**《数据库与信息系统基础》**

**课程设计**

**学院** 环境与测绘学院 **专 业** 地理信息科学 **班级** 2019-1班

**学号** 07192329 **姓名** 武成龙 **指导教师** 张华

**环境与测绘学院**

**2021年 7月 30日**

目录

[目录 2](#_Toc10016)

[摘要 8](#_Toc28468)

[第一章 概述 9](#_Toc14787)

[第二章 需求分析 10](#_Toc17316)

[第三章 软件概要设计 16](#_Toc13194)

[第四章 数据库设计 19](#_Toc7339)

[第五章 详细设计 21](#_Toc21117)

[第六章 用户使用说明书 60](#_Toc19028)

[第七章 结束语 62](#_Toc4122)

**《数据库与信息系统基础》课程设计任务书**

**一、课程设计目的**

《数据库与信息系统基础》课程设计是《数据库与信息系统基础》课程的后续实践性课程，是一门独立开设的实践性课程，也是毕业设计等后续课程的基础。《数据库与信息系统基础》课程设计对于巩固数据库知识，加强学生的实际动手能力和提高学生综合素质十分必要。

《数据库与信息系统基础》课程设计的主要目标是：

1、加深对数据库系统原理及程序设计的理论知识的理解和应用。

2、通过设计实际的数据库系统课题，进一步熟悉数据库管理系统的操作技术，提高动手能力，提高分析问题和解决问题的能力。

3、学习基本的数据库编程方法。掌握利用某种语言环境进行程序设计与调试的方法。（注：软件开发语言为C#，后台数据库为SQL server等）。

**二、选题依据**

本课程设计是对《数据库与信息系统基础》课堂教学的一个重要补充，与理论学习起着相辅相成的作用，是实施《数据库与信息系统基础》教学的一个重要组成部分。通过本课程设计，

**三、课程设计的题目要求**

能体现《数据库与信息系统基础》基本内容。

**四、课程设计的任务**

背景资料：

无论是一个销售公司还是一个销售商店，都有很多的销售数据需要管理。手工管理这些繁琐复杂的数据方式已经不适应时代的潮流，实现数据的规范化、自动化的计算机管理，是一个管理规范、运作高效的企事业单位的必然要求。本课程设计就是为了实现给销售单位高效管理而设计的。

本系统主要考虑了最为普遍的销售管理需求：

（1）进货时入库登记，以便日后查看和统计进货情况；

（2）销售时把相应数量的商品从现存商品库中减去，并记录于另外的一个销售库中，以便日后查看和统计销售信息；

（3）是销售就难免会有顾客退货情况发生，因此还需建立一个相应的退货管理，把退回来的货加入现存商品库，再从销售库中减去，进行退货登记。

（4）还要能够随时查看各种信息表，如员工信息、进货信息、销售信息、退货信息。

（5）除了以上一些基本的需求外，实际中公司用来进行决策和管理分析的数据才是用得最多的数据。因此还需要有一些相应的关于以上数据的统计信息。如希望查看各个时间段（如今日、本月、本年度等）的进货和销售情况，以便对以后的运作进行一个准确的预测和正确的决断。

（6）为了调动员工的积极性，员工的薪水应该和他的销售情况挂钩的，因此还需要能够查看各员工的销售情况，以便对他们的业绩进行考核和评比。

以上数据如果用传统的方法管理，则显得繁琐且容易出错，透明度也不高，容易使人有机可乘，造成公司账目混乱。

本设计正是针对一般销售公司都有的以上需求而设计的，可以有效管理数据。主要功能有

交易管理，包括进货登记管理、销售登记管理、退货登记管理；进货统计管理，其中包括有关进货的各种统计数据表，如今日进货统计、本月进货统计、本季度进货统计、本年度销售统计等：销售统计管理，其中包括有关销售的各种统计数据表，如今日销售统计、本月销售统计、本季度销售统计、本年度销售统计等；员工销售业绩的考核，可以很方便地查看各员工的销售情况，以便对各员工的业绩进行考核比较。

2、内容：学生根据所学的数据库原理与程序设计的知识，针对销售管理信息系统，进行系统的需求分析，系统设计，数据库设计，编码，测试等，完成题目要求的功能，从而达到掌握开发一个小型数据库的目的。

3、操作过程：

1）进行系统调查，搜集资料。

2）进行系统需求分析和系统设计，写出系统分析和系统设计报告。

3）设计数据库模型并进行优化，确定数据库结构、功能结构和系统安全性和完整性要求。

3）完成数据库定义工作，实现系统数据的数据录入和数据处理。

4）实现应用程序的设计、编程、优化功能，实现数据安全性、数据完整性和并发控制技术等功能，并针对具体课题问题提出解决方法。

5）对系统的各个应用程序进行集成调试和测试，进一步优化系统性能，改善系统用户界面。

**五、设计技术要求**

本设计总体的系统功能是实现销售公司销售管理的系统化、的功能有：规范化和自动化。

（1）系统登录：包括用户登录、添加用户、修改密码、退出系统等。

（2）交易管理：包括进货登记、销售登记、退货登记等。其中进货登记中还包含进货厂商的登记，如果是新的进货厂商，可以在此登记入库。进货和销售的主要交易都是在此进行：每次交易(包括进货、销售和退货)都需要对商品名、生产厂商、商品型号、单价、数量、总金额、交易日期、经手业务员编号等各项进行记录。

（3）进货统计：包括今日进货统计、本月进货统计、本季度进货统计、本年度进货统计等。其中每次进货统计都是按进货厂商和进货金额顺序排列的。使用一个表格显示所选时间段的全部进货数据，包括进货编号、商品名、生产厂商、产品型号、单价、数量、总金额、进货日期、业务员编号等；再使用另一表格显示分别从各个厂商进货的金额，还有一个文本框用于显示此时间段的总进货金额。

（4）销售统计：包括今日销售统计、本月销售统计、本季度销售统计、本年度销售统计等。其中每次销售统计都是按产品厂商和销售金额顺序排列的，用户可以一目了然地看出各种型号产品的销售额。使用一个表格显示所选时间段的全部销售数据，包括销售编号、商品名、生产厂商、产品型号、单价、数量、总金额、销售日期、业务员编号等；再使用另一表格分别显示各个厂商产品的销售金额，还有一个文本框用于显示此时间段的总销售金额。

（5）业绩查看：实现各员工销售情况的统计，可以一次性显示所有员工的销售情况，也可以根据员工编号查看各个员工的销售情况。

（6）查看数据表：实现各种数据表的浏览，包括进货表、销售表、退货表、员工表、进货商表。浏览时此数据表的所有信息项全部显示。如果是员工信息或进货商信息有变化，还可以对员工表和进货商表进行修改，其他的表则不能进行修改，防止有人为造假数据。

以上的数据库、表结构设计、菜单设计、窗体设计可参考附件《数据库与信息系统基础课程设计参考》，但编程语言必须用C#语言。

**六、课程设计报告要求**

1、设计报告应包括封面、任务书、目录、摘要及关键字、正文、参考文献(资料)等内容，以及附图或附件等材料。

2、题目字体用小三，黑体，正文字体用五号字，宋体，小标题用四号及小四，宋体，并用A4纸打印。

3、页眉写上数据库课程设计报告，页脚写上页码。

4、报告具体格式如下：

1)封面

见附1。

2) 任务书

3) 目录

目录页中每行均由标题名称和页码组成。

中英文摘要

扼要叙述本设计的主要内容、特点，文字要精练。英文摘要与中文摘要对应。

5) 正文

第一章 概述

包括项目背景、编写目的、软件定义、开发环境等内容。

需求分析

陈述用户的功能、性能等的需求，画出数据流图，给出数据字典等。

第三章 软件概要设计

给出系统目标、总体设计、运行设计等，画出软件功能图，描述每一个功能所完成的任务。

第四章 数据库设计

包括：数据库概念设计（画出E－R图）

数据库逻辑设计（把E－R图转换为关系模型并进行优化，要求所有关系达到3NF，定义视图、索引、主关键字、权限等。）

详细设计

说明每个功能模块具体是怎么实现的。

用户使用说明书。

结束语

写出完成本课程设计的心得，领会数据库理论与软件开发实践的关系，有哪些收获。介绍整个系统中自己认为最满意、最得意的地方，介绍应用程序中重点创意的技术实现技巧、核心程序等。指出软件还需要哪些改进。

6) 参考文献

参考文献只选用最主要的列入，未公开发表的资料或协作成果，应征得有关方面的同意，以脚注方式顺序标明。

# 摘要

数据库的使用能使得数据管理更加流畅，数据走向更加清晰，无论是大公司还是个体户都需要使用数据库来管理相关数据。数据库是建立在数理逻辑和集合操作基础上的，具有高效、可靠、完整、自同步等特性，是业务系统进行数据控制的最佳选择。本次课程设计实验尝试使用C#制作GUI界面实现数据库的管理和维护可视化，内部连接SQL sever，使用SQL语句来实现对数据库的数据操纵，最终目的是让管理人员在管理数据库时大体上实现数据一键入库，一键出库的操作，更方便地管理数据文件。程序设计上尽量做到简明，无论是代码还是界面，尽力实现可读性高、上手轻松的特点。

关键词：C# SQL sever 销售管理

Summary

The use of database can make data management more smooth and data trend more clear. Both large companies and individual businesses need to use database to manage relevant data. Database is based on mathematical logic and set operation. It has the characteristics of high efficiency, reliability, integrity and self synchronization. It is the best choice for business system to control data. This course design experiment attempts to use c# to make the GUI interface to realize the visualization of database management and maintenance, internally connect SQL Sever, and use SQL statements to realize the data manipulation of the database. The ultimate purpose is to enable managers to generally realize the operation of data one key in and one key out when managing the database, so as to manage data files more conveniently. The program design shall be concise as far as possible. Whether it is code or interface, try to realize the characteristics of high readability and easy to use.

Keywords: C# SQL sever Sales Management

# 概述

## 项目背景

无论是一个销售公司还是一个销售商店，每天都有大量的销售数据需要管理。手工管理这些繁琐复杂的数据方式已经不适应时代的潮流，实现数据的规范化、自动化的计算机管理，是一个管理规范、运作高效的企事业单位的必然要求。本课程设计就是为了销售单位高效管理而设计的。

## 编写目的

《数据库与信息系统基础》课程设计是《数据库与信息系统基础》课程的后续实践性课程，是一门独立开设的实践性课程，也是毕业设计等后续课程的基础。《数据库与信息系统基础》课程设计对于巩固数据库知识，加强学生的实际动手能力和提高学生综合素质十分必要。

《数据库与信息系统基础》课程设计的主要目标是：

1、加深对数据库系统原理及程序设计的理论知识的理解和应用。

2、通过设计实际的数据库系统课题，进一步熟悉数据库管理系统的操作技术，提高动手能力，提高分析问题和解决问题的能力。

3、学习基本的数据库编程方法。掌握利用某种语言环境进行程序设计与调试的方法。

## 三、软件定义

该软件是使用Visual Studio 2019，基于C #开发的一款轻量小型的数据库管理软件，主要面向的是小中型企业的管理，目的是提高管理效率。

## 开发环境

1. Windows 10系统
2. Microsoft SQL Sever Management Studio 18
3. Visual Studio 2019

# 需求分析

## 性能需求

本系统主要考虑了最为普遍的销售管理需求：

1、进货时**入库登记和厂商登记**，以便日后查看和统计进货情况；

2、销售时把相应数量的商品从现存商品库中减去，进行销售登记，记录于另外的一个销售库中，以便日后查看和统计销售信息；

3、是销售就难免会有顾客退货情况发生，因此还需建立一个相应的退货管理，把退回来的货加入现存商品库，再从销售库中减去，进行退货登记。

4、还要能够随时查看各种信息表，如员工信息、进货信息、销售信息、退货信息。

5、除了以上一些基本的需求外，实际中公司用来进行决策和管理分析的数据才是用得最多的数据。因此还需要有一些相应的关于以上数据的统计信息。如希望查看各个时间段（如今日、本月、本年度等）的进货和销售情况，以便对以后的运作进行一个准确的预测和正确的决断。

6、为了调动员工的积极性，员工的薪水应该和他的销售情况挂钩的，因此还需要能够查看各员工的销售情况，以便对他们的业绩进行考核和评比。

以上数据如果用传统的方法管理，则显得繁琐且容易出错，透明度也不高，容易使人有机可乘，造成公司账目混乱。本设计正是针对一般销售公司都有的以上需求而设计的，可以有效管理数据。

## 主要功能

用户登录，能让用户使用用户名和密码进行登录，也支持注册、修改密码的操作；

交易管理，包括进货登记管理、销售登记管理、退货登记管理；

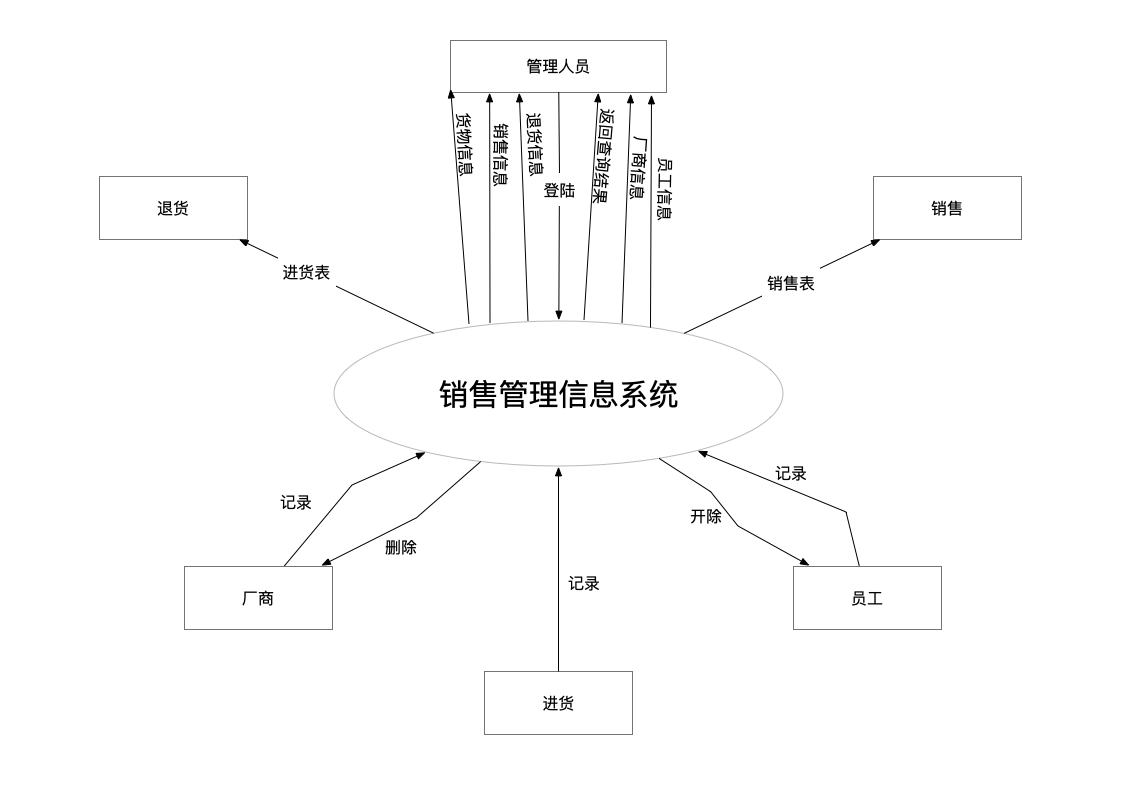
进货统计管理，其中包括有关进货的各种统计数据表，如今日进货统计、本月进货统计、本季度进货统计、本年度销售统计等；

销售统计管理，其中包括有关销售的各种统计数据表，如今日销售统计、本月销售统计、本季度销售统计、本年度销售统计等；

员工销售业绩的考核，可以很方便地查看各员工的销售情况，以便对各员工的业绩进行考核比较；

数据表查看，能快速查看各项数据表的内部情况，并对各别特定数据库直接能够进行操作，如删除或修改。

## 数据流图



## 数据字典

1、数据结构描述

（1）名称：manufacturer

含义说明：厂商信息

组成结构：厂商编号+厂商名称+法人代表+电话+厂商地址

（2）名称：goods

含义说明：进货信息

组成结构：商品编号+生产厂商+商品名+型号+单价+数量+总金额+进货年+进货月+进货日+业务员编号

（3）名称：sell

含义说明：销售信息

组成结构：商品编号+生产厂商+商品名+型号+单价+数量+总金额+销售年+销售月+销售日+业务员编号

（4）名称：employee

含义说明：员工信息

组成结构：员工编号+员工姓名+员工电话+员工地址

（5）名称：retreat

含义说明：退货信息

组成结构：商品编号+生产厂商+商品名+型号+单价+数量+总金额+退货年+退货月+退货日+业务员编号

2、数据流的描述

（1）数据流名称：进货登记

简述：登记所进货物信息

数据流来源：管理员

数据流去向：goods，manufacturer

数据流组成：商品编号+生产厂商+商品名+型号+单价+数量+总金额+进货年+进货月+进货日+业务员编号+厂商编号+厂商名称+法人代表+电话+厂商地址

（2）数据流名称：销售登记

简述：登记销售货物信息

数据流来源：goods

数据流去向：sell

数据流组成：商品编号+生产厂商+商品名+型号+单价+数量+总金额+销售年+销售月+销售日+业务员编号

（3）数据流名称：退货登记

简述：登记所退货物信息

数据流来源：sell

数据流去向：retreat

数据流组成：商品编号+生产厂商+商品名+型号+单价+数量+总金额+退货年+退货月+退货日+业务员编号

（4）数据流名称：进货统计

简述：统计进货的信息

数据流来源：goods

数据流去向：goods

数据流组成：商品编号+生产厂商+商品名+型号+单价+数量+总金额+进货年+进货月+进货日

（5）数据流名称：销售统计

简述：统计销售的信息

数据流来源：sell

数据流去向：sell

数据流组成：商品编号+生产厂商+商品名+型号+单价+数量+总金额+销售年+销售月+销售日+业务员编号

（6）数据流名称：业绩查询

简述：查询员工的销售额

数据流来源：employee，sell

数据流去向：employee，sell

数据流组成：商品编号+生产厂商+商品名+型号+单价+数量+总金额+销售年+销售月+销售日+业务员编号+员工编号+员工姓名+员工电话+员工地址

3.主要数据存储的定义

（1）存储名称：现存货表

说明：查询存货信息

输入：goods

输出：all goods

数据结构：商品编号+生产厂商+商品名+型号+单价+数量+总金额+进货年+进货月+进货日+业务员编号

（2）存储名称：已售商品表

说明：查询已售商品的信息

输入：sell

输出：all sell

数据结构：商品编号+生产厂商+商品名+型号+单价+数量+总金额+销售年+销售月+销售日+业务员编号

（3）存储名称：退货表

说明：查询退货表的信息

输入：retreat

输出：all retreat

数据结构：商品编号+生产厂商+商品名+型号+单价+数量+总金额+退货年+退货月+退货日+业务员编号

（4）存储名称：员工表

说明：查询员工信息

输入：employee

输出：all employee

数据结构：员工编号+员工姓名+员工电话+员工地址

（5）存储名称：进货商表

说明：查询进货商的信息

输入：manufacturer

输出：all manufacturer

数据结构：厂商编号+厂商名称+法人代表+电话+厂商地址

1. 数据表项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sellSystem | employee | 员工编号 | int(255) |  |
| sellSystem | employee | 员工姓名 | varchar(10) |  |
| sellSystem | employee | 员工电话 | varchar(20) |  |
| sellSystem | employee | 员工地址 | varchar(100) |  |
| sellSystem | goods | 商品编号 | int(255) |  |
| sellSystem | goods | 生产厂商 | varchar(20) |  |
| sellSystem | goods | 商品名 | varchar(20) |  |
| sellSystem | goods | 型号 | varchar(20) |  |
| sellSystem | goods | 单价 | double(255,0) |  |
| sellSystem | goods | 数量 | decimal(10,0) |  |
| sellSystem | goods | 总金额 | double(255,0) |  |
| sellSystem | goods | 进货年 | smallint(255) |  |
| sellSystem | goods | 进货月 | smallint(255) |  |
| sellSystem | goods | 进货日 | smallint(255) |  |
| sellSystem | goods | 业务员编号 | int(255) |  |
| sellSystem | manufacturer | 厂商编号 | int(255) |  |
| sellSystem | manufacturer | 厂商名称 | varchar(20) |  |
| sellSystem | manufacturer | 法人代表 | varchar(10) |  |
| sellSystem | manufacturer | 厂商地址 | varchar(100) |  |
| sellSystem | manufacturer | 电话 | varchar(20) |  |
| sellSystem | retreat | 退货编号 | int(255) |  |
| sellSystem | retreat | 厂商 | varchar(20) |  |
| sellSystem | retreat | 商品名 | varchar(20) |  |
| sellSystem | retreat | 型号 | varchar(20) |  |
| sellSystem | retreat | 单价 | double(255,0) |  |
| sellSystem | retreat | 数量 | int(255) |  |
| sellSystem | retreat | 总金额 | double(255,0) |  |
| sellSystem | retreat | 退货年 | smallint(255) |  |
| sellSystem | retreat | 退货月 | smallint(255) |  |
| sellSystem | retreat | 退货日 | smallint(255) |  |
| sellSystem | retreat | 业务员编号 | int(255) |  |
| sellSystem | sell | 商品编号 | int(255) |  |
| sellSystem | sell | 生产厂商 | varchar(20) |  |
| sellSystem | sell | 商品名 | varchar(50) |  |
| sellSystem | sell | 型号 | varchar(50) |  |
| sellSystem | sell | 单价 | double(255,0) |  |
| sellSystem | sell | 数量 | int(255) |  |
| sellSystem | sell | 总金额 | double(255,0) |  |
| sellSystem | sell | 销售年 | smallint(255) |  |
| sellSystem | sell | 销售月 | smallint(255) |  |
| sellSystem | sell | 销售日 | smallint(255) |  |
| sellSystem | sell | 业务员编号 | int(255) |  |
| sellSystem | userdb | 用户编号 | int(255) |  |
| sellSystem | userdb | 用户名 | varchar(10) |  |
| sellSystem | userdb | 密码 | varchar(10) |  |

# 软件概要设计

## 一、系统目标

1、系统登录：包括用户登录、添加用户、修改密码、退出系统等。

2、交易管理：包括进货登记、销售登记、退货登记等。其中进货登记中还包含进货厂商的登记，如果是新的进货厂商，可以在此登记入库。进货和销售的主要交易都是在此进行：每次交易(包括进货、销售和退货)都需要对商品名、生产厂商、商品型号、单价、数量、总金额、交易日期、经手业务员编号等各项进行记录。

3、进货统计：包括今日进货统计、本月进货统计、本季度进货统计、本年度进货统计等。其中每次进货统计都是按进货厂商和进货金额顺序排列的。使用一个表格显示所选时间段的全部进货数据，包括进货编号、商品名、生产厂商、产品型号、单价、数量、总金额、进货日期、业务员编号等；再使用另一表格显示分别从各个厂商进货的金额，还有一个文本框用于显示此时间段的总进货金额。

4、销售统计：包括今日销售统计、本月销售统计、本季度销售统计、本年度销售统计等。其中每次销售统计都是按产品厂商和销售金额顺序排列的，用户可以一目了然地看出各种型号产品的销售额。使用一个表格显示所选时间段的全部销售数据，包括销售编号、商品名、生产厂商、产品型号、单价、数量、总金额、销售日期、业务员编号等；再使用另一表格分别显示各个厂商产品的销售金额，还有一个文本框用于显示此时间段的总销售金额。

5、业绩查看：实现各员工销售情况的统计，可以一次性显示所有员工的销售情况，也可以根据员工编号查看各个员工的销售情况。

6、查看数据表：实现各种数据表的浏览，包括进货表、销售表、退货表、员工表、进货商表。浏览时此数据表的所有信息项全部显示。如果是员工信息或进货商信息有变化，还可以对员工表和进货商表进行修改，其他的表则不能进行修改，防止有人为造假数据。

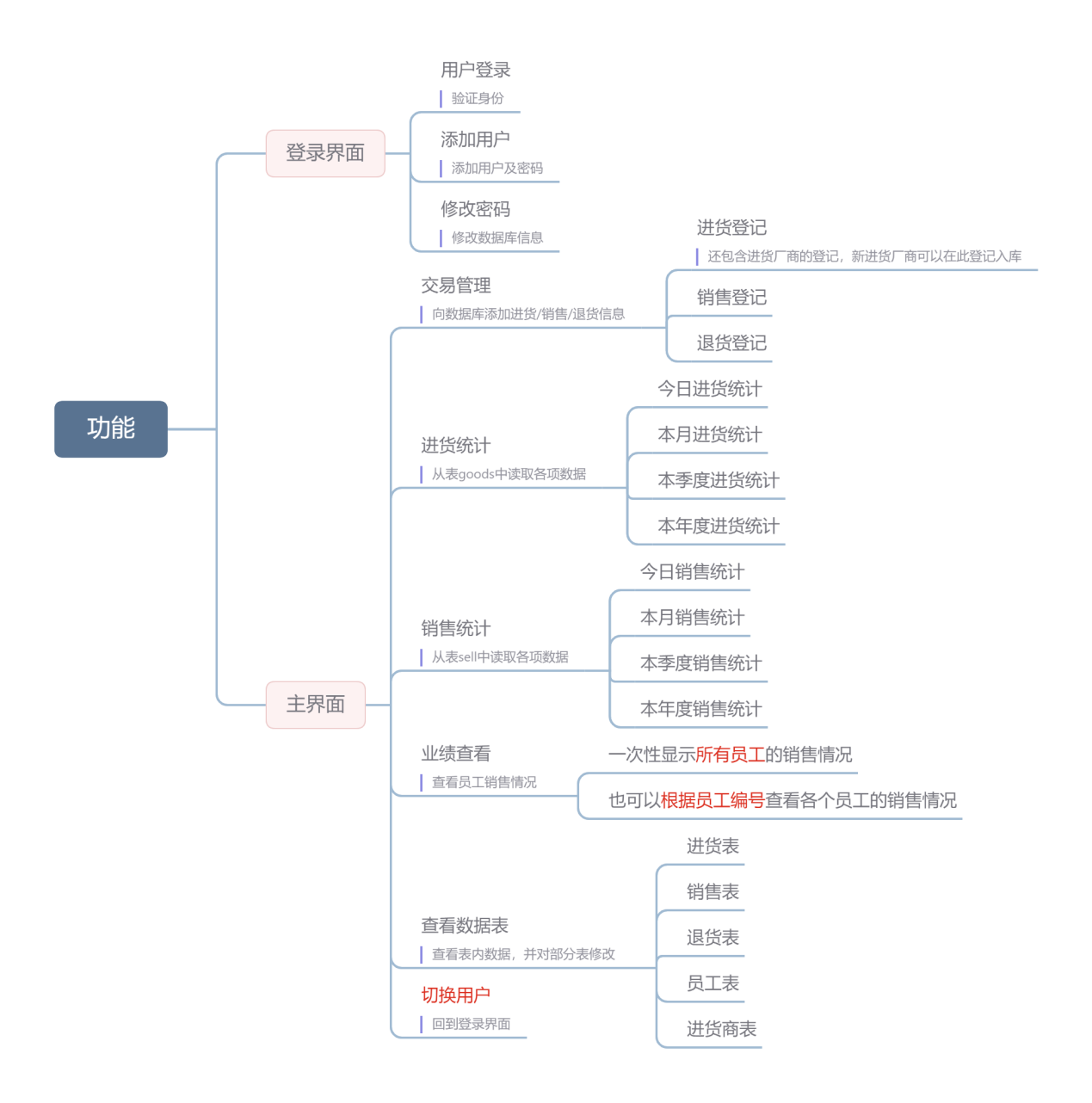
## 二、总体设计

|  |  |  |  |
| --- | --- | --- | --- |
| 销售管理信息系统 | | | |
| 登录界面 | | 登录  添加用户  修改密码 | |
| 主界面 | 交易管理 | 进货登记 | 添加 |
| 销售登记 |
| 退货登记 |
| 进货统计 | 今日进货量 | |
| 本月进货量 | |
| 本季度进货量 | |
| 今年进货量 | |
| 销售统计 | 今日销售量 | |
| 本月销售量 | |
| 本季度销售量 | |
| 今年销售量 | |
| 业绩查看 | 业绩查看 | 查询 |
| 查看数据表 | 现存货表 | |
| 已售商品表 | |
| 退货表 | |
| 员工表 | 修改  删除 |
| 进货商表 |
| 切换用户 | 切换用户 | |

## 运行设计

目标程序使用VS2019编译器运行，软件设计应当表现出层次结构,它应利用各个软件部件之间的控制关系。设计应当是模块化的,即该软件应当从逻辑上被划分成多个部件,分别实现各种特定功能和子功能。设计最终应当给出具体的模块(例如子程序或过程),这些模块就具有独立的功能特性。应当应用在软件需求分析期间得到的信息,采取循环的方法来获得设计。

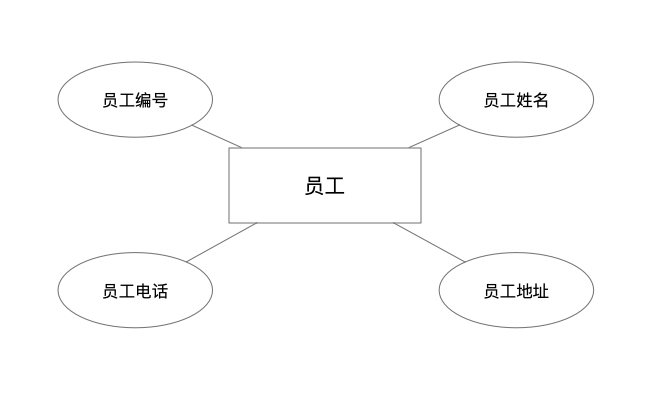
## 软件功能图



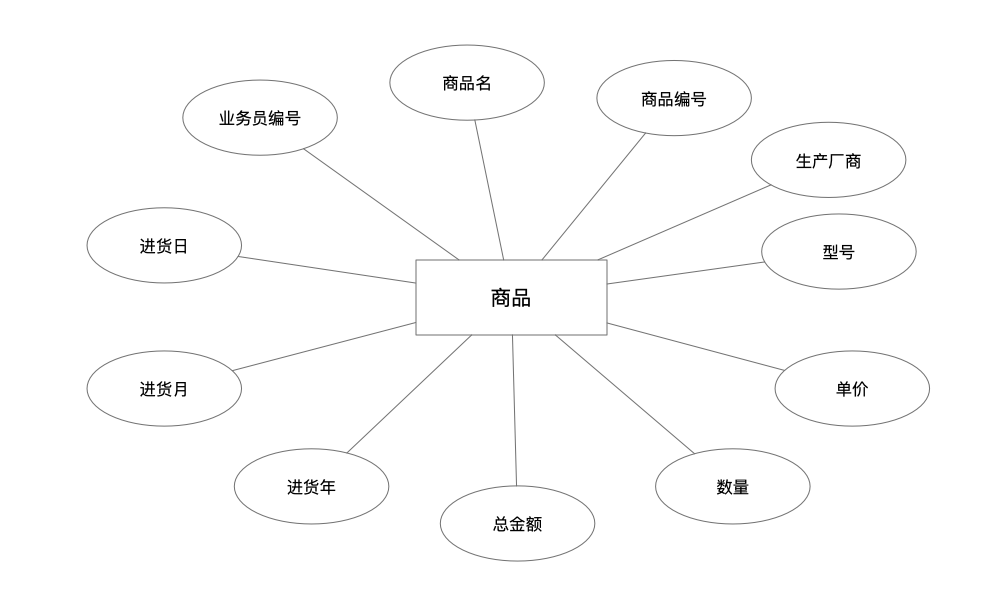
# 数据库设计

## 数据库概念设计

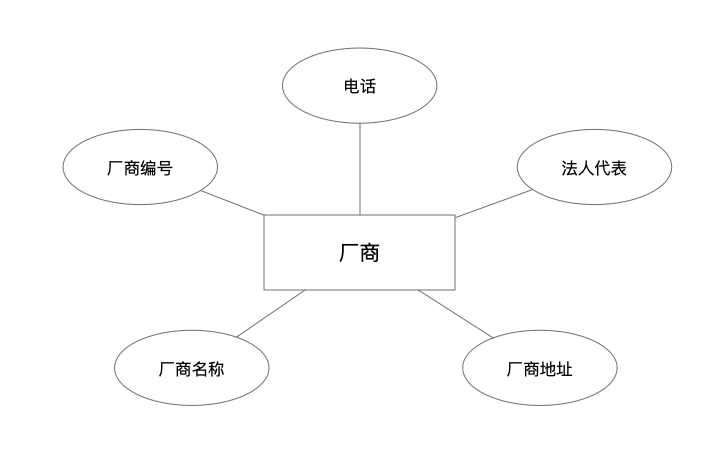
### 员工E-R图



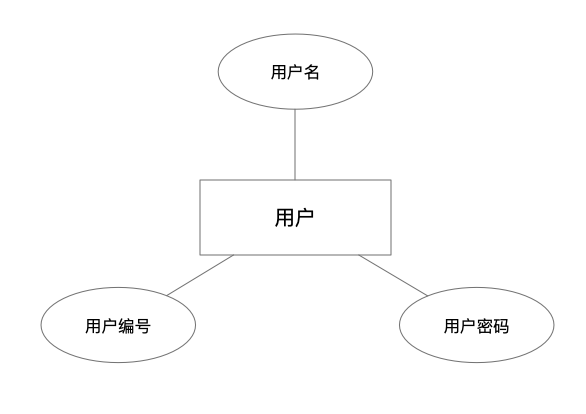
### 2.商品E-R图



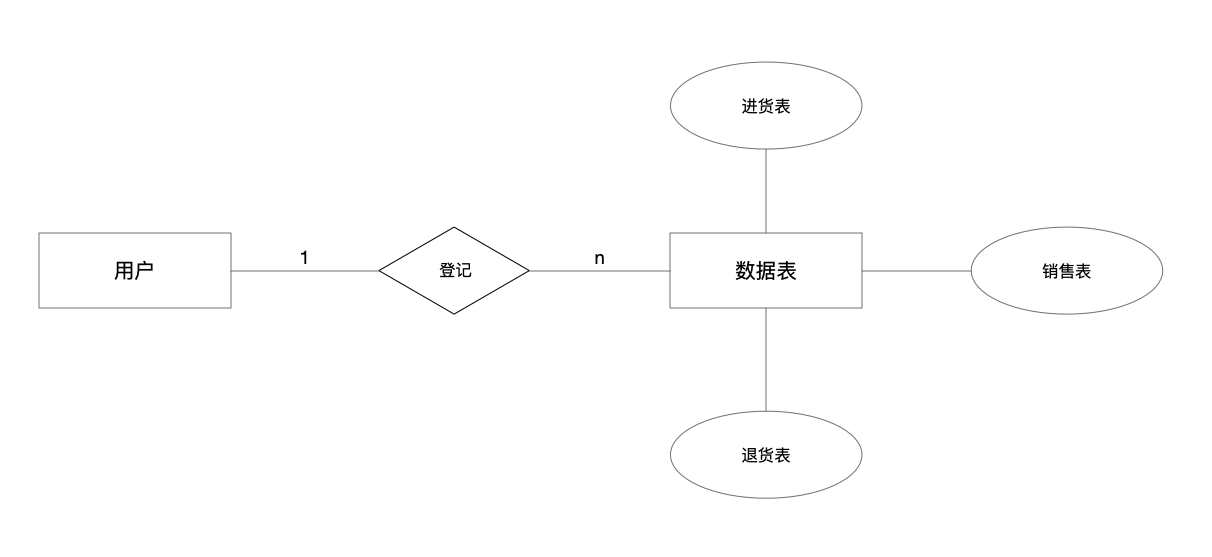
### 3.厂商E-R图



1. **用户E-R图**

****

1. **登记查询E-R图**

****

## **数据库逻辑设计**

1、关系模式：

员工信息（员工编号、员工姓名、员工电话、员工地址）

商品信息（业务员编号、商品编号、生产厂商、商品名、型号、单价、数量、总金额、进货年月日、销售年月日、退货年月日）

厂商信息（厂商编号、厂商名称、法人代表、电话、厂商地址）

登记查询（业务员编号、商品编号、生产厂商、商品名、型号、单价、数量、总金额、进货年月日、销售年月日、退货年月日）

2、优化为3NF ：

员工信息（员工编号、员工姓名、员工电话、员工地址）

商品属性（商品编号、生产厂商、商品名、型号、单价、数量、总金额）

商品信息（商品编号、业务员编号、进货年月日、销售年月日、退货年月日）

厂商信息（厂商编号、厂商名称、法人代表、电话、厂商地址）

登记查询（业务员编号、商品编号）

关键字：员工编号，商品编号，厂商编号，业务员编号

# 详细设计

## 建立数据库

根据《数据库与信息系统基础》课程设计任务书创建数据库文件sellSystem，该数据库中包含6个表，分别是employee表、goods表、sell表、retreat表、manufacturer表、userdb表。

1. 建立数据库sellSystem

CREATE DATABASE sellSystem

1. 建立employee表

CREATE TABLE `employee` (

`员工编号` int(255) NOT NULL AUTO\_INCREMENT,

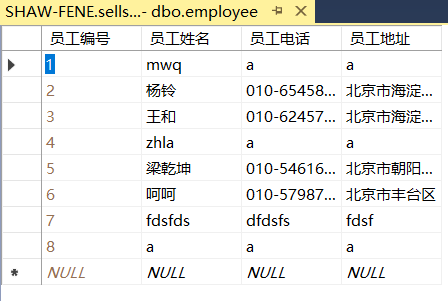
`员工姓名` varchar(10) DEFAULT NULL,

`员工电话` varchar(20) DEFAULT NULL,

`员工地址` varchar(100) DEFAULT NULL,

PRIMARY KEY (`员工编号`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;



1. 建立manufacture表

CREATE TABLE `manufacturer` (

`厂商编号` int(255) NOT NULL AUTO\_INCREMENT,

`厂商名称` varchar(20) NOT NULL,

`法人代表` varchar(10) DEFAULT NULL,

`厂商地址` varchar(100) DEFAULT NULL,

`电话` varchar(20) DEFAULT NULL,

PRIMARY KEY (`厂商编号`),

UNIQUE KEY `u\_cm` (`厂商名称`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;



1. 建立sell表

CREATE TABLE `sell` (

`商品编号` int(255) NOT NULL AUTO\_INCREMENT,

`生产厂商` varchar(20) NOT NULL,

`商品名` varchar(50) DEFAULT NULL,

`型号` varchar(50) DEFAULT NULL,

`单价` double(255,0) DEFAULT '0',

`数量` int(255) DEFAULT '0',

`总金额` double(255,0) DEFAULT '0',

`销售年` smallint(255) DEFAULT (date\_format(now(),\_utf8mb4'%Y')),

`销售月` smallint(255) DEFAULT (date\_format(now(),\_utf8mb4'%m')),

`销售日` smallint(255) DEFAULT (date\_format(now(),\_utf8mb4'%d')),

`业务员编号` int(255) DEFAULT NULL,

PRIMARY KEY (`商品编号`),

KEY `FK\_sell\_em` (`业务员编号`),

KEY `FK\_sell\_ma` (`生产厂商`),

CONSTRAINT `FK\_sell\_em` FOREIGN KEY (`业务员编号`) REFERENCES `employee` (`员工编号`),

CONSTRAINT `FK\_sell\_ma` FOREIGN KEY (`生产厂商`) REFERENCES `manufacturer` (`厂商名称`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;



1. 建立retreat表

CREATE TABLE `retreat` (

`退货编号` int(255) NOT NULL AUTO\_INCREMENT,

`厂商` varchar(20) DEFAULT NULL,

`商品名` varchar(20) DEFAULT NULL,

`型号` varchar(20) DEFAULT NULL,

`单价` double(255,0) DEFAULT '0',

`数量` int(255) DEFAULT '0',

`总金额` double(255,0) DEFAULT '0',

`退货年` smallint(255) DEFAULT NULL,

`退货月` smallint(255) DEFAULT NULL,

`退货日` smallint(255) DEFAULT NULL,

`业务员编号` int(255) DEFAULT NULL,

PRIMARY KEY (`退货编号`),

KEY `FK\_retreat\_em` (`业务员编号`),

KEY `FK\_retreat\_ma` (`厂商`),

CONSTRAINT `FK\_retreat\_em` FOREIGN KEY (`业务员编号`) REFERENCES `employee` (`员工编号`),

CONSTRAINT `FK\_retreat\_ma` FOREIGN KEY (`厂商`) REFERENCES `manufacturer` (`厂商名称`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;



1. 建立goods表

CREATE TABLE `goods` (

`商品编号` int(255) NOT NULL AUTO\_INCREMENT,

`生产厂商` varchar(20) DEFAULT NULL,

`商品名` varchar(20) DEFAULT NULL,

`型号` varchar(20) DEFAULT NULL,

`单价` double(255,0) DEFAULT '0',

`数量` decimal(10,0) DEFAULT '0',

`总金额` double(255,0) DEFAULT NULL,

`进货年` smallint(255) DEFAULT (date\_format(now(),\_utf8mb4'%Y')),

`进货月` smallint(255) DEFAULT (date\_format(now(),\_utf8mb4'%m')),

`进货日` smallint(255) DEFAULT (date\_format(now(),\_utf8mb4'%d')),

`业务员编号` int(255) DEFAULT NULL,

PRIMARY KEY (`商品编号`),

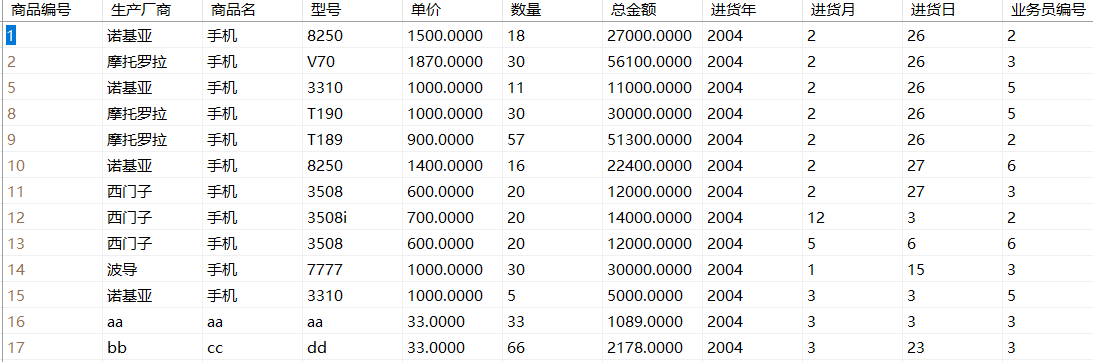
KEY `FK\_goods\_em` (`业务员编号`),

KEY `FK\_goods\_ma` (`生产厂商`),

CONSTRAINT `FK\_goods\_em` FOREIGN KEY (`业务员编号`) REFERENCES `employee` (`员工编号`),

CONSTRAINT `FK\_goods\_ma` FOREIGN KEY (`生产厂商`) REFERENCES `manufacturer` (`厂商名称`)

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;



7.建立userdb表

CREATE TABLE `userdb` (

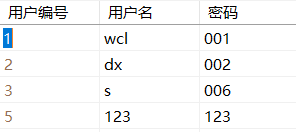
`用户编号` int(255) NOT NULL AUTO\_INCREMENT,

`用户名` varchar(10) DEFAULT NULL,

`密码` varchar(10) DEFAULT NULL,

PRIMARY KEY (`用户编号`)

) ENGINE=InnoDB AUTO\_INCREMENT=22 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;



## 应用程序设计

1. 登陆界面和主界面设计



登陆界面



管理员界面

功能：

鼠标左键单击“确认”按钮时，判断用户名和登录密码是否正确，不正确则提示错误，正确则成功登录，弹出管理员界面；鼠标左键单击“添加用户”按钮时，弹出注册窗口。单击“修改密码”时，弹出修改密码窗口。

代码：

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

{

if (textBox1.Text != "" && textBox2.Text != "")

{

login();//不为空则尝试登录

}

else

{

MessageBox.Show("输入不完全，请重新输入");

}

}

//登录方法

public void login()

{

Dao dao = new Dao();

string sql = "select \* from userdb where 用户名='" + textBox1.Text + "' and 密码='" + textBox2.Text + "'";

IDataReader dc = dao.Read(sql);

if (dc.Read())

{

MessageBox.Show("登录成功");

//打开管理员窗体

Admin a = new Admin();

this.Hide();

a.ShowDialog();

this.Show();

}

else

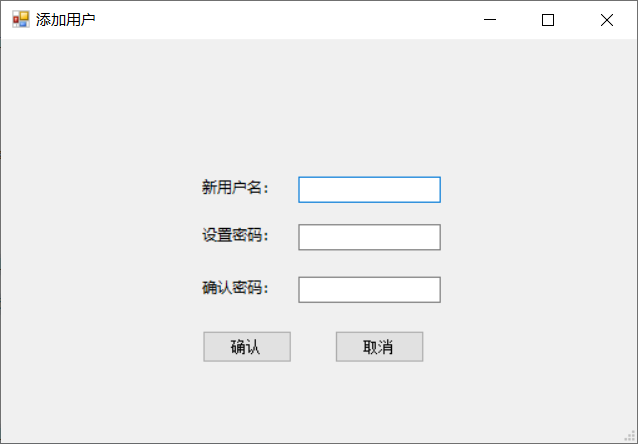
{

MessageBox.Show("登录失败，请重新输入！");

}

dao.DaoClose();

}



用户注册

功能：

输入新用户名可以添加到数据库userdb中，若两次密码输入不同则提示“两次密码不同，请重新输入！”。

单击确认添加，单击取消返回登录界面。

代码：

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

{

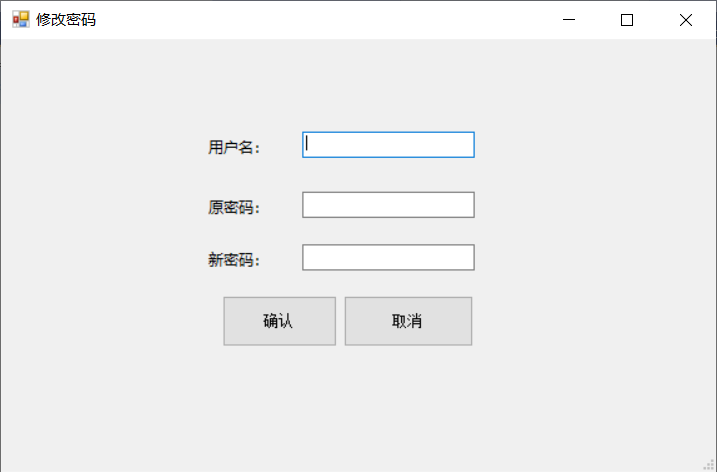
addUser a = new addUser();

this.Hide();

a.ShowDialog();

this.Show();

}



修改密码

功能：

修改已有用户的密码。若原密码和新密码相同，提示“新密码不能与原密码相同！”，单击确认修改，单击取消返回登录界面。

代码：

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

{

changePW c = new changePW();

this.Hide();

c.ShowDialog();

this.Show();

}

2.实现与数据库的连接

通过using System.Data.SqlClient实现连接功能，通过编写特定函数，以下代码来实现对SQL语句的翻译：

public class DATA : DbContext

{//用户存储数据库ID和用户名的Uname

public static string UID = "", Uname = "";//登陆用户的ID

//连接字符串。

public DATA()

: base("name=DATA")

{

}

}

class Dao

{//用于连接数据库

SqlConnection sc;

public SqlConnection Connect()

{

string str = @"Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";//数据库连接字符串

sc = new SqlConnection(str);//创建数据库连接对象

sc.Open();//打开数据库

return sc;//返回数据库连接对象

}

public SqlCommand Command(string sql)

{

SqlCommand cmd = new SqlCommand(sql, Connect());

return cmd;

}

public int Execute(string sql)//更新

{

return Command(sql).ExecuteNonQuery();

}

public SqlDataReader Read(string sql)//读取

{

return Command(sql).ExecuteReader();

}

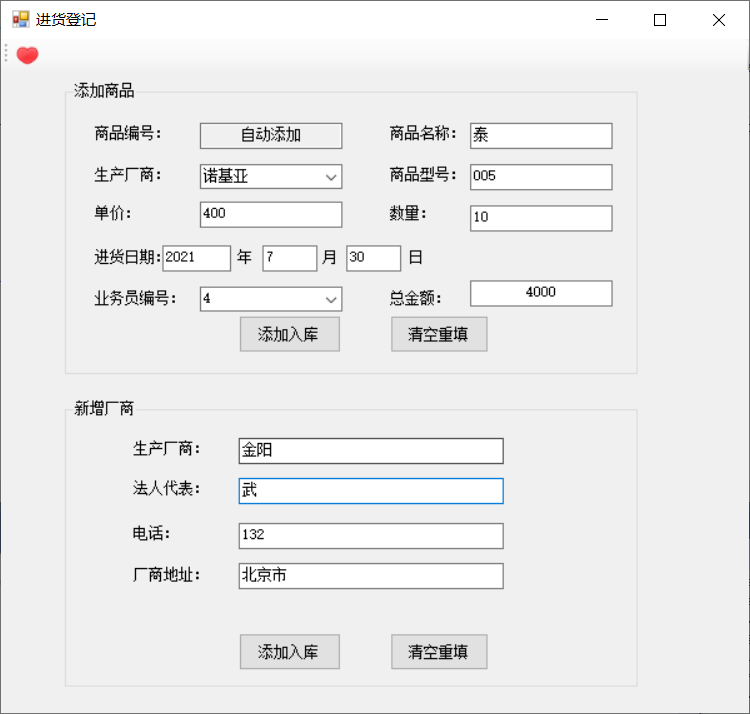
public void DaoClose()

{

sc.Close();//关闭数据库连接

}

}

1. 实现交易管理  
     
   进货登记和新厂商登记

功能：

输入新商品信息和新厂商信息；从员工表中选取全部员工编号作为经手员工下拉栏备选项；鼠标左键单击“添加入库”按钮时，检查信息是否填写完整，不完整则提示“输入不许为空”；鼠标左键单击红心按钮时，返回主界面。单击“清空重填”可以清楚所有信息重新填写。

代码：

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

{

if (textBox2.Text != "" && textBox7.Text != "" && textBox4.Text != "" && textBox5.Text != "" && textBox6.Text != "" && comboBox2.Text != "" && textBox8.Text != "" && textBox9.Text != "" && textBox10.Text != "" && textBox11.Text != "")

{

Dao dao = new Dao();

string sql = $"insert into goods(生产厂商,商品名,型号,单价,数量,总金额,进货年,进货月,进货日,业务员编号) values ('{textBox2.Text}','{textBox7.Text}','{textBox4.Text}',{textBox5.Text},{textBox6.Text},{textBox11.Text},{textBox8.Text},{textBox9.Text},{textBox10.Text},{comboBox2.Text})";

int n = dao.Execute(sql);

if (n > 0)

{

MessageBox.Show("添加商品成功！");

}

else

{

MessageBox.Show("添加商品失败，请重新输入！");

}

dao.DaoClose();

}

else

{

MessageBox.Show("输入不许为空");

}

}

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

{

textBox12.Text = "";

textBox13.Text = "";

textBox14.Text = "";

textBox15.Text = "";

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

Admin goodsIn = new Admin();

this.Hide();

goodsIn.ShowDialog();

}

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

{

if ( textBox12.Text != "" && textBox13.Text != "" && textBox14.Text != "" && textBox15.Text != "" )

{

Dao dao = new Dao();

string sql = $"insert into manufacturer(厂商名称,法人代表,电话,厂商地址) values ('{textBox12.Text}','{textBox13.Text}','{textBox14.Text}','{textBox15.Text}')";

int n = dao.Execute(sql);

if (n > 0)

{

MessageBox.Show("添加厂商成功！");

}

else

{

MessageBox.Show("添加厂商失败，请重新输入！");

}

}

else

{

MessageBox.Show("输入不许为空");

}

}

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

{

textBox12.Text = "";

textBox13.Text = "";

textBox14.Text = "";

textBox15.Text = "";

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

Admin goodsIn = new Admin();

this.Hide();

goodsIn.ShowDialog();

}

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

{

if (textBox5.Text != "" && textBox6.Text != "")

{

int t = Convert.ToInt32(textBox5.Text) \*

Convert.ToInt32(textBox6.Text);

textBox11.Text = Convert.ToString(t);//计算结果填充总金额栏

}

}

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

{

if (textBox5.Text != "" && textBox6.Text != "")

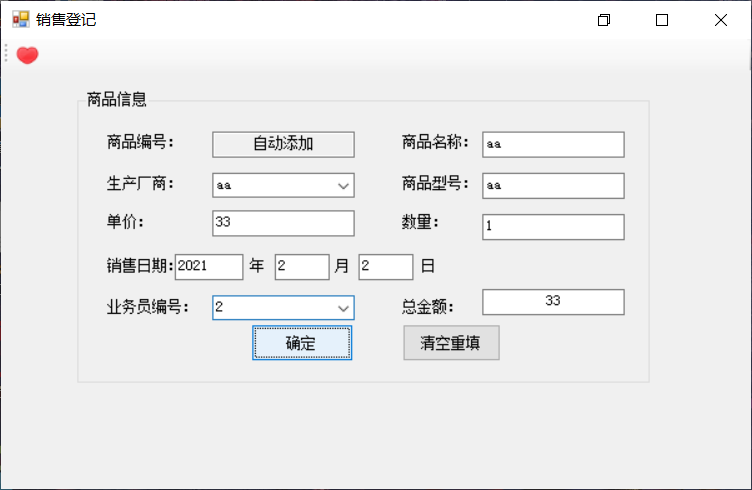
{

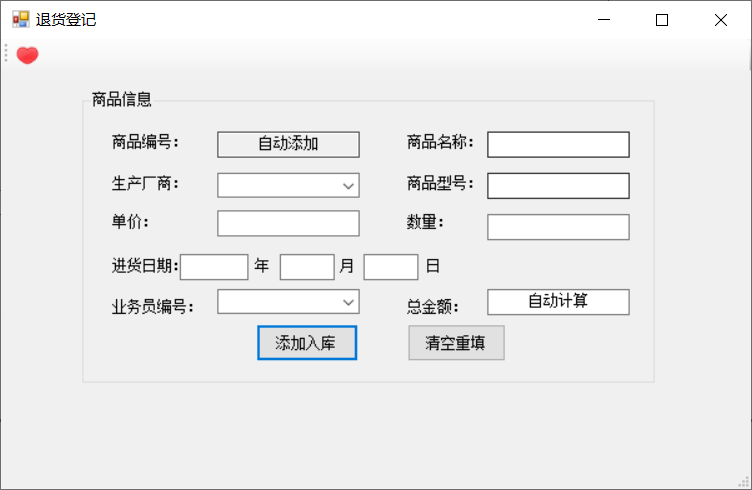
int t = Convert.ToInt32(textBox5.Text) \*

Convert.ToInt32(textBox6.Text);

textBox11.Text = Convert.ToString(t);//计算结果填充总金额栏

}

  
销售登记

  
退货登记

功能：

根据当前是销售登记界面还是退货登记界面，从厂商表中选取全部厂商名作为厂商下拉栏备选项，从员工表中选取全部员工编号作为经手员工下拉栏备选项；鼠标左键单击“添加入库”按钮时，检查信息是否填写完整，不完整则提示“请输入完整信息！”；若的检查均通过，修改数据库中相关表；鼠标左键单击红心按钮时返回主界面。

代码：

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

{

if (comboBox1.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox5.Text != "" && textBox6.Text != "" && comboBox2.Text != "" && textBox8.Text != "" && textBox9.Text != "" && textBox10.Text != "" && textBox11.Text != "")

{

Dao dao = new Dao();

//插入到sell表

string sql1 = $"insert into sell(生产厂商,商品名,型号,单价,数量,总金额,销售年,销售月,销售日,业务员编号) values " +

$"('{comboBox1.Text}','{textBox3.Text}','{textBox4.Text}',{textBox5.Text},{textBox6.Text},{textBox11.Text},{textBox8.Text},{textBox9.Text},{textBox10.Text},{comboBox2.Text})";

int n = dao.Execute(sql1);

//修改goods表

SqlConnection conn;

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

string sql2 = $"select 单价,数量,总金额 from goods where 商品名= '" + textBox3.Text + "' and 型号='" + textBox4.Text + "'and 生产厂商='" + comboBox1.Text + "'";

DataSet ds2;

ds2 = new DataSet();

SqlDataAdapter da2;

da2 = new SqlDataAdapter(sql2, conn);

da2.Fill(ds2, "operator");

int single, count, total, sells;//原单价,原数量,原总价,销售数量

//记录进货表中的原单价,原数量,原总价,销售数量

single = Convert.ToInt32(ds2.Tables["operator"].Rows[0][0]);

count = Convert.ToInt32(ds2.Tables["operator"].Rows[0][1]);

total = Convert.ToInt32(ds2.Tables["operator"].Rows[0][2]);

sells = Convert.ToInt32(textBox4.Text);//记录要销售的数量

if (sells > count)//要出售的数量大于进货数量

{

MessageBox.Show("商品数量不足！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

if (n > 0)

{

MessageBox.Show("添加销售记录成功！");

}

else

{

MessageBox.Show("添加销售记录失败，请重新输入！");

}

dao.DaoClose();

}

else

{

MessageBox.Show("输入不许为空");

}

}

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

{

comboBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

comboBox1.Text = "";

textBox8.Text = "";

textBox9.Text = "";

textBox10.Text = "";

textBox11.Text = "";

}

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

{

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

Admin goodsIn = new Admin();

this.Hide();

goodsIn.ShowDialog();

}

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

{

comboBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

comboBox1.Text = "";

textBox8.Text = "";

textBox9.Text = "";

textBox10.Text = "";

textBox11.Text = "";

}

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

{

if (comboBox1.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox5.Text != "" && textBox6.Text != "" && comboBox2.Text != "" && textBox8.Text != "" && textBox9.Text != "" && textBox10.Text != "" && textBox11.Text != "")

{

Dao dao = new Dao();

string sql = $"insert into retreat(生产厂商,商品名,型号,单价,数量,总金额,退货年,退货月,退货日,业务员编号) values ('{comboBox1.Text}','{textBox3.Text}','{textBox4.Text}',{textBox5.Text},{textBox6.Text},{textBox11.Text},{textBox8.Text},{textBox9.Text},{textBox10.Text},{comboBox2.Text})";

int n = dao.Execute(sql);

if (n > 0)

{

MessageBox.Show("退货成功！");

}

else

{

MessageBox.Show("退货失败，请重新输入！");

}

dao.DaoClose();

}

else

{

MessageBox.Show("输入不许为空");

}

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

Admin goodsIn = new Admin();

this.Hide();

goodsIn.ShowDialog();

}

4.实现进货/销售统计查询



进货查询

功能：

根据选择的条件本日/本月/本季度/本年度，以及输入的年月日查询进货信息。

代码：

int B = 1;

SqlConnection conn;

SqlDataAdapter da;

DataSet ds;

private void toolStripButton1\_Click(object sender, EventArgs e)

{

todayIn main = new todayIn();//主界面

this.Close();//关闭进货登记界面

main.Show();//显示主界面

}

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

{

if (B == 1)//日

{

if (textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from goods where 进货日=" + textBox4.Text +

" and 进货月=" + textBox3.Text +

" and 进货年=" + textBox2.Text +

" order by 生产厂商,总金额", conn);

ds = new DataSet();

da.Fill(ds, "goods");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 进货" +

"总金额 from goods where 进货日=" + textBox4.Text +

" and 进货月=" + textBox3.Text +

" and 进货年=" + textBox2.Text +

"group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from goods" +

" where 进货日=" + textBox4.Text +

" and 进货月=" + textBox3.Text +

" and 进货年=" + textBox4.Text, conn);

conn.Close();

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

else if (B == 2)//月

{

if (textBox2.Text == "" || textBox3.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from goods where " +

"进货月=" + textBox3.Text +

" and 进货年=" + textBox2.Text +

" order by 生产厂商,总金额", conn);

ds = new DataSet();

da.Fill(ds, "goods");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 进货" +

"总金额 from goods where " +

"进货月=" + textBox3.Text +

" and 进货年=" + textBox2.Text +

"group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from goods" +

" where" +

" 进货月=" + textBox3.Text +

" and 进货年=" + textBox2.Text, conn);

conn.Close();

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

else if (B == 3)//季度

{

if (textBox2.Text == "" || textBox3.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

string jd;

if (Convert.ToInt32(textBox3.Text) >= 1 && Convert.ToInt32(textBox3.Text) <= 3)

{ jd = "进货月=1 or 进货月=2 or 进货月=3 "; }

else if (Convert.ToInt32(textBox3.Text) >= 4 && Convert.ToInt32(textBox3.Text) <= 6)

{ jd = "进货月=4 or 进货月=5 or 进货月=6 "; }

else if (Convert.ToInt32(textBox3.Text) >= 7 && Convert.ToInt32(textBox3.Text) <= 9)

{ jd = "进货月=7 or 进货月=8 or 进货月=9"; }

else

{ jd = "进货月=10 or 进货月=11 or 进货月=12 "; }

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from goods where (" + jd +

") and 进货年=" + textBox2.Text +

" order by 生产厂商,进货月,总金额", conn);

ds = new DataSet();

da.Fill(ds, "goods");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 进货" +

"总金额 from goods where (" + jd +

") and 进货年=" + textBox2.Text +

" group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from goods" +

" where (" + jd +

") and 进货年=" + textBox2.Text, conn);

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

else//年

{

if (textBox2.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from goods where " +

"进货年=" + textBox2.Text +

" order by 生产厂商,总金额", conn);

ds = new DataSet();

da.Fill(ds, "goods");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 进货" +

"总金额 from goods where " +

"进货年=" + textBox2.Text +

"group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from goods" +

" where" +

" 进货年=" + textBox2.Text, conn);

conn.Close();

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

}

private void 本日ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 1;

}

private void 本月ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 2;

}

private void 本年度ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 4;

}

private void toolStripButton1\_Click\_1(object sender, EventArgs e)

{

Admin a = new Admin();

this.Close();//关闭进货登记界面

a.Show();//显示主界面

}

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

{

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

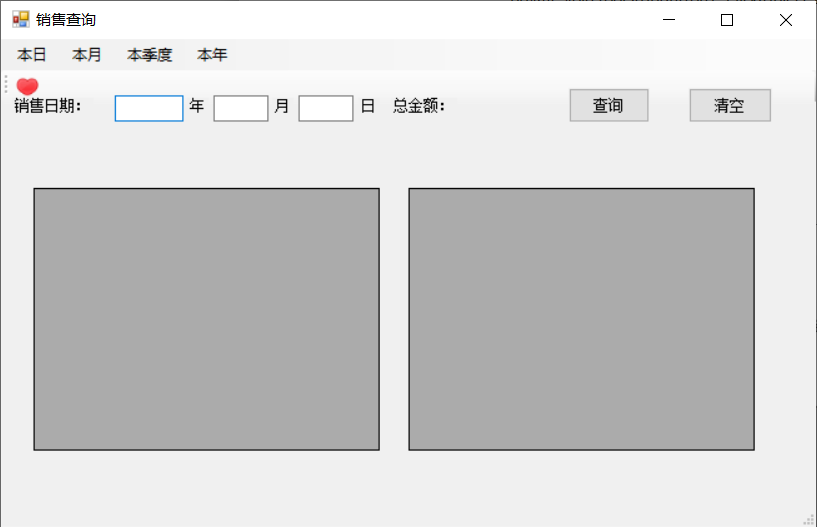
}

private void 本季度ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 3;

}





销售查询

功能：

根据选择的条件本日/本月/本季度/本年度，以及输入的年月日查询进货信息。

代码：

SqlConnection conn;

SqlDataAdapter da;

DataSet ds;

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

{

}

private void 本日ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 1;

}

private void 本月ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 2;

}

private void 本季度ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 3;

}

private void 本年ToolStripMenuItem\_Click(object sender, EventArgs e)

{

B = 4;

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

Admin main = new Admin();//主界面

this.Close();//关闭销售登记界面

main.Show();//显示主界面

}

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

{

if (B == 1)//日

{

if (textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from sell where 销售日=" + textBox4.Text +

" and 销售月=" + textBox3.Text +

" and 销售年=" + textBox2.Text +

" order by 生产厂商,总金额", conn);

ds = new DataSet();

da.Fill(ds, "sell");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 销售" +

"总金额 from sell where 销售日=" + textBox4.Text +

" and 销售月=" + textBox3.Text +

" and 销售年=" + textBox2.Text +

"group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from sell" +

" where 销售日=" + textBox4.Text +

" and 销售月=" + textBox3.Text +

" and 销售年=" + textBox4.Text, conn);

conn.Close();

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

else if (B == 2)//月

{

if (textBox2.Text == "" || textBox3.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from sell where " +

"销售月=" + textBox3.Text +

" and 销售年=" + textBox2.Text +

" order by 生产厂商,总金额", conn);

ds = new DataSet();

da.Fill(ds, "goods");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 销售" +

"总金额 from sell where " +

"销售月=" + textBox3.Text +

" and 销售年=" + textBox2.Text +

"group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from sell" +

" where" +

" 销售月=" + textBox3.Text +

" and 销售年=" + textBox2.Text, conn);

conn.Close();

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

else if (B == 3)//季度

{

if (textBox2.Text == "" || textBox3.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

string jd;

if (Convert.ToInt32(textBox3.Text) >= 1 && Convert.ToInt32(textBox3.Text) <= 3)

{ jd = "销售月=1 or 销售月=2 or 销售月=3 "; }

else if (Convert.ToInt32(textBox3.Text) >= 4 && Convert.ToInt32(textBox3.Text) <= 6)

{ jd = "销售月=4 or 销售月=5 or 销售月=6 "; }

else if (Convert.ToInt32(textBox3.Text) >= 7 && Convert.ToInt32(textBox3.Text) <= 9)

{ jd = "销售月=7 or 销售月=8 or 销售月=9"; }

else

{ jd = "销售月=10 or 销售月=11 or 销售月=12 "; }

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from sell where (" + jd +

") and 销售年=" + textBox2.Text +

" order by 生产厂商,销售月,总金额", conn);

ds = new DataSet();

da.Fill(ds, "goods");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 销售" +

"总金额 from sell where (" + jd +

") and 销售年=" + textBox2.Text +

" group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from sell" +

" where (" + jd +

") and 进货年=" + textBox2.Text, conn);

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

else//年

{

if (textBox2.Text == "")

MessageBox.Show("请输入完整日期！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da = new SqlDataAdapter("select \* from sell where " +

"销售年=" + textBox2.Text +

" order by 生产厂商,总金额", conn);

ds = new DataSet();

da.Fill(ds, "goods");

dataGridView2.DataSource = ds.Tables["goods"];

da = new SqlDataAdapter("select 生产厂商,sum(总金额) as 销售" +

"总金额 from sell where " +

"销售年=" + textBox2.Text +

"group by 生产厂商 order by 生产厂商", conn);

ds = new DataSet();

da.Fill(ds, "goods1");

dataGridView1.DataSource = ds.Tables["goods1"];

da = new SqlDataAdapter("select sum(总金额) from sell" +

" where" +

" 销售年=" + textBox2.Text, conn);

conn.Close();

ds = new DataSet();

da.Fill(ds, "goods2");

label5.Text = ds.Tables["goods2"].Rows[0][0].ToString();

label5.Show();

}

}

}

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

{

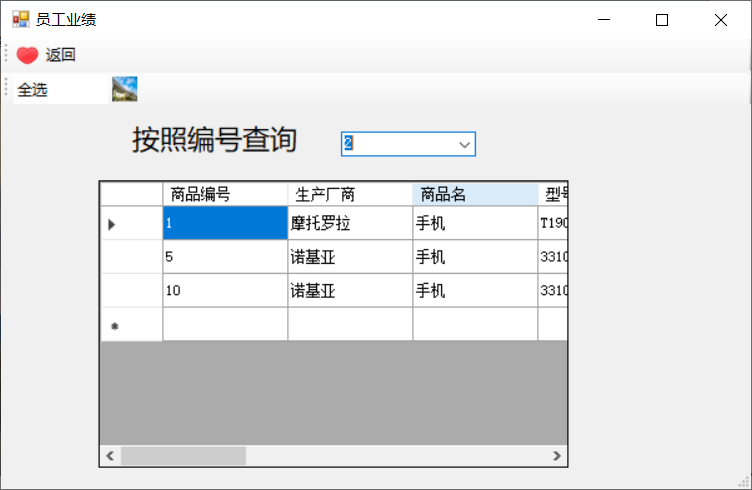
textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

}

5.实现业绩查看



功能：

可以下拉选择员工编号查询，也可以全选查看所有人的信息。

代码：

SqlConnection conn;

SqlDataAdapter da, da1;

DataSet ds, ds1;

private void toolStripButton1\_Click(object sender, EventArgs e)

{

Admin a = new Admin();

this.Close();//关闭进货登记界面

a.Show();//显示主界面

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da1 = new SqlDataAdapter("select \* from sell", conn);

ds1 = new DataSet();

da1.Fill(ds1, "sell");

conn.Close();

dataGridView1.DataSource = ds1.Tables["sell"];

}

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

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

ds = new DataSet();

if (conn.State == ConnectionState.Closed)

{

conn.Open(); //若未连接，连接

}

da = new SqlDataAdapter("select 员工编号 from employee", conn);

da.Fill(ds, "员工编号");//引入用户名

conn.Close();

comboBox1.DisplayMember = "员工编号";//逐条显示用户名

comboBox1.DataSource = ds.Tables["员工编号"];//用户名作为登录界面的数据源

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=SHAW-FENE;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

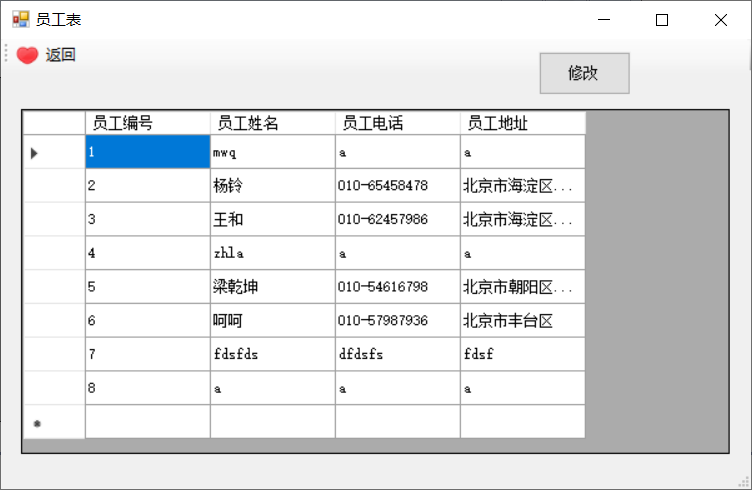
da1 = new SqlDataAdapter("select \* from sell where 业务员编号=" + comboBox1.Text, conn);

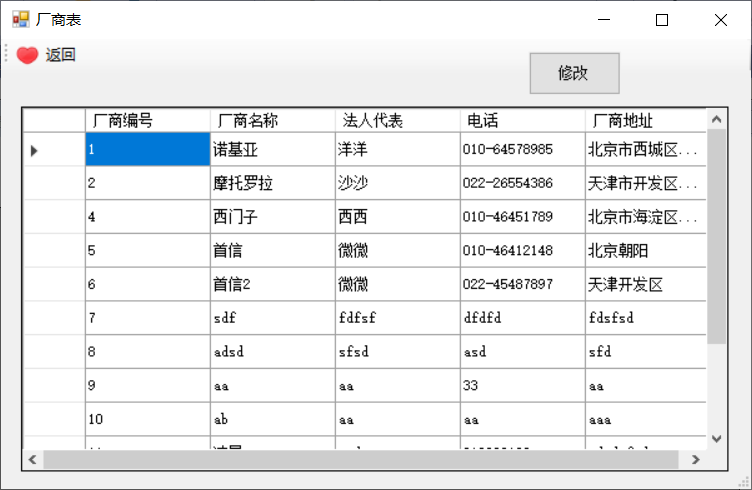
ds1 = new DataSet();

da1.Fill(ds1, "sell1");

dataGridView1.DataSource = ds1.Tables["sell1"];

}

6.实现报表查看  




功能：

显示对应表格；若是员工表/进货商表，显示“修改”按钮，鼠标左键单击“修改”按钮时，显示修改员工信息窗体/修改进货商信息窗体；

鼠标左键单击红心按钮时，返回主窗体；

代码：

public partial class Formseetable : Form

{

public Formseetable()

{

InitializeComponent();

}

//初始化一些标签

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

{

label1.Text = "表单详情:";

button1.Hide();

}

//返回按钮

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

{

mainform frmmain = new mainform();//主界面

this.Hide();

frmmain.Show();

}

SqlConnection conn;

SqlDataAdapter da, da1;

DataSet ds, ds1;

//查看退货表单

private void toolStripButton3\_Click(object sender, EventArgs e)

{

button1.Hide();

label1.Text = "退货表详情: ";

dataGridView1.DataSource = null;

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da1 = new SqlDataAdapter("select \* from retreat order by 退货年 desc,退货月 desc,退货日 desc", conn);

ds1 = new DataSet();

da1.Fill(ds1, "retreat");

conn.Close();

dataGridView1.DataSource = ds1.Tables["retreat"];

}

internal void toolStripButton5\_Click()

{

throw new NotImplementedException();

}

//查看员工表详情

private void toolStripButton4\_Click(object sender, EventArgs e)

{

button1.Show();

label1.Text = "员工表详情: ";

dataGridView1.DataSource = null;

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da1 = new SqlDataAdapter("select \* from employee order by 员工编号", conn);

ds1 = new DataSet();

da1.Fill(ds1, "employee");

conn.Close();

dataGridView1.DataSource = ds1.Tables["employee"];

}

//查看进货表详情

public void toolStripButton5\_Click(object sender, EventArgs e)

{

button1.Show();//显示修改按钮

label1.Text = "进货商表详情: ";

dataGridView1.DataSource = null;

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da1 = new SqlDataAdapter("select \* from manufacturer order by 厂商编号", conn);

ds1 = new DataSet();

da1.Fill(ds1, "manufacturer");

conn.Close();

dataGridView1.DataSource = ds1.Tables["manufacturer"];

}

//修改按钮

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

{

if(label1.Text== "进货商表详情: ")//弹出修改进货商的窗口

{

Formchangemanu frmchangemanu = new Formchangemanu();

this.Close();

frmchangemanu.Show();

}

else if(label1.Text == "员工表详情: ")//弹出修改员工的窗口

{

Formchangeemployee frmchangemployee = new Formchangeemployee();

this.Close();

frmchangemployee.Show();

}

else

{

MessageBox.Show("此项不可修改!", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

//显示销售表详情

private void toolStripButton2\_Click(object sender, EventArgs e)

{

button1.Hide();

label1.Text = "销售表详情: ";

dataGridView1.DataSource = null;

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da1 = new SqlDataAdapter("select \* from sell order by 销售年 desc,销售月 desc,销售日 desc", conn);

ds1 = new DataSet();

da1.Fill(ds1, "sell");

conn.Close();

dataGridView1.DataSource = ds1.Tables["sell"];

}

//显示进货表详情

private void toolStripButton1\_Click(object sender, EventArgs e)

{

button1.Hide();

label1.Text = "进货表详情: ";

dataGridView1.DataSource = null;

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";

if (conn.State == ConnectionState.Closed)

{ conn.Open(); }

da1 = new SqlDataAdapter("select \* from goods order by 进货年 desc,进货月 desc,进货日 desc", conn);

ds1 = new DataSet();

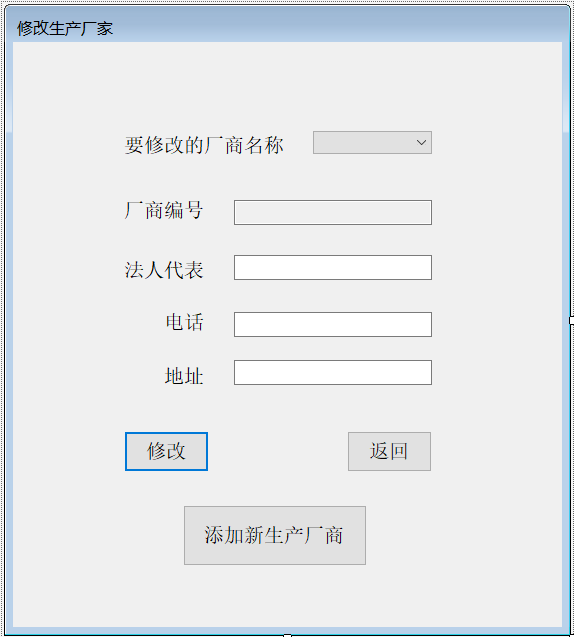
da1.Fill(ds1, "goods");

dataGridView1.DataSource = ds1.Tables["goods"];

}

}

}



功能：

选取厂商表中全部厂商名作为要修改的厂商名称下拉栏备选项；选取的厂商变化时，自动显示厂商编号和其他原信息；鼠标左键单击“修改”时，修改厂商表中相应数据；鼠标左键单击“返回”按钮时，返回查看厂商表界面；鼠标左键单击“添加新生产厂商”按钮时，显示添加厂商界面。

代码：

private void button1\_Click(object sender, EventArgs e)//添加新厂商

{

Formaddmnft frmaddmnft = new Formaddmnft();//添加厂商界面

frmaddmnft.Show();

frmaddmnft.frm = 1;//控制添加厂商窗口关闭后返回哪个界面

}

SqlConnection conn;

SqlCommand com;

SqlDataAdapter da,da1;

DataSet ds,ds1;

//返回按钮

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

{

Formseetable frmseetable = new Formseetable();//返回查看数据表

this.Close();

frmseetable.Show();

}

//修改按钮

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

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

com = new SqlCommand("update manufacturer set 法人代表='"+ textBox2.Text +

"',电话='" +textBox3.Text + "',厂商地址='" + textBox4.Text +

"'where 厂商名称='" +comboBox1.Text+"'", conn);

com.ExecuteNonQuery();//执行修改厂商信息界面

conn.Close();

MessageBox.Show("修改厂商信息成功！", "提示",

MessageBoxButtons.OK, MessageBoxIcon.Information);//修改厂商信息成功

//刷新界面

Formchangemanu formchangemanu = new Formchangemanu();

this.Close();

formchangemanu.Show();

}

//下拉栏所选厂商变化时

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

ds1 = new DataSet();

if (conn.State == ConnectionState.Closed)

{

conn.Open(); //若未连接，连接

}

da1 = new SqlDataAdapter("select \* from manufacturer where 厂商名称='"

+ comboBox1.Text+"'", conn);

ds1.Clear();

da1.Fill(ds1, "厂商表");//引入厂商信息

conn.Close();

//显示原来厂商信息

textBox1.Text = ds1.Tables["厂商表"].Rows[0][0].ToString();

textBox2.Text = ds1.Tables["厂商表"].Rows[0][2].ToString();

textBox3.Text = ds1.Tables["厂商表"].Rows[0][3].ToString();

textBox4.Text = ds1.Tables["厂商表"].Rows[0][4].ToString();

}

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

{

try

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

ds = new DataSet();

if (conn.State == ConnectionState.Closed)

{

conn.Open(); //若未连接，连接

}

da = new SqlDataAdapter("select 厂商名称 from manufacturer group by 厂商名称", conn);

da.Fill(ds, "厂商名称");//引入厂商名称

conn.Close();

comboBox1.DisplayMember = "厂商名称";//逐条显示厂商名称

comboBox1.DataSource = ds.Tables["厂商名称"];//用厂商名称作为修改厂商界面的数据源

}

catch (Exception ex)

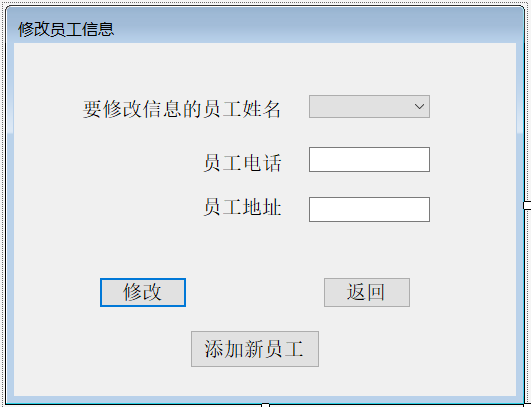
{

MessageBox.Show(ex.Message.ToString());

}

}

}



修改员工信息界面

功能：

从员工表中选取所有员工姓名作为要修改的员工姓名下拉栏备选项；所选的员工姓名变化时，自动显示原信息；鼠标左键单击“修改”按钮时，修改员工表中相应信息；鼠标左键单击“添加新员工”按钮时，显示添加新员工界面；鼠标左键单击“返回”按钮时，返回查看员工表界面。

代码：

SqlConnection conn;

SqlCommand com;

SqlDataAdapter da, da1;

DataSet ds, ds1;

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

ds1 = new DataSet();

if (conn.State == ConnectionState.Closed)

{

conn.Open(); //若未连接，连接

}

da1 = new SqlDataAdapter("select \* from employee where 员工姓名='"

+ comboBox1.Text + "'", conn);

ds1.Clear();

da1.Fill(ds1, "员工表");//引入员工姓名信息

conn.Close();

//显示员工的原电话号码

textBox1.Text = ds1.Tables["员工表"].Rows[0][2].ToString();

//显示员工的原地址号码

textBox2.Text = ds1.Tables["员工表"].Rows[0][3].ToString();

}

//修改按钮

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

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

com = new SqlCommand("update employee set 员工电话='" + textBox1.Text +

"',员工地址='" + textBox2.Text +

"'where 员工姓名='" + comboBox1.Text + "'", conn);

com.ExecuteNonQuery();//执行修改命令

conn.Close();

MessageBox.Show("修改员工信息成功！", "提示",

MessageBoxButtons.OK, MessageBoxIcon.Information);//修改员工信息成功

//返回查看员工表界面

Formchangeemployee formchangeemployee = new Formchangeemployee();

this.Close();

formchangeemployee.Show();

}

//添加新员工按钮

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

{

//打开添加员工界面

Formaddemployee frmaddemployee = new Formaddemployee();

frmaddemployee.Show();

}

//返回按钮

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

{

Formseetable frmseetable = new Formseetable();//返回查看数据表

this.Close();

frmseetable.Show();

}

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

{

try

{

conn = new SqlConnection();

conn.ConnectionString = "Data Source=LAPTOP-M1M538S7\\SQLEXPRESS;Initial Catalog=sellsystem;Integrated Security=True";//连接数据库

ds = new DataSet();

if (conn.State == ConnectionState.Closed)

{

conn.Open(); //若未连接，连接

}

da = new SqlDataAdapter("select 员工姓名 from employee group by 员工姓名", conn);

da.Fill(ds, "员工姓名");//引入员工姓名

conn.Close();

comboBox1.DisplayMember = "员工姓名";//逐条显示员工姓名

comboBox1.DataSource = ds.Tables["员工姓名"];//用员工姓名作为修改界面的数据源

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

7.实现切换用户  


# 用户使用说明书

## 一、软件用途

本软件的开发是为实现销售单位的高效管理而设计的。

## 二、具体操作

1. 系统登陆

在相应的框内输入用户名和密码，点击登录即可实现登陆。

1. 用户注册

若第一次进入系统，则可通过点击做下角的添加用户按钮添加用户，在相应的框内输入您的用户名和密码，并二次确认密码无误后点击确定即可添加新用户至数据库里。

1. 修改密码

若您想修改密码，则可点击登陆界面右下角的修改密码按钮进行密码的修改，输入新的密码并确认无误后，点击确定即可修改密码。

1. 交易管理

①进货登记

在菜单上选择交易管理-进货登记即可弹出进货登记界面，通过填写各项数据，点击添加入库便可实现入库操作，点击清空重填即可清空所有信息。新的厂商也可以在这个界面上添加自己的信息。

②销售登记

在菜单上选择交易管理-销售登记即可弹出销售登记界面，通过填写销售信息点击销售即可登记销售。点击清空重填即可清空所有信息。

③退货登记

在菜单上选择交易管理-退货登记即可弹出退货登记界面，通过填写退货信息即可登记退货。按钮设置见销售登记。

1. 进货统计

①今日进货量：在菜单上选择进货统计-今日进货量即可。

②本月进货量：在菜单上选择进货统计-本月进货量即可。

③本季度进货量：在菜单上选择进货统计-本季度进货量即可。

④今年进货量：在菜单上选择进货统计-今年进货量即可。

1. 销售统计

①今日销售量：在菜单上选择销售统计-今日销售量即可。

②本月销售量：在菜单上选择销售统计-本月销售量即可。

③本季度销售量：在菜单上选择销售统计-本季度销售量即可。

④今年销售量：在菜单上选择销售统计-今年销售量即可。

1. 业绩查看

在菜单上选择业绩查看-业绩查看即可弹出业绩查看界面。

可以查询特定员工的销售情况，也可点击显示全部员工销售额来显示全部员工的销售额。

1. 查看数据表

①现存货表：在菜单上选择查看数据表-现存货表即可弹出现存货表界面。

通过输入商品名和型号查询商品的销售情况，也可点击底部的显示全部现存商品来显示全部现存商品。

②已售商品表：在菜单上选择查看数据表-已售商品表即可弹出已售商品表界面。

通过输入商品名和型号查询商品的现存情况，也可点击底部的显示全部售出商品来显示全部已售商品。

③退货表：在菜单上选择查看数据表-退货表即可弹出退货表界面。

④员工表：在菜单上选择查看数据表-员工货表即可弹出员工表界面。

若要修改记录，点击左下角的修改记录按钮，在弹出的文本框中输入修改后的数据，单击回车键即可修改数据，再次点击修改完成即可。

若想删除记录，点击想要删除的任意一行，待蓝色高亮行后，点击删除记录即可删除记录。

⑤进货商表：在菜单上选择查看数据表-进货商表即可弹出厂商界面。

具体操作见查询员工货表。

# 结束语

我花了大约三天写完，其难度和时长远远超过我的想象。体验了一把程序员的生活，不停地搜索，不停bug又不停Debug。实践是检验真理的唯一标准，看来我对数据库理论的把握还远远不够。我最满意的是登记信息部分，我把返回和清空重填功能加了进去，更加人性化，更方便操作。不过有些文字还需要换样式优化，包括总金额可以采取自动计算的方式显示。