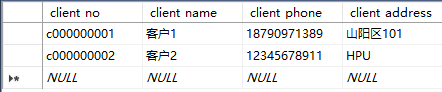


图4-12.利用SQL Server Management Studio插入数据

也可采用SQL语句的方式插入数据：

Insert INTO HW(hw\_no,hw\_name,hw\_inprice,hw\_outprice)

values('hw00000001','苹果','20','25')

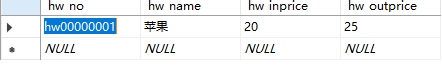


图4-13.使用SQL语句插入数据

同理，利用相同的方法向其他的表中也插入数据或者删除数据。

## 数据查询

1. 查询2019-01-12货物号为hw00000001的入仓总数，其SQL语句为：

SELECT SUM(in\_num)

FROM Intable

WHERE(hw\_no='hw00000001' and in\_date='2019-01-12')

效果如图所示：

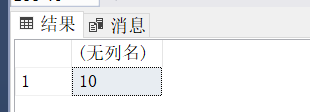


图4-14. 查询1执行的结果

1. 查询供应货物数量累计超过50的供应商的名字

SQL语句：

SELECT suppliers.supplier\_name

FROM suppliers

where

(

suppliers.supplier\_no

in

(

SELECT supplier\_no

FROM Intable

GROUP by supplier\_no

HAVING(SUM(Intable.in\_num)>=50)

)

)

效果：

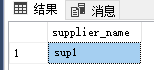


图4-15. 查询2的输出结果

1. 查询2019-10-22的仓库总收入：

SQL语句：

SELECT -SUM(in\_money)+SUM(out\_money) as '当天收入'

FROM Intable,Outtable

WHERE(in\_date='2019-10-22' and out\_date='2019-10-22')

效果：



图4-16. 查询3的结果

## 触发器的创建

为Intable表创建Insert触发器,一旦in\_num小于0就进行ROLLBACK操作。

SQL语句：

USE CKGL

GO

CREATE trigger T1\_insert

ON Intable

FOR INSERT

AS

DECLARE @num int

SELECT @num=Intable.in\_num

FROM Intable

IF @num<0

ROLLBACK

GO

效果：当入仓的数量为负数的时候，显示ERROR错误。

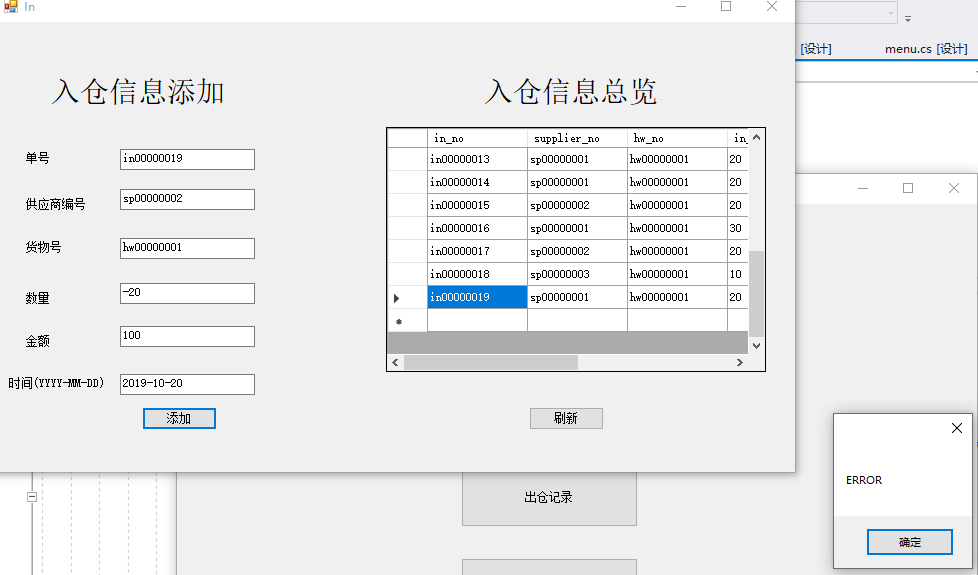


图4-17. 触发器的建立

## 存储过程的创建

创建一个存储过程QUERY能够所有的查询入仓的信息。

代码：

USE CKGL

GO

CREATE PROCEDURE QUERY

AS

SELECT \* FROM Intable

GO

# 应用系统功能实现

应用系统使用了C#的窗体应用程序创建。

## C#介绍

C#是微软推出的一种基于.NET框架的、面向对象的高级编程语言。C#以.NET框架类库作为基础，拥有类似Visual Basic的快速开发能力。C#由安德斯·海尔斯伯格主持开发，微软在2000年发布了这种语言，希望借助这种语言来取代Java。C#已经成为Ecma国际和国际标准组织的标准规范。

## 效果

### 登录界面



图5-1. 登录界面

登录首先要让应用程序与数据库创建连接，连接方式为：先创立一个string说明服务器名称、数据库名称、用户名、密码。

连接代码：

stringConStr="server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

用户在输入账户和密码并且按下登录之后，利用textBox获取用户的输入内容，假设输入的账户string是a，密码string 为b，那么就要在Managers表中查询是否有该记录，如果有的话，则登录成功，否则则显示：“账户/密码错误”。同时还要catch error防止连接不上数据库而造成死循环。

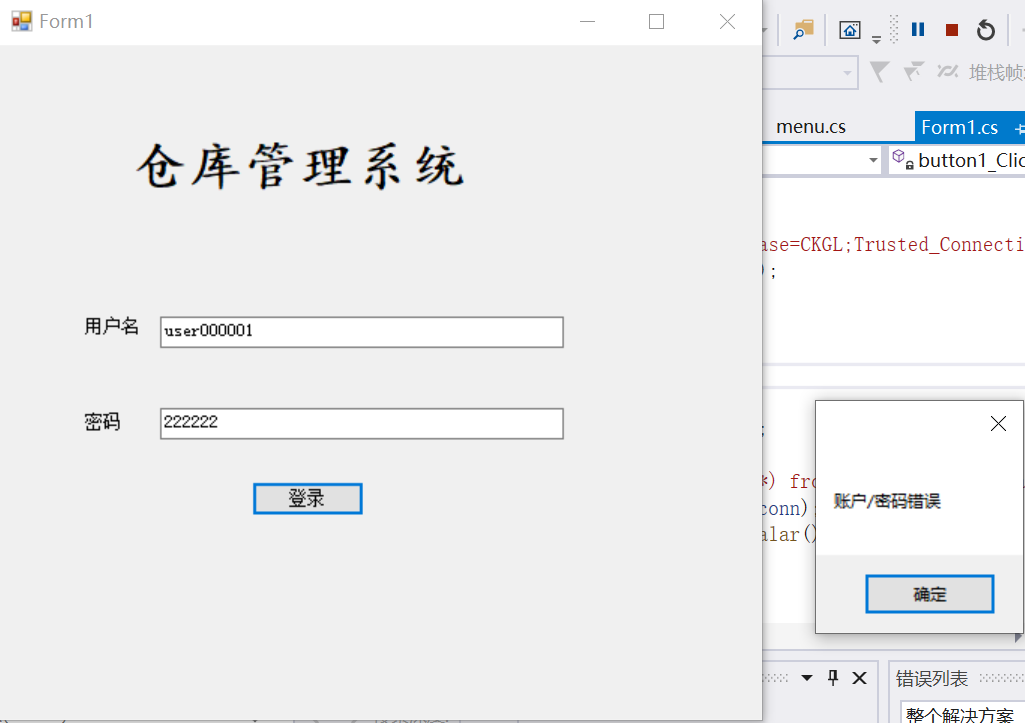


图5-2. 账户/密码输入错误的显示

### 菜单界面



5-3. 菜单界面

登录成功后就会打开菜单页面，菜单界面包括四个功能：货物查询、入仓记录管理、出仓记录管理、登录密码修改。点击对应的按钮就会跳转到相应的界面。

### 货物查询



图5-4. 货物查询界面

查询的时候需要输入查询的货物的编号，点击查询按钮，就会显示出该货物的基本信息：名称、进价、出价、剩余的数量。



图5-5. 货物查询实现效果

### 入仓信息管理

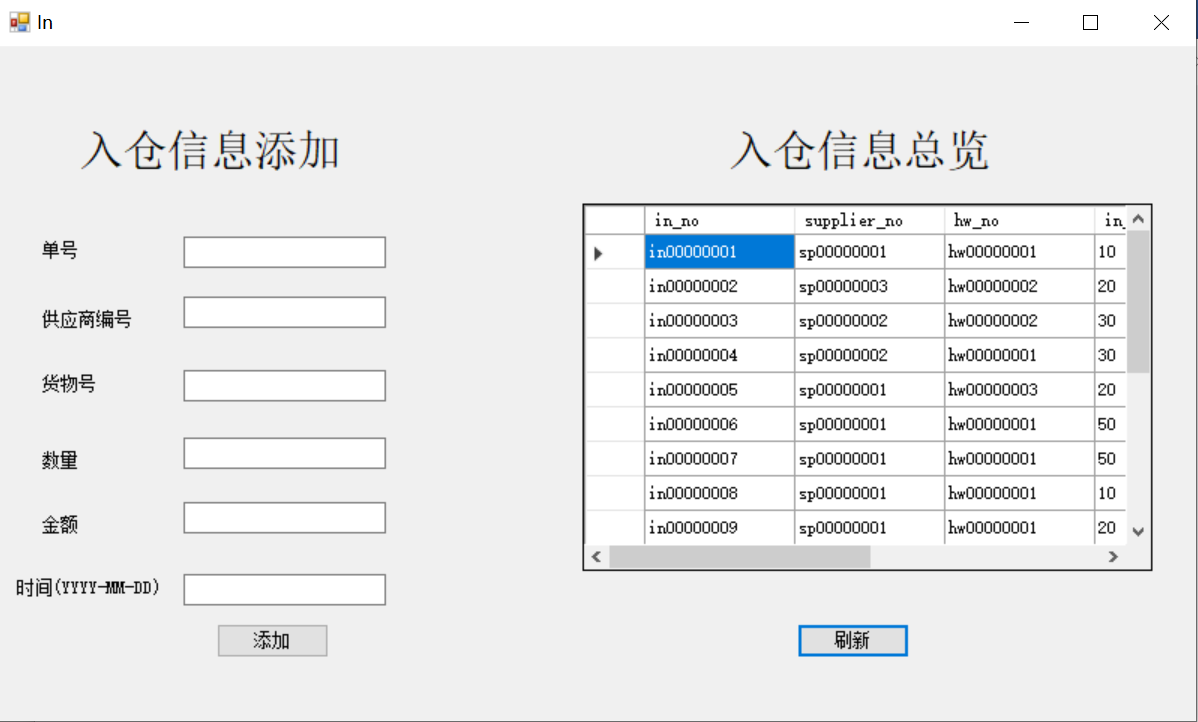


图5-6. 入仓信息管理界面

左侧部分时候入仓信息的添加，包括：单号、供应商编号、货物号、数量、金额和时间。右侧是入仓信息的一个总览，管理员输入入仓单号的信息之后，点击添加，对应的单号就会添加到数据库的Intable表中，点击右侧的刷新按钮，就能读取到最新的表格信息，由于Intable表中存在着入仓单号的主键约束，因此输入的单号不能重复，同时输入的其他信息格式也不能有误，否则会提示“ERROR” 信息。同时，由于入仓货物有一定的数量，因此也需要对HW表中的hw\_num进行更新。



图5-7. 入仓信息的添加

### 出仓信息管理

出仓信息管理的设计和入仓信息管理的设计基本一致。



图5-8. 出仓信息管理界面

出仓信息管理系统也存在着单号的主键约束，因此在输入的时候必须严格按照格式进行输入。由于出仓有对应的出仓量，因此还需要判断输入的出仓量是否大于当前该货物号的库存，如果出仓量大于库存，则提升库存不够，那么该出库操作是无法执行的。



图5-9．库存不足信息提示

### 密码修改



图5-10. 密码修改

输入账户、密码、新密码、重复新密码，单击修改密码，只要格式合法，就能修改密码。



图5-11. 修改成功

## 实现过程

### 登录界面代码

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CKGLDB

{

public partial class Form1 : Form

{

string user, pwd;

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

user = textBox1.Text;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

pwd = textBox2.Text;

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

string ConStr = "server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

if (conn.State == ConnectionState.Open)

{

if (user == "" || pwd == "") {

MessageBox.Show("密码和账户不能为空");

}

string sql = string.Format("select count(\*) from Managers where Manage\_no='{0}' and Manager\_psd='{1}'", user, pwd);//查询是否有该条记录，根据账户密码

SqlCommand command = new SqlCommand(sql, conn);

int i = Convert.ToInt32(command.ExecuteScalar());

if (i > 0)

{

Form me = new menu();

me.Show();

this.Hide();

}

else

{

MessageBox.Show("账户/密码错误");

}

}

else

{

MessageBox.Show("数据库连接失败");

}

}

catch

{

MessageBox.Show("ERROR");

}

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

}

}

### 主界面代码

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CKGLDB

{

public partial class menu : Form

{

public menu()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form que = new QueryHW();

que.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Form ins = new In();

ins.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Form outts = new outtable();

outts.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

}

}

}

### 货物查询代码

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CKGLDB

{

public partial class QueryHW : Form

{

public QueryHW()

{

InitializeComponent();

}

string hwid;

private void textBox1\_TextChanged(object sender, EventArgs e)

{

hwid = textBox1.Text;

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void QueryHW\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

string ConStr = "server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

if (conn.State == ConnectionState.Open)//连接到数据库

{

string sql = string.Format("select hw\_name from HW where hw\_no='{0}'",hwid);//查询货物名字

SqlCommand command = new SqlCommand(sql, conn);//执行操作

object obj = command.ExecuteScalar();;

if (obj != null)

{

this.textBox2.Text = obj.ToString();

}

else

{

MessageBox.Show("输入id有误！！");

}

obj = null;//清空

sql= string.Format("select hw\_inprice from HW where hw\_no='{0}'", hwid);//查询货物名字

command = new SqlCommand(sql, conn);//执行操作

obj = command.ExecuteScalar(); ;

if (obj != null)

{

this.textBox3.Text = obj.ToString();

}

else

{

MessageBox.Show("输入id有误！！");

}

obj = null;//清空

sql = string.Format("select hw\_outprice from HW where hw\_no='{0}'", hwid);//查询货物名字

command = new SqlCommand(sql, conn);//执行操作

obj = command.ExecuteScalar(); ;

if (obj != null)

{

this.textBox4.Text = obj.ToString();

}

else

{

MessageBox.Show("输入id有误！！");

}

obj = null;//清空

sql = string.Format("select hw\_num from HW where hw\_no='{0}'", hwid);//查询货物名字

command = new SqlCommand(sql, conn);//执行操作

obj = command.ExecuteScalar(); ;

if (obj != null)

{

this.textBox5.Text = obj.ToString();

}

else

{

MessageBox.Show("输入id有误！！");

}

conn.Close();

conn.Dispose();

}

else

{

MessageBox.Show("数据库连接失败");

}

}

catch

{

MessageBox.Show("ERROR");

}

}

}

}

### 入仓信息管理代码

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CKGLDB

{

public partial class In : Form

{

public In()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)//单号

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)//供应商编号

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)//货物号

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)//数量

{

}

private void In\_Load(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)//金额

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)//日期

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)//刷新

{

try

{

string ConStr = "server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

if (conn.State == ConnectionState.Open)

{

string sql = "select \* from Intable";

SqlCommand sc = new SqlCommand(sql, conn);

SqlDataAdapter sda = new SqlDataAdapter(sc);

DataSet ds = new DataSet();

sda.Fill(ds, "表名");

dataGridView1.DataSource = ds;

dataGridView1.DataMember = "表名";

conn.Close();

conn.Dispose();

}

else

{

MessageBox.Show("数据库连接失败");

}

}

catch

{

MessageBox.Show("ERROR");

}

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

string ConStr = "server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

if (conn.State == ConnectionState.Open)

{

string sql = "insert into Intable (in\_no,supplier\_no,hw\_no,in\_num,in\_money,in\_date) values('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "')";//编号

SqlCommand cmd = conn.CreateCommand();//创建数据库命令

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

int x = int.Parse(textBox4.Text);//增加的数字

sql = string.Format("Select hw\_num from HW where hw\_no='{0}'", textBox3.Text);

SqlCommand command = new SqlCommand(sql, conn);//执行操作

object obj = command.ExecuteScalar();

int now = (int)obj;

now += x;

//更新HW表

sql = string.Format("update HW set hw\_num = '{0}' where hw\_no = '{1}'",now.ToString(),textBox3.Text);

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

conn.Close();

conn.Dispose();

MessageBox.Show("添加成功");

}

else

{

MessageBox.Show("数据库连接失败");

}

}

catch

{

MessageBox.Show("ERROR");

}

}

}

}

### 出仓信息管理代码

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CKGLDB

{

public partial class outtable : Form

{

public outtable()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

string ConStr = "server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

if (conn.State == ConnectionState.Open)

{

int x = int.Parse(textBox4.Text);//减少的数字

//先查询还有多少库存now

string sql = string.Format("Select hw\_num from HW where hw\_no='{0}'", textBox3.Text);

SqlCommand command = new SqlCommand(sql, conn);//执行操作

object obj = command.ExecuteScalar();

int now = (int)obj;

if (now >= x)

{

sql = "insert into Outtable (out\_no,client\_no,hw\_no,out\_num,out\_money,out\_date) values('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "')";//编号

SqlCommand cmd = conn.CreateCommand();//创建数据库命令

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

now -= x;

//更新HW表

sql = string.Format("update HW set hw\_num = '{0}' where hw\_no = '{1}'", now.ToString(), textBox3.Text);

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

}

else

{

MessageBox.Show("库存不足");

}

conn.Close();

conn.Dispose();

MessageBox.Show("添加成功");

}

else

{

MessageBox.Show("数据库连接失败");

}

}

catch

{

MessageBox.Show("ERROR");

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

string ConStr = "server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

if (conn.State == ConnectionState.Open)

{

string sql = "select \* from Outtable";

SqlCommand sc = new SqlCommand(sql, conn);

SqlDataAdapter sda = new SqlDataAdapter(sc);

DataSet ds = new DataSet();

sda.Fill(ds, "表名");

dataGridView1.DataSource = ds;

dataGridView1.DataMember = "表名";

conn.Close();

conn.Dispose();

}

else

{

MessageBox.Show("数据库连接失败");

}

}

catch

{

MessageBox.Show("ERROR");

}

}

}

}

### 密码修改代码

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CKGLDB

{

public partial class changepwd : Form

{

public changepwd()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

string ConStr = "server=DESKTOP-PV2RSHT;database=CKGL;Trusted\_Connection=SSPI";

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

if (conn.State == ConnectionState.Open)

{

string user = textBox4.Text;

string oldpwd = textBox1.Text;

string n1pwd = textBox2.Text;

string n2pwd = textBox3.Text;

if (user == "" || oldpwd == "")

{

MessageBox.Show("密码和账户不能为空");

}

string sql = string.Format("select count(\*) from Managers where Manage\_no='{0}' and Manager\_psd='{1}'", user, oldpwd);//查询是否有该条记录，根据账户密码

SqlCommand command = new SqlCommand(sql, conn);

int i = Convert.ToInt32(command.ExecuteScalar());

if (i > 0)

{

if (n1pwd != n2pwd)

{

MessageBox.Show("输入不一致");

}

else

{

sql = string.Format("update Managers set Manager\_psd = '{0}' where Manage\_no = '{1}'", n1pwd, user);

SqlCommand cmd = conn.CreateCommand();

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

MessageBox.Show("修改成功");

}

}

}

else

{

MessageBox.Show("数据库连接失败");

}

}

catch

{

MessageBox.Show("ERROR");

}

}

}

}

# 总结

通过此次数据库课程设计，我不仅加深掌握了数据库的理论知识，还将这些理论知识应用到了课程设计中，同时我也掌握了C#对数据库的连接以及SQL命令在C#环境下的执行，在这个过程中，我通过在网上查找相关资料和技术，不断的学习并将其成功的实现出来，感觉很有成就感，这也是我未来的计算机发展之路的一个良好的铺垫，同时我也明白了对于某些学科的学习，不能仅仅纸上谈兵，而是需要多多动手实践，最后诚挚的感谢老师给了我这次能够从头到尾独立的设计并实现一个数据库系统的机会。