**《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 地信17-1班**

**学号 07172336 姓名  张清昱 指导教师 张华**

**环境与测绘学院**

**2019年 7 月 17 日**

**《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#或ASP或python或C++或Java或JavaScript，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#或ASP或python或C++或Java或JavaScript。

**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。数据库系统一般提供高效的数据控制和数据检索功能，采用SQL语言来进行数据操作。本次课程设计实验尝试使用Python制作GUI界面让数据库的管理和维护可视化，内部连接Mysql数据库，使用SQL语句来实现对数据库的数据操纵，最终目的能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，让程序在能力范围内没有使用上的障碍或Bug。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、内容讨喜的特点。

关键词：Python Mysql 销售管理

Summary

The use of the database can make data management more fluent and the data trend clearer. Both large companies and individual households need to use a database to manage related data. The database is based on mathematical logic and setting operations. It is efficient, reliable, complete, and self-synchronizing. It is the best choice for business systems to control data. The database system generally provides efficient data control and data retrieval functions, and uses the SQL language for data operations. This course design experiment attempts to use Python to create a GUI interface to visualize the management and maintenance of the database, internally connect to the Mysql database, and use SQL statements to implement data manipulation of the database. The ultimate goal is to enable managers to implement data in the management of the database. Typing the library, one-click outbound operation, it is more convenient to manage the data file, so that the program has no obstacles or bugs in the scope of its capabilities. The program design is as concise as possible, whether it is code or interface, try to achieve the characteristics of high readability, easy to get started, and content is pleasing.

Keywords: Python Mysql Sales Management

数据库课程设计报告...................................................................................................................- 1 -

[摘要 - 6 -](#_Toc1841050072)

[第一章 概述 - 8 -](#_Toc1584174128)

[一、项目背景 - 8 -](#_Toc712313790)

[二、编写目的 - 8 -](#_Toc1784020152)

[三、软件定义 - 9 -](#_Toc860015250)

[四、开发环境 - 9 -](#_Toc1711362440)

[第二章 需求分析 - 9 -](#_Toc1620044809)

[一、性能需求 - 9 -](#_Toc147944550)

[二、主要功能 - 10 -](#_Toc1865472271)

[三、数据流图 - 11 -](#_Toc1878696144)

[四、数据字典 - 11 -](#_Toc794030367)

[第三章 软件概要设计 - 16 -](#_Toc804995711)

[一、系统目标 - 16 -](#_Toc415938677)

[二、总体设计 - 17 -](#_Toc622073354)

[三、运行设计 - 18 -](#_Toc1236467082)

[四、软件功能图 - 18 -](#_Toc102995155)

[第四章 数据库设计 - 20 -](#_Toc167750603)

[一、数据库概念设计 - 20 -](#_Toc1885839757)

[二、数据库逻辑设计 - 22 -](#_Toc597649826)

[第五章 详细设计 - 24 -](#_Toc919608563)

[一、建立数据库 - 24 -](#_Toc421310882)

[二、应用程序设计 - 27 -](#_Toc718409615)

[第六章 用户使用说明书 - 90 -](#_Toc1157335871)

[一、软件用途  - 90 -](#_Toc1584593018)

[二、具体操作 - 90 -](#_Toc1310147079)

[第七章 结束语 - 95 -](#_Toc1492124062)

# 概述

## 项目背景

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。

## 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Pycharm，基于Python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，销售更加轻松。

## 开发环境

1. macOS Mojave 10.14.3系统
2. Mysql-8.0.16 for macos10.14-x86\_64
3. PyCharm CE 2018.3.6
4. Navicat for MySql 12.1.19 TNT Edition

# 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，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

用户登陆，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；

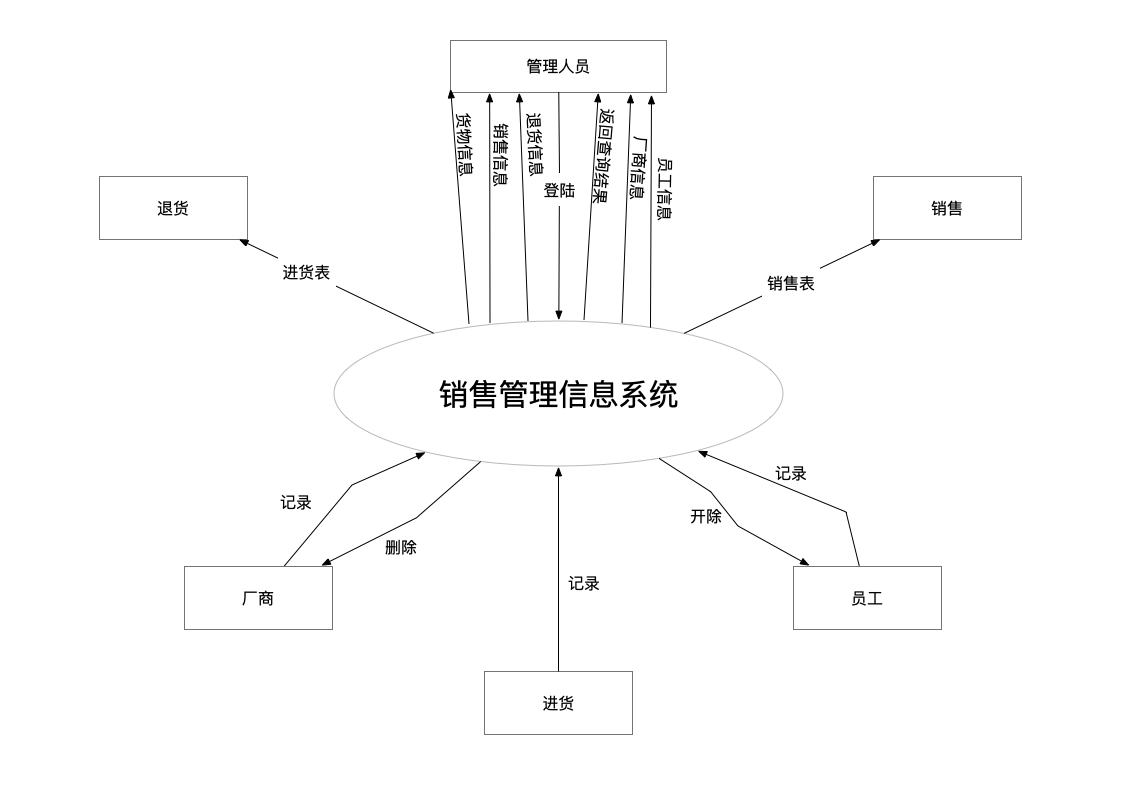
进货统计管理，其中包括有关进货的各种统计数据表，如今日进货统计、本月进货统计、本季度进货统计、本年度销售统计等；

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。

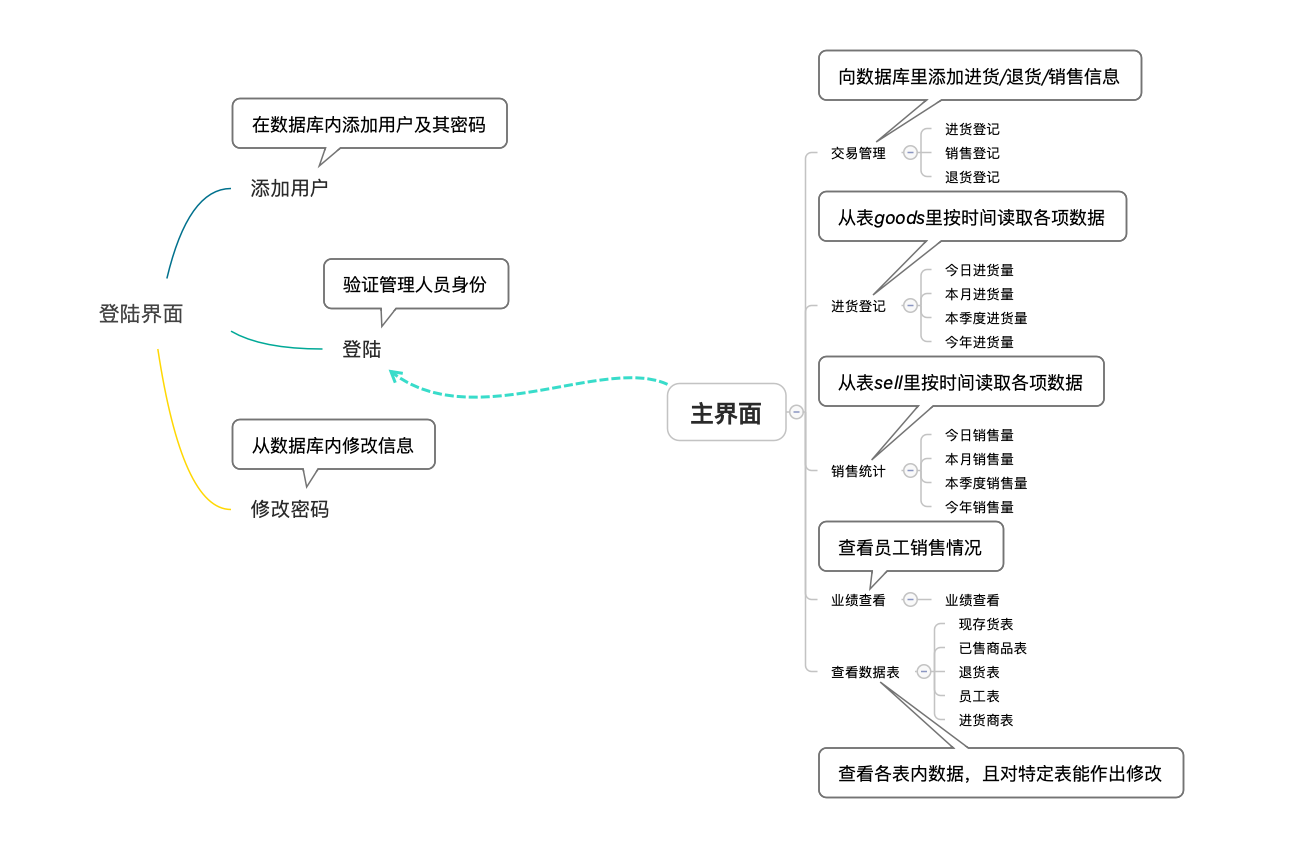
## 二、总体设计

|  |  |  |  |
| --- | --- | --- | --- |
| 销售管理信息系统 | | | |
| 登录界面 | | 登陆  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 |

## 运行设计

目标程序预计占用内存在50M以内，使用python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,采取循环反复的方法来获得设计。

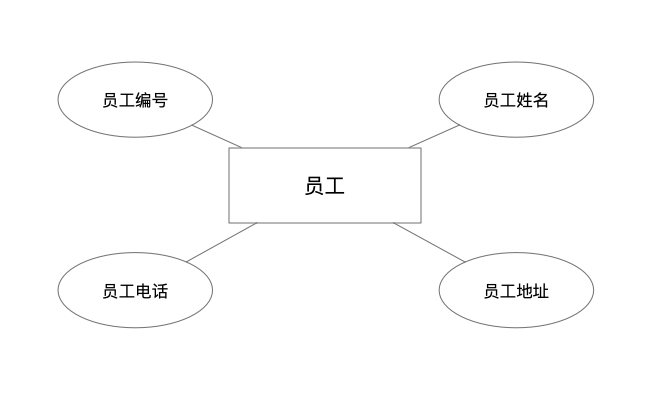
## 软件功能图



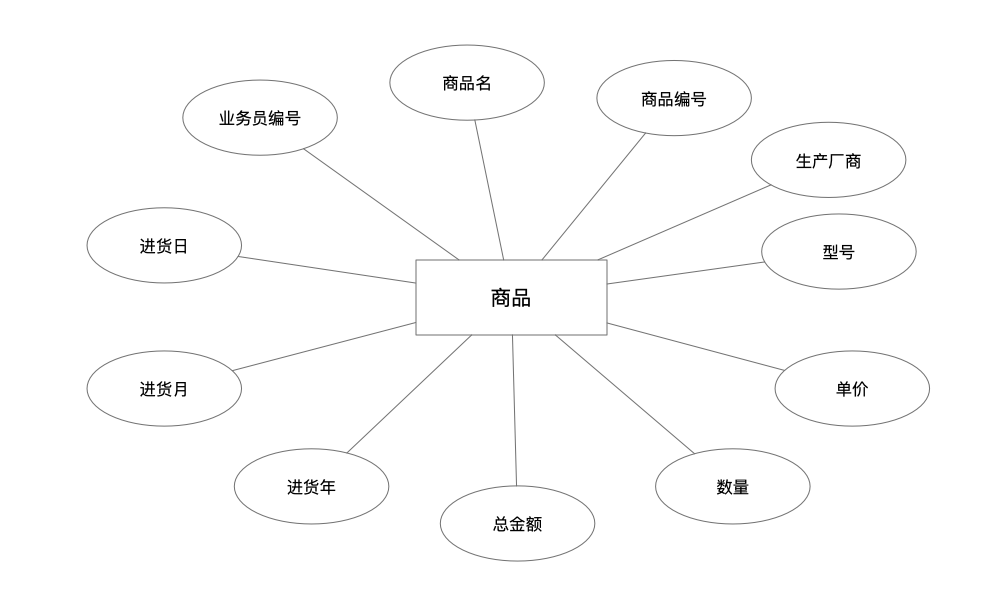
# 数据库设计

## 数据库概念设计

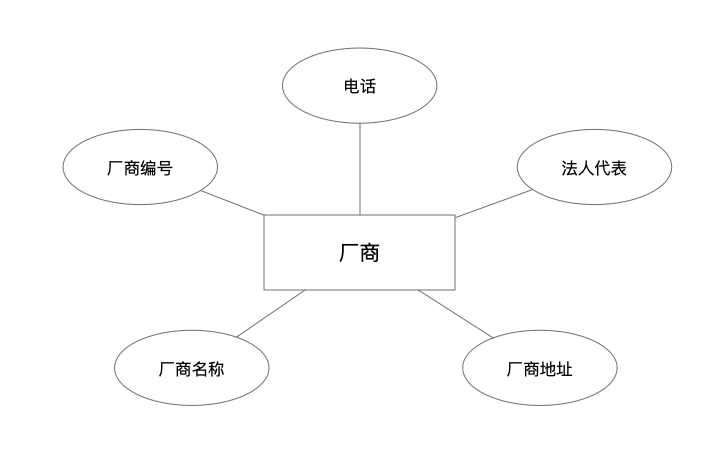
### 员工E-R图



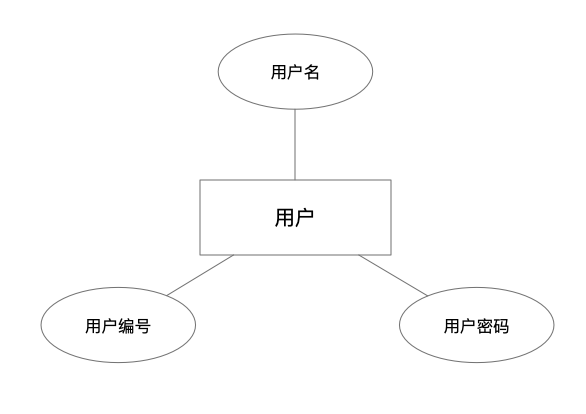
### 2.商品E-R图



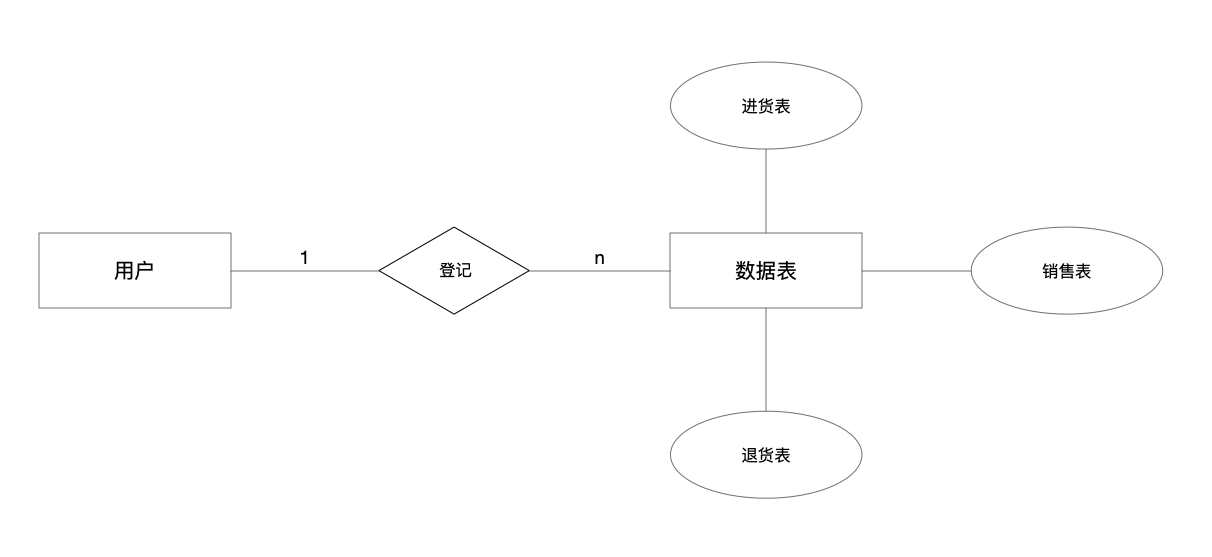
### 3.厂商E-R图



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、3NF 优化：

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）

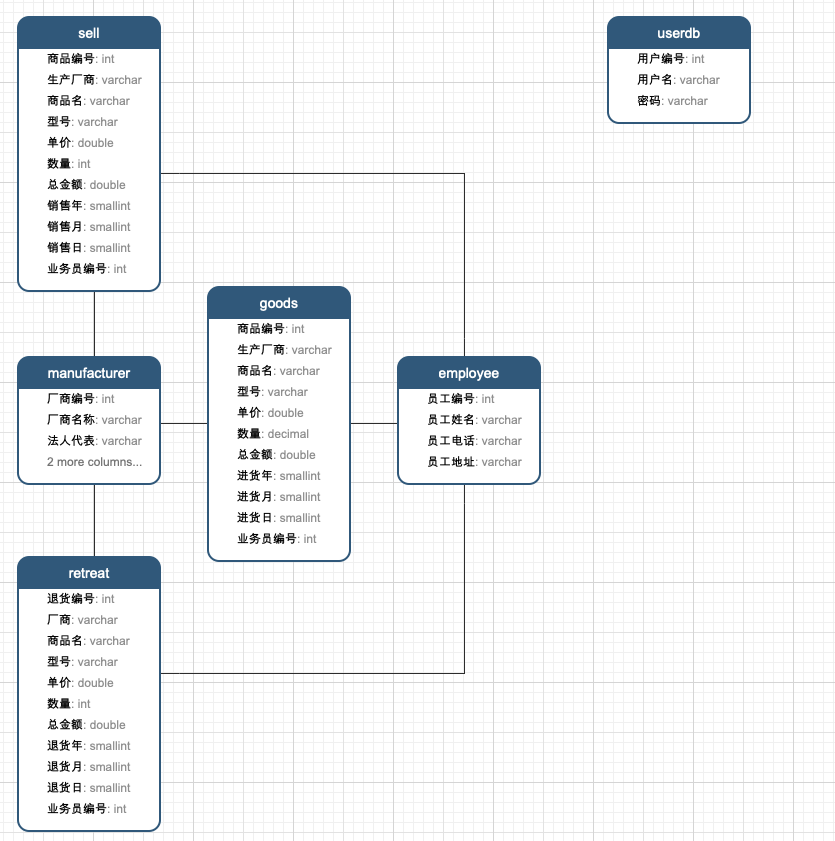
商品信息（商品编号、业务员编号、进货年月日、销售年月日、退货年月日）

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

3、表单关系图



# 详细设计

## 建立数据库

在MySql中根据《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,

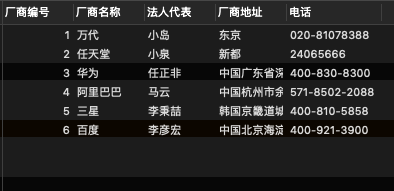
`厂商地址` varchar(100) DEFAULT NULL,

`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` (

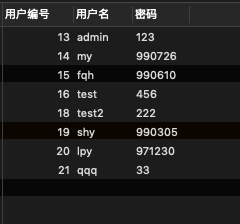
`用户编号` int(255) NOT NULL AUTO\_INCREMENT,

`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



主界面



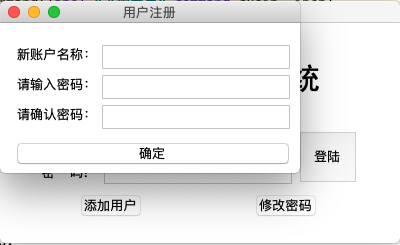
菜单

代码如下：

*# -\*-coding:gbk-\*-***from** tkinter **import** \*  
**import** tkinter.messagebox **as** message  
**import** excuate  
**import** jygl  
**import** jhtj  
**import** xstj  
**import** yjck  
**import** cksjb  
  
  
top2 = Tk()  
top2.geometry(**"600x400+500+300"**)  
top2.title(**'销售管理信息系统 v1.0'**)  
mainmenu = Menu(top2)  
menu\_jygl = Menu(mainmenu)  
mainmenu.add\_cascade(label=**"交易管理"**, menu=menu\_jygl)  
menu\_jygl.add\_command(label=**"进货登记"**, command=jygl.\_jhdj)  
menu\_jygl.add\_command(label=**"销售登记"**, command=jygl.\_xsdj)  
menu\_jygl.add\_command(label=**"退货登记"**, command=jygl.\_thdj)  
  
menu\_jhtj = Menu(mainmenu)  
mainmenu.add\_cascade(label=**"进货统计"**, menu=menu\_jhtj)  
menu\_jhtj.add\_command(label=**"今日进货量"**, command=jhtj.\_jrjh)  
menu\_jhtj.add\_command(label=**"本月进货量"**, command=jhtj.\_byjh)  
menu\_jhtj.add\_command(label=**"本季度进货量"**, command=jhtj.\_jdjh)  
menu\_jhtj.add\_command(label=**"今年进货量"**, command=jhtj.\_jnjh)  
  
menu\_xstj = Menu(mainmenu)  
mainmenu.add\_cascade(label=**"销售统计"**, menu=menu\_xstj)  
menu\_xstj.add\_command(label=**"今日销售量"**, command=xstj.\_jrxs)  
menu\_xstj.add\_command(label=**"本月销售量"**, command=xstj.\_byxs)  
menu\_xstj.add\_command(label=**"本季度销售量"**, command=xstj.\_jdxs)  
menu\_xstj.add\_command(label=**"今年销售量"**,command=xstj.\_jnxs)  
  
menu\_yjck = Menu(mainmenu)  
mainmenu.add\_cascade(label=**"业绩查看"**, menu=menu\_yjck)  
menu\_yjck.add\_command(label=**"业绩查看"**, command=yjck.\_yjck)  
  
menu\_cksjb = Menu(mainmenu)  
mainmenu.add\_cascade(label=**"查看数据表"**, menu=menu\_cksjb)  
menu\_cksjb.add\_command(label=**"现存货表"**, command=cksjb.\_xchb)  
menu\_cksjb.add\_command(label=**"已售商品表"**, command=cksjb.\_scys)  
menu\_cksjb.add\_command(label=**"退货表"**, command=cksjb.\_xsth)  
menu\_cksjb.add\_command(label=**"员工表"**, command=cksjb.\_xsyg)  
menu\_cksjb.add\_command(label=**"进货商表"**,command=cksjb.\_jhsb)  
  
png = PhotoImage(file=**'0.png'**)  
Label(image=png).pack()  
  
top2.config(menu=mainmenu)  
top2.withdraw()  
  
  
**def** \_load():  
 char = **"SELECT 密码 FROM userdb WHERE 用户名 = '%s'"** % usertext.get()  
 findp = excuate.cal\_r(char)  
 findu = excuate.cal\_r(**"SELECT 用户名 FROM userdb"**)  
 y = []  
 **for** x **in** findu:  
 y.append(x[0])  
  
 **if** usertext.get() **in** y:  
 **if** pwtext.get() == findp[0][0]:  
 message.showinfo(**'提示'**, **'登陆成功！'**)  
 usertext.delete(0, END)  
 pwtext.delete(0, END)  
 top.destroy()  
 top2.update()  
 top2.deiconify()  
 **else**:  
 message.showerror(**'提示'**,**'用户名或密码错误！'**)  
 usertext.delete(0, END)  
 pwtext.delete(0, END)  
 **else**:  
 message.showerror(**'提示'**, **'用户名或密码错误！'**)  
 usertext.delete(0, END)  
 pwtext.delete(0, END)  
  
 change\_win = Toplevel(top)  
 change\_win.title(**'修改密码'**)  
 change\_win.geometry(**"300x190+500+300"**)  
 Label(change\_win, text=**"您的账户名称："**).place(x=10, y=20)  
 Label(change\_win, text=**"输入您的密码："**).place(x=10, y=50)  
 Label(change\_win, text=**"输入新的密码："**).place(x=10, y=80)  
 Label(change\_win, text=**"再次输入密码："**).place(x=10, y=110)  
 Button(change\_win, text=**"确定"**, width=30, command=\_change).place(x=16, y=152)  
 change\_user = Entry(change\_win)  
 change\_pw\_b = Entry(change\_win)  
 change\_pw\_n1 = Entry(change\_win)  
 change\_pw\_n2 = Entry(change\_win)  
 change\_user.place(x=100, y=20)  
 change\_pw\_b.place(x=100, y=50)  
 change\_pw\_n1.place(x=100, y=80)  
 change\_pw\_n2.place(x=100, y=110)  
  
top = Tk()  
top.geometry(**"400x220+500+300"**)  
top.title(**'销售管理信息系统登陆'**)  
  
lab1 = Label(top,  
 text=**"用户名："**,  
 font=(**"黑体"**, 14),  
 width=10,  
 height=2,  
 anchor=**"e"**)  
lab2 = Label(top,  
 text=**"密 码："**,  
 font=(**"黑体"**, 14),  
 width=10,  
 anchor=**"e"**,  
 height=2)  
lab3 = Label(top,  
 text=**"销售管理信息系统"**,  
 font=(**"黑体"**, 30),  
 height=3  
 ).pack()  
usertext = Entry(top)  
pwtext = Entry(top, show=**'\*'**)  
  
load = Button(top, text=**"登陆"**, width=6, height=3, command=\_load)  
addu = Button(top, text=**"添加用户"**, command=\_adduser)  
repw = Button(top, text=**"修改密码"**, command=\_changepw)  
  
lab1.place(x=22,y=100)  
lab2.place(x=22,y=130)  
usertext.place(x=102, y=105)  
pwtext.place(x=102, y=135)  
load.place(x=299, y=108)  
addu.place(x=80, y=172)  
repw.place(x=255, y=172)  
  
top.mainloop()

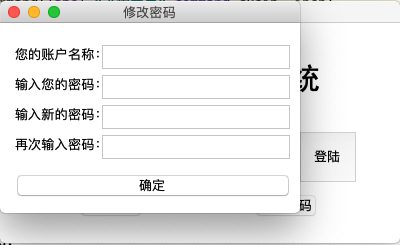
在上述代码中添加如下函数代码以实现添加用户：

**def** \_adduser():  
 **def** \_add():  
 **if** add\_pw.get() != add\_pw2.get():  
 message.showerror(**'错误'**, **'两次密码输入不一致！'**)  
 add\_pw.delete(0, END)  
 add\_pw2.delete(0, END)  
 **else**:  
 char = **"INSERT INTO userdb(用户名,密码) VALUES ('%s','%s')"** % (add\_user.get(),add\_pw.get())  
 excuate.cal\_nr(char)  
 message.showinfo(**'提示'**, **'注册成功！'**)  
 add\_win.destroy()  
  
 add\_win = Toplevel(top)  
 add\_win.title(**'用户注册'**)  
 add\_win.geometry(**"300x150+500+300"**)  
 Label(add\_win, text=**"新账户名称："**).place(x=12, y=20)  
 Label(add\_win, text=**"请输入密码："**).place(x=12, y=50)  
 Label(add\_win, text=**"请确认密码："**).place(x=12, y=80)  
 Button(add\_win, text=**"确定"**, width=30, command=\_add).place(x=16, y=120)  
 add\_user = Entry(add\_win)  
 add\_pw = Entry(add\_win)  
 add\_pw2 = Entry(add\_win)  
 add\_user.place(x=100, y=20)  
 add\_pw.place(x=100, y=50)  
 add\_pw2.place(x=100, y=80)



用户注册

在主体代码中添加以下函数代码以实现更改密码操作：  
**def** \_changepw():  
 **def** \_change():  
 char = **"SELECT 密码 FROM userdb WHERE 用户名 = '%s'"** % change\_user.get()  
 findp = excuate.cal\_r(char)  
 findu = excuate.cal\_r(**"SELECT 用户名 FROM userdb"**)  
 y = []  
 **for** x **in** findu:  
 y.append(x[0])  
  
 **if** change\_user.get() **in** y:  
 **if** change\_pw\_b.get() == findp[0][0]:  
 **if** change\_pw\_n1.get() != change\_pw\_n2.get():  
 message.showerror(**'错误'**, **'两次密码输入不一致！'**)  
 change\_pw\_n1.delete(0, END)  
 change\_pw\_n2.delete(0, END)  
 **else**:  
 char = **"UPDATE userdb SET 密码 = '%s' WHERE 用户名 = '%s'"** % (change\_pw\_n1.get(),change\_user.get())  
 excuate.cal\_nr(char)  
 message.showinfo(**'提示'**, **'修改成功！'**)  
 change\_win.destroy()  
 **else**:  
 message.showerror(**'错误'**, **'原密码输入错误！'**)  
 change\_pw\_b.delete(0, END)  
 **else**:  
 message.showerror(**'错误'**, **'该用户不存在！'**)  
 change\_user.delete(0, END)  
 change\_pw\_b.delete(0, END)



修改密码

2.实现对MySql数据库的连接

通过导入mysql.connector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：

*# -\*-coding:gbk-\*-***import** mysql.connector  
  
**def** cal\_r(char):  
 mydb = mysql.connector.connect(  
 host=**"localhost"**,  
 user=**"root"**,  
 passwd=**"1qw23e00"**,  
 database=**"sellSystem"**)  
  
 mycursor = mydb.cursor()  
 mycursor.autocommit = **True** mycursor.execute(char)  
  
 **return** mycursor.fetchall()  
  
  
**def** cal\_nr(char):  
 mydb = mysql.connector.connect(  
 host=**"localhost"**,  
 user=**"root"**,  
 passwd=**"1qw23e00"**,  
 database=**"sellSystem"**)  
  
 mycursor = mydb.cursor()  
 mycursor.autocommit = **True** mycursor.execute(char)  
 mydb.commit()  
  
**def** fet\_list(column, db):  
 find = cal\_r(**"SELECT %s FROM %s"** % (column, db))  
 y = []  
 **for** x **in** find:  
 y.append(x[0])  
 **return** y

1. 实现交易管理

需要在文件开头声明导入以下文件与模块：

*# -\*-coding:gbk-\*-***from** tkinter **import** \*  
**import** tkinter.messagebox **as** message  
**import** tkinter.ttk **as** ttk  
**import** excuate

①进货登记



添加商品入库

**def** \_jhdj():  
 **def** \_clearsp():  
 text\_spmc.delete(0, END)  
 text\_sccs.delete(0, END)  
 text\_xh.delete(0, END)  
 text\_dj.delete(0, END)  
 text\_sl.delete(0, END)  
 text\_j\_n.delete(0, END)  
 text\_j\_y.delete(0, END)  
 text\_j\_r.delete(0, END)  
 text\_ywybh.delete(0, END)  
  
 **def** \_clearcs():  
 text\_csmc.delete(0, END)  
 text\_frdb.delete(0, END)  
 text\_dh.delete(0, END)  
 text\_csdz.delete(0, END)  
  
 **def** \_insertsp():  
 **if** text\_sccs.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请选择生产厂商！'**)  
 **return  
 if** text\_spmc.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入商品名称！'**)  
 **return  
 if** text\_xh.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入型号！'**)  
 **return  
 if** text\_dj.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入单价！'**)  
 **return  
 if** text\_sl.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入数量！'**)  
 **return  
 if** text\_j\_n.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入进货年份！'**)  
 **return  
 if** text\_j\_y.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请输入进货月份！'**)  
 **return  
 if** text\_j\_r.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请输入进货日！'**)  
 **return  
 if** text\_ywybh.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请选择业务员编号！'**)  
 **return** char = **"""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 ('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')"""** % (  
 text\_sccs.get(),text\_spmc.get(),text\_xh.get(),text\_dj.get(),text\_sl.get(),  
 int(text\_dj.get())\*int(text\_sl.get()),text\_j\_n.get(),text\_j\_y.get(),text\_j\_r.get(),text\_ywybh.get())  
 excuate.cal\_nr(char)  
 message.showinfo(**'提示'**, **'插入成功！'**)  
 \_clearsp()  
  
 **def** \_insertcs():  
 char = **"""INSERT INTO manufacturer(厂商名称,法人代表,厂商地址,电话)  
 VALUES ('%s','%s','%s','%s')"""** % (  
 text\_csmc.get(),text\_frdb.get(),text\_csdz.get(),text\_dh.get())  
 excuate.cal\_nr(char)  
 message.showinfo(**'提示'**, **'插入成功！'**)  
 \_clearcs()  
   
 jy\_jh\_win = Tk()  
 jy\_jh\_win.geometry(**"450x490+600+200"**)  
 jy\_jh\_win.title(**'添加商品入库'**)  
  
 frame\_tjsp = LabelFrame(jy\_jh\_win, text=**"添加商品"**, labelanchor=**"nw"**)  
 frame\_tjsp.place(relx=0.05, rely=0.02, relwidth=0.9, relheight=0.45)  
  
 Label(frame\_tjsp, text=**"商品编号："**).place(relx=0.03, rely=0.03)  
 text\_spbh = Entry(frame\_tjsp, width=8)  
 text\_spbh.insert(0, **'自动添加'**)  
 text\_spbh.place(relx=0.28, rely=0.03)  
  
 Label(frame\_tjsp, text=**"商品名称："**).place(relx=0.49, rely=0.03)  
 text\_spmc = Entry(frame\_tjsp, width=8)  
 text\_spmc.place(relx=0.69, rely=0.03)  
  
 Label(frame\_tjsp, text=**"生产厂商："**).place(relx=0.03, rely=0.2)  
 val1 = excuate.fet\_list(**"厂商名称"**,**"manufacturer"**)  
 text\_sccs = ttk.Combobox(frame\_tjsp, width=6, values=val1)  
 text\_sccs.place(relx=0.28, rely=0.2)  
  
 Label(frame\_tjsp, text=**"型号："**).place(relx=0.49, rely=0.2)  
 text\_xh = Entry(frame\_tjsp, width=8)  
 text\_xh.place(relx=0.69, rely=0.2)  
  
 Label(frame\_tjsp, text=**"单价："**).place(relx=0.03, rely=0.37)  
 text\_dj = Entry(frame\_tjsp, width=8)  
 text\_dj.place(relx=0.28, rely=0.37)  
  
 Label(frame\_tjsp, text=**"数量："**).place(relx=0.49, rely=0.37)  
 text\_sl = Entry(frame\_tjsp, width=8)  
 text\_sl.place(relx=0.69, rely=0.37)  
  
 Label(frame\_tjsp, text=**"进货日期："**).place(relx=0.03, rely=0.53)  
 text\_j\_n = Entry(frame\_tjsp, width=3)  
 text\_j\_n.place(relx=0.28, rely=0.53)  
 Label(frame\_tjsp, text=**"年"**).place(relx=0.38, rely=0.53)  
 text\_j\_y = Entry(frame\_tjsp, width=3)  
 text\_j\_y.place(relx=0.46, rely=0.53)  
 Label(frame\_tjsp, text=**"月"**).place(relx=0.58, rely=0.53)  
 text\_j\_r = Entry(frame\_tjsp, width=3)  
 text\_j\_r.place(relx=0.64, rely=0.53)  
 Label(frame\_tjsp, text=**"日"**).place(relx=0.78, rely=0.53)  
  
 Label(frame\_tjsp, text=**"业务员编号："**).place(relx=0.03, rely=0.69)  
 val2 = excuate.fet\_list(**"员工编号"**,**"employee"**)  
 text\_ywybh = ttk.Combobox(frame\_tjsp, width=4, values=val2)  
 text\_ywybh.place(relx=0.31, rely=0.69)  
  
 Label(frame\_tjsp, text=**"总金额："**).place(relx=0.49, rely=0.69)  
 text\_zje = Entry(frame\_tjsp, width=6)  
 text\_zje.place(relx=0.69, rely=0.69)  
 text\_zje.insert(0, **'自动计算'**)  
  
 tjrk\_but = Button(frame\_tjsp, text=**"添加入库"**, width=8, command=\_insertsp)  
 tjrk\_but.place(relx=0.2, rely=0.85)  
  
 tjrk\_but = Button(frame\_tjsp, text=**"清空重填"**, width=8, command=\_clearsp)  
 tjrk\_but.place(relx=0.58, rely=0.85)  
  
  
 frame\_tjcs = LabelFrame(jy\_jh\_win, text=**"添加厂商"**, labelanchor=**"nw"**)  
 frame\_tjcs.place(relx=0.05, rely=0.50, relwidth=0.9, relheight=0.45)  
  
 Label(frame\_tjcs, text=**"厂商名称："**).place(relx=0.03, rely=0.03)  
 text\_csmc = Entry(frame\_tjcs, width=24)  
 text\_csmc.place(relx=0.31, rely=0.03)  
  
 Label(frame\_tjcs, text=**"法人代表："**).place(relx=0.03, rely=0.23)  
 text\_frdb = Entry(frame\_tjcs, width=24)  
 text\_frdb.place(relx=0.31, rely=0.23)  
  
 Label(frame\_tjcs, text=**"电话："**).place(relx=0.03, rely=0.47)  
 text\_dh = Entry(frame\_tjcs, width=24)  
 text\_dh.place(relx=0.31, rely=0.47)  
  
 Label(frame\_tjcs, text=**"厂商地址："**).place(relx=0.03, rely=0.68)  
 text\_csdz = Entry(frame\_tjcs, width=24)  
 text\_csdz.place(relx=0.31, rely=0.68)  
  
 tjrk\_but = Button(frame\_tjcs, text=**"添加入库"**, width=8, command=\_insertcs)  
 tjrk\_but.place(relx=0.2, rely=0.85)  
  
 tjrk\_but = Button(frame\_tjcs, text=**"清空重填"**, width=8, command=\_clearcs)  
 tjrk\_but.place(relx=0.58, rely=0.85)

②销售登记



销售登记

**def** \_xsdj():  
 **def** \_clearsp():  
 text\_spmc.delete(0, END)  
 text\_sccs.delete(0, END)  
 text\_xh.delete(0, END)  
 text\_dj.delete(0, END)  
 text\_sl.delete(0, END)  
 text\_j\_n.delete(0, END)  
 text\_j\_y.delete(0, END)  
 text\_j\_r.delete(0, END)  
 text\_ywybh.delete(0, END)  
  
 **def** \_sellsp():  
 **if** text\_sccs.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请选择生产厂商！'**)  
 **return  
 if** text\_spmc.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入商品名称！'**)  
 **return  
 if** text\_xh.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入型号！'**)  
 **return  
 if** text\_dj.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入单价！'**)  
 **return  
 if** text\_sl.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入数量！'**)  
 **return  
 if** text\_j\_n.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入销售年份！'**)  
 **return  
 if** text\_j\_y.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请输入销售月份！'**)  
 **return  
 if** text\_j\_r.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请输入销售日！'**)  
 **return  
 if** text\_ywybh.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请选择业务员编号！'**)  
 **return** val3 = excuate.fet\_list(**"商品名"**, **"goods"**)  
 **if** text\_spmc.get() **in** val3:  
 num\_goods = excuate.cal\_r(**"SELECT 数量 FROM goods WHERE 商品名 = '%s'"** % text\_spmc.get())  
 num = num\_goods[0][0] - int(text\_sl.get())  
 **if** num >= 0:  
 char1 = **"""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 ('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')"""** % (  
 text\_sccs.get(), text\_spmc.get(), text\_xh.get(), text\_dj.get(), text\_sl.get(),  
 int(text\_dj.get()) \* int(text\_sl.get()), text\_j\_n.get(), text\_j\_y.get(), text\_j\_r.get(),  
 text\_ywybh.get())  
 excuate.cal\_nr(char1)  
  
 char2 = **"UPDATE goods SET 数量 = '%s' WHERE 商品名 = '%s'"** % (  
 num, text\_spmc.get())  
 excuate.cal\_nr(char2)  
  
 message.showinfo(**'提示'**, **'插入销售记录成功！'**)  
 \_clearsp()  
 **else**:  
 message.showerror(**'错误'**, **'剩余货品数量不足！'**)  
 text\_sl.delete(0, END)  
 **else**:  
 message.showerror(**'错误'**, **'没有该种商品！'**)  
 \_clearsp()  
  
 jy\_xs\_win = Tk()  
 jy\_xs\_win.geometry(**"450x290+600+200"**)  
 jy\_xs\_win.title(**"销售登记"**)  
  
 Label(jy\_xs\_win, text=**"商品编号："**).place(relx=0.03, rely=0.03)  
 text\_spbh = Entry(jy\_xs\_win, width=8)  
 text\_spbh.insert(0, **'自动添加'**)  
 text\_spbh.place(relx=0.28, rely=0.03)  
  
 Label(jy\_xs\_win, text=**"商品名称："**).place(relx=0.49, rely=0.03)  
 text\_spmc = Entry(jy\_xs\_win, width=8)  
 text\_spmc.place(relx=0.69, rely=0.03)  
  
 Label(jy\_xs\_win, text=**"生产厂商："**).place(relx=0.03, rely=0.2)  
 val1 = excuate.fet\_list(**"厂商名称"**, **"manufacturer"**)  
 text\_sccs = ttk.Combobox(jy\_xs\_win, width=6, values=val1)  
 text\_sccs.place(relx=0.28, rely=0.2)  
  
 Label(jy\_xs\_win, text=**"型号："**).place(relx=0.49, rely=0.2)  
 text\_xh = Entry(jy\_xs\_win, width=8)  
 text\_xh.place(relx=0.69, rely=0.2)  
  
 Label(jy\_xs\_win, text=**"单价："**).place(relx=0.03, rely=0.37)  
 text\_dj = Entry(jy\_xs\_win, width=8)  
 text\_dj.place(relx=0.28, rely=0.37)  
  
 Label(jy\_xs\_win, text=**"数量："**).place(relx=0.49, rely=0.37)  
 text\_sl = Entry(jy\_xs\_win, width=8)  
 text\_sl.place(relx=0.69, rely=0.37)  
  
 Label(jy\_xs\_win, text=**"销售日期："**).place(relx=0.03, rely=0.53)  
 text\_j\_n = Entry(jy\_xs\_win, width=3)  
 text\_j\_n.place(relx=0.28, rely=0.53)  
 Label(jy\_xs\_win, text=**"年"**).place(relx=0.38, rely=0.53)  
 text\_j\_y = Entry(jy\_xs\_win, width=3)  
 text\_j\_y.place(relx=0.46, rely=0.53)  
 Label(jy\_xs\_win, text=**"月"**).place(relx=0.58, rely=0.53)  
 text\_j\_r = Entry(jy\_xs\_win, width=3)  
 text\_j\_r.place(relx=0.64, rely=0.53)  
 Label(jy\_xs\_win, text=**"日"**).place(relx=0.78, rely=0.53)  
  
 Label(jy\_xs\_win, text=**"业务员编号："**).place(relx=0.03, rely=0.69)  
 val2 = excuate.fet\_list(**"员工编号"**, **"employee"**)  
 text\_ywybh = ttk.Combobox(jy\_xs\_win, width=4, values=val2)  
 text\_ywybh.place(relx=0.31, rely=0.69)  
  
 Label(jy\_xs\_win, text=**"总金额："**).place(relx=0.49, rely=0.69)  
 text\_zje = Entry(jy\_xs\_win, width=6)  
 text\_zje.place(relx=0.69, rely=0.69)  
 text\_zje.insert(0, **'自动计算'**)  
  
 tjrk\_but = Button(jy\_xs\_win, text=**"销售"**, width=8, command=\_sellsp)  
 tjrk\_but.place(relx=0.2, rely=0.85)  
  
 tjrk\_but = Button(jy\_xs\_win, text=**"清空重填"**, width=8, command=\_clearsp)  
 tjrk\_but.place(relx=0.58, rely=0.85)

③退货登记



退货登记

**def** \_thdj():  
 **def** \_clearsp():  
 text\_spmc.delete(0, END)  
 text\_sccs.delete(0, END)  
 text\_xh.delete(0, END)  
 text\_dj.delete(0, END)  
 text\_sl.delete(0, END)  
 text\_j\_n.delete(0, END)  
 text\_j\_y.delete(0, END)  
 text\_j\_r.delete(0, END)  
 text\_ywybh.delete(0, END)  
  
 **def** \_retreatsp():  
 **if** text\_sccs.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请选择生产厂商！'**)  
 **return  
 if** text\_spmc.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入商品名称！'**)  
 **return  
 if** text\_xh.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入型号！'**)  
 **return  
 if** text\_dj.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入单价！'**)  
 **return  
 if** text\_sl.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入数量！'**)  
 **return  
 if** text\_j\_n.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**,**'请输入退货年份！'**)  
 **return  
 if** text\_j\_y.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请输入退货月份！'**)  
 **return  
 if** text\_j\_r.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请输入退货日！'**)  
 **return  
 if** text\_ywybh.get():  
 **pass  
 else**:  
 message.showerror(**'提示'**, **'请选择业务员编号！'**)  
 **return** val3 = excuate.fet\_list(**"商品名"**, **"sell"**)  
 **if** text\_spmc.get() **in** val3:  
 val4 = excuate.fet\_list(**"商品名"**, **"retreat"**)  
 **if** text\_spmc.get() **in** val4:  
 num\_retreat = excuate.cal\_r(**"SELECT 数量 FROM retreat WHERE 商品名 = '%s'"** % text\_spmc.get())  
 numm = num\_retreat[0][0] + int(text\_sl.get())  
 char3 = **"UPDATE retreat SET 数量 = '%s' WHERE 商品名 = '%s'"** % (  
 numm,text\_spmc.get())  
 excuate.cal\_nr(char3)  
 **else**:  
 char1 = **"""INSERT INTO retreat (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 ('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')"""** % (  
 text\_sccs.get(), text\_spmc.get(), text\_xh.get(), text\_dj.get(), text\_sl.get(),  
 int(text\_dj.get()) \* int(text\_sl.get()), text\_j\_n.get(), text\_j\_y.get(), text\_j\_r.get(),  
 text\_ywybh.get())  
 excuate.cal\_nr(char1)  
  
 num\_sell = excuate.cal\_r(**"SELECT 数量 FROM sell WHERE 商品名 = '%s'"** % text\_spmc.get())  
 num = num\_sell[0][0] - int(text\_sl.get())  
  
 char2 = **"UPDATE sell SET 数量 = '%s' WHERE 商品名 = '%s'"** % (  
 num, text\_spmc.get())  
 excuate.cal\_nr(char2)  
  
 num\_goods = excuate.cal\_r(**"SELECT 数量 FROM goods WHERE 商品名 = '%s'"** % text\_spmc.get())  
 nummb = num\_goods[0][0] + int(text\_sl.get())  
  
 char2 = **"UPDATE goods SET 数量 = '%s' WHERE 商品名 = '%s'"** % (  
 nummb, text\_spmc.get())  
 excuate.cal\_nr(char2)  
  
 message.showinfo(**'提示'**, **'插入退货记录成功！'**)  
 \_clearsp()  
 **else**:  
 message.showerror(**'错误'**, **'没有售出过该种商品！'**)  
 \_clearsp()  
  
  
  
 jy\_th\_win = Tk()  
 jy\_th\_win.geometry(**"450x290+600+200"**)  
 jy\_th\_win.title(**"退货登记"**)  
  
 Label(jy\_th\_win, text=**"商品编号："**).place(relx=0.03, rely=0.03)  
 text\_spbh = Entry(jy\_th\_win, width=8)  
 text\_spbh.insert(0, **'自动添加'**)  
 text\_spbh.place(relx=0.28, rely=0.03)  
  
 Label(jy\_th\_win, text=**"商品名称："**).place(relx=0.49, rely=0.03)  
 text\_spmc = Entry(jy\_th\_win, width=8)  
 text\_spmc.place(relx=0.69, rely=0.03)  
  
 Label(jy\_th\_win, text=**"生产厂商："**).place(relx=0.03, rely=0.2)  
 val1 = excuate.fet\_list(**"厂商名称"**, **"manufacturer"**)  
 text\_sccs = ttk.Combobox(jy\_th\_win, width=6, values=val1)  
 text\_sccs.place(relx=0.28, rely=0.2)  
  
 Label(jy\_th\_win, text=**"型号："**).place(relx=0.49, rely=0.2)  
 text\_xh = Entry(jy\_th\_win, width=8)  
 text\_xh.place(relx=0.69, rely=0.2)  
  
 Label(jy\_th\_win, text=**"单价："**).place(relx=0.03, rely=0.37)  
 text\_dj = Entry(jy\_th\_win, width=8)  
 text\_dj.place(relx=0.28, rely=0.37)  
  
 Label(jy\_th\_win, text=**"数量："**).place(relx=0.49, rely=0.37)  
 text\_sl = Entry(jy\_th\_win, width=8)  
 text\_sl.place(relx=0.69, rely=0.37)  
  
 Label(jy\_th\_win, text=**"退货日期："**).place(relx=0.03, rely=0.53)  
 text\_j\_n = Entry(jy\_th\_win, width=3)  
 text\_j\_n.place(relx=0.28, rely=0.53)  
 Label(jy\_th\_win, text=**"年"**).place(relx=0.38, rely=0.53)  
 text\_j\_y = Entry(jy\_th\_win, width=3)  
 text\_j\_y.place(relx=0.46, rely=0.53)  
 Label(jy\_th\_win, text=**"月"**).place(relx=0.58, rely=0.53)  
 text\_j\_r = Entry(jy\_th\_win, width=3)  
 text\_j\_r.place(relx=0.64, rely=0.53)  
 Label(jy\_th\_win, text=**"日"**).place(relx=0.78, rely=0.53)  
  
 Label(jy\_th\_win, text=**"业务员编号："**).place(relx=0.03, rely=0.69)  
 val2 = excuate.fet\_list(**"员工编号"**, **"employee"**)  
 text\_ywybh = ttk.Combobox(jy\_th\_win, width=4, values=val2)  
 text\_ywybh.place(relx=0.31, rely=0.69)  
  
 Label(jy\_th\_win, text=**"总金额："**).place(relx=0.49, rely=0.69)  
 text\_zje = Entry(jy\_th\_win, width=6)  
 text\_zje.place(relx=0.69, rely=0.69)  
 text\_zje.insert(0, **'自动计算'**)  
  
 tjrk\_but = Button(jy\_th\_win, text=**"退货"**, width=8, command=\_retreatsp)  
 tjrk\_but.place(relx=0.2, rely=0.85)  
  
 tjrk\_but = Button(jy\_th\_win, text=**"清空重填"**, width=8, command=\_clearsp)  
 tjrk\_but.place(relx=0.58, rely=0.85)

1. 实现进货统计

需要在文件开头声明导入以下文件与模块：

*# -\*-coding:gbk-\*-***from** tkinter **import** \*  
**import** tkinter.ttk **as** ttk  
**import** excuate  
**import** time

①今日进货量



今日进货

**def** \_jrjh():  
 jh\_jr\_win = Tk()  
 jh\_jr\_win.title(**"今日进货"**)  
 jh\_jr\_win.geometry(**"600x340+300+250"**)  
  
 *# 查询结果窗口* tree\_jr = ttk.Treeview(jh\_jr\_win, show=**"headings"**)  
 tree\_jr.place(relx=0.01, rely=0.01)  
  
 tree\_jr[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**, **"单价"**, **"数量"**, **"总金额"**, **"业务员编号"**)  
  
 tree\_jr.column(**"商品编号"**, width=62)  
 tree\_jr.column(**"生产厂商"**, width=75)  
 tree\_jr.column(**"商品名"**, width=75)  
 tree\_jr.column(**"型号"**, width=75)  
 tree\_jr.column(**"单价"**, width=75)  
 tree\_jr.column(**"数量"**, width=75)  
 tree\_jr.column(**"总金额"**, width=75)  
 tree\_jr.column(**"业务员编号"**, width=75)  
  
 tree\_jr.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_jr.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_jr.heading(**"商品名"**, text=**"商品名"**)  
 tree\_jr.heading(**"型号"**, text=**"型号"**)  
 tree\_jr.heading(**"单价"**, text=**"单价"**)  
 tree\_jr.heading(**"数量"**, text=**"数量"**)  
 tree\_jr.heading(**"总金额"**, text=**"总金额"**)  
 tree\_jr.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号, 单价, 数量, 总金额, 业务员编号  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s' AND 进货日 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s' AND 进货日 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:8]  
 tree\_jr.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(jh\_jr\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.62)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商进货总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=100)  
 tree\_cs.column(**"各厂商进货总金额"**, width=130)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商进货总金额"**, text=**"各厂商进货总金额"**)  
  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s' AND 进货日 = '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s' AND 进货日 = '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(jh\_jr\_win, text=**"进货总金额："**).place(relx=0.45, rely=0.76)  
 text\_jhzje = Entry(jh\_jr\_win)  
 text\_jhzje.place(relx=0.6, rely=0.75)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s' AND 进货日 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 jh\_jr\_win.mainloop()

②本月进货量



本月进货

**def** \_byjh():  
 jh\_by\_win = Tk()  
 jh\_by\_win.title(**"本月进货"**)  
 jh\_by\_win.geometry(**"600x340+300+250"**)  
  
 *# 查询结果窗口* tree\_by = ttk.Treeview(jh\_by\_win, show=**"headings"**)  
 tree\_by.place(relx=0.01, rely=0.01)  
  
 tree\_by[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**, **"单价"**, **"数量"**, **"总金额"**, **"业务员编号"**)  
  
 tree\_by.column(**"商品编号"**, width=62)  
 tree\_by.column(**"生产厂商"**, width=75)  
 tree\_by.column(**"商品名"**, width=75)  
 tree\_by.column(**"型号"**, width=75)  
 tree\_by.column(**"单价"**, width=75)  
 tree\_by.column(**"数量"**, width=75)  
 tree\_by.column(**"总金额"**, width=75)  
 tree\_by.column(**"业务员编号"**, width=75)  
  
 tree\_by.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_by.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_by.heading(**"商品名"**, text=**"商品名"**)  
 tree\_by.heading(**"型号"**, text=**"型号"**)  
 tree\_by.heading(**"单价"**, text=**"单价"**)  
 tree\_by.heading(**"数量"**, text=**"数量"**)  
 tree\_by.heading(**"总金额"**, text=**"总金额"**)  
 tree\_by.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号, 单价, 数量, 总金额, 业务员编号  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime())  
 )  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:8]  
 tree\_by.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(jh\_by\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.62)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商进货总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=100)  
 tree\_cs.column(**"各厂商进货总金额"**, width=130)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商进货总金额"**, text=**"各厂商进货总金额"**)  
  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(jh\_by\_win, text=**"进货总金额："**).place(relx=0.45, rely=0.76)  
 text\_jhzje = Entry(jh\_by\_win)  
 text\_jhzje.place(relx=0.6, rely=0.75)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 jh\_by\_win.mainloop()

③本季度进货量



本季度进货

**def** \_jdjh():  
 jh\_jd\_win = Tk()  
 jh\_jd\_win.title(**"本季度进货"**)  
 jh\_jd\_win.geometry(**"600x340+300+250"**)  
  
 *# 查询结果窗口* tree\_by = ttk.Treeview(jh\_jd\_win, show=**"headings"**)  
 tree\_by.place(relx=0.01, rely=0.01)  
  
 tree\_by[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**, **"单价"**, **"数量"**, **"总金额"**, **"业务员编号"**)  
  
 tree\_by.column(**"商品编号"**, width=62)  
 tree\_by.column(**"生产厂商"**, width=75)  
 tree\_by.column(**"商品名"**, width=75)  
 tree\_by.column(**"型号"**, width=75)  
 tree\_by.column(**"单价"**, width=75)  
 tree\_by.column(**"数量"**, width=75)  
 tree\_by.column(**"总金额"**, width=75)  
 tree\_by.column(**"业务员编号"**, width=75)  
  
 tree\_by.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_by.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_by.heading(**"商品名"**, text=**"商品名"**)  
 tree\_by.heading(**"型号"**, text=**"型号"**)  
 tree\_by.heading(**"单价"**, text=**"单价"**)  
 tree\_by.heading(**"数量"**, text=**"数量"**)  
 tree\_by.heading(**"总金额"**, text=**"总金额"**)  
 tree\_by.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 A = 0  
 B = 0  
 C = int(time.strftime(**"%m"**, time.localtime()))  
 **if** C **in** range(1, 4):  
 A = 1  
 B = 3  
 **elif** C **in** range(4, 7):  
 A = 4  
 B = 6  
 **elif** C **in** range(7, 10):  
 A = 7  
 B = 9  
 **elif** C **in** range(10, 13):  
 A = 10  
 B = 12  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号, 单价, 数量, 总金额, 业务员编号  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 >= '%s' AND 进货月 <= '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM goods  
 WHERE 进货年 = '%s' AND 进货月 >= '%s' AND 进货月 <= '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:8]  
 tree\_by.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(jh\_jd\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.62)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商进货总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=100)  
 tree\_cs.column(**"各厂商进货总金额"**, width=130)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商进货总金额"**, text=**"各厂商进货总金额"**)  
  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 >= '%s' AND 进货月 <= '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 >= '%s' AND 进货月 <= '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(jh\_jd\_win, text=**"进货总金额："**).place(relx=0.45, rely=0.76)  
 text\_jhzje = Entry(jh\_jd\_win)  
 text\_jhzje.place(relx=0.6, rely=0.75)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s' AND 进货月 >= '%s' AND 进货月 <= '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 jh\_jd\_win.mainloop()

④今年进货量



今年进货

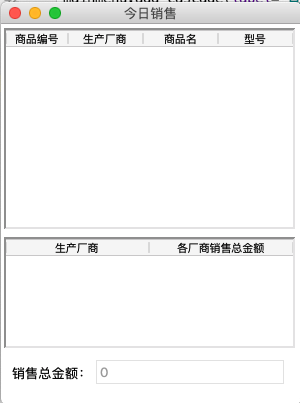
**def** \_jnjh():  
 jh\_jn\_win = Tk()  
 jh\_jn\_win.title(**"今年进货"**)  
 jh\_jn\_win.geometry(**"600x340+300+250"**)  
  
 *# 查询结果窗口* tree\_by = ttk.Treeview(jh\_jn\_win, show=**"headings"**)  
 tree\_by.place(relx=0.01, rely=0.01)  
  
 tree\_by[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**, **"单价"**, **"数量"**, **"总金额"**, **"业务员编号"**)  
  
 tree\_by.column(**"商品编号"**, width=62)  
 tree\_by.column(**"生产厂商"**, width=75)  
 tree\_by.column(**"商品名"**, width=75)  
 tree\_by.column(**"型号"**, width=75)  
 tree\_by.column(**"单价"**, width=75)  
 tree\_by.column(**"数量"**, width=75)  
 tree\_by.column(**"总金额"**, width=75)  
 tree\_by.column(**"业务员编号"**, width=75)  
  
 tree\_by.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_by.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_by.heading(**"商品名"**, text=**"商品名"**)  
 tree\_by.heading(**"型号"**, text=**"型号"**)  
 tree\_by.heading(**"单价"**, text=**"单价"**)  
 tree\_by.heading(**"数量"**, text=**"数量"**)  
 tree\_by.heading(**"总金额"**, text=**"总金额"**)  
 tree\_by.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号, 单价, 数量, 总金额, 业务员编号  
 FROM goods  
 WHERE 进货年 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM goods  
 WHERE 进货年 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:8]  
 tree\_by.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(jh\_jn\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.62)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商进货总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=100)  
 tree\_cs.column(**"各厂商进货总金额"**, width=130)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商进货总金额"**, text=**"各厂商进货总金额"**)  
  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM goods  
 WHERE 进货年 = '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(jh\_jn\_win, text=**"进货总金额："**).place(relx=0.45, rely=0.76)  
 text\_jhzje = Entry(jh\_jn\_win)  
 text\_jhzje.place(relx=0.6, rely=0.75)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM goods  
 WHERE 进货年 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 jh\_jn\_win.mainloop()

1. 实现销售统计

需要在文件开头声明导入以下文件与模块：

*# -\*-coding:gbk-\*-***from** tkinter **import** \*  
**import** tkinter.ttk **as** ttk  
**import** excuate  
**import** time

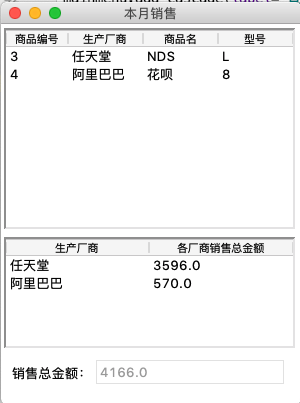
①今日销售量



今日销售

**def** \_jrxs():  
 xs\_jr\_win = Tk()  
 xs\_jr\_win.title(**"今日销售"**)  
 xs\_jr\_win.geometry(**"300x380+300+250"**)  
  
 *# 查询结果窗口* tree\_jr = ttk.Treeview(xs\_jr\_win, show=**"headings"**)  
 tree\_jr.place(relx=0.01, rely=0.01)  
  
 tree\_jr[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**)  
  
 tree\_jr.column(**"商品编号"**, width=62)  
 tree\_jr.column(**"生产厂商"**, width=75)  
 tree\_jr.column(**"商品名"**, width=75)  
 tree\_jr.column(**"型号"**, width=75)  
  
 tree\_jr.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_jr.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_jr.heading(**"商品名"**, text=**"商品名"**)  
 tree\_jr.heading(**"型号"**, text=**"型号"**)  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s' AND 销售日 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s' AND 销售日 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:4]  
 tree\_jr.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(xs\_jr\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.56)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商销售总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=143)  
 tree\_cs.column(**"各厂商销售总金额"**, width=144)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商销售总金额"**, text=**"各厂商销售总金额"**)  
  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s' AND 销售日 = '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s' AND 销售日 = '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(xs\_jr\_win, text=**"销售总金额："**).place(relx=0.02, rely=0.89)  
 text\_jhzje = Entry(xs\_jr\_win)  
 text\_jhzje.place(relx=0.31, rely=0.88)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s' AND 销售日 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()),time.strftime(**"%m"**, time.localtime()),time.strftime(**"%d"**, time.localtime())  
 )  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 xs\_jr\_win.mainloop()

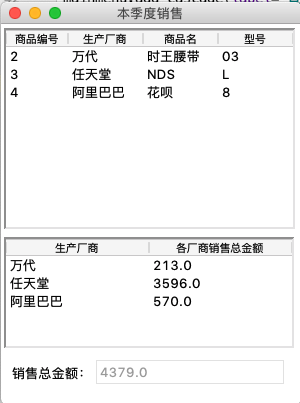
②本月销售量



本月销售

**def** \_byxs():  
 xs\_by\_win = Tk()  
 xs\_by\_win.title(**"本月销售"**)  
 xs\_by\_win.geometry(**"300x380+300+250"**)  
  
 *# 查询结果窗口* tree\_jr = ttk.Treeview(xs\_by\_win, show=**"headings"**)  
 tree\_jr.place(relx=0.01, rely=0.01)  
  
 tree\_jr[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**)  
  
 tree\_jr.column(**"商品编号"**, width=62)  
 tree\_jr.column(**"生产厂商"**, width=75)  
 tree\_jr.column(**"商品名"**, width=75)  
 tree\_jr.column(**"型号"**, width=75)  
  
 tree\_jr.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_jr.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_jr.heading(**"商品名"**, text=**"商品名"**)  
 tree\_jr.heading(**"型号"**, text=**"型号"**)  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:4]  
 tree\_jr.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(xs\_by\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.56)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商销售总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=143)  
 tree\_cs.column(**"各厂商销售总金额"**, width=144)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商销售总金额"**, text=**"各厂商销售总金额"**)  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(xs\_by\_win, text=**"销售总金额："**).place(relx=0.02, rely=0.89)  
 text\_jhzje = Entry(xs\_by\_win)  
 text\_jhzje.place(relx=0.31, rely=0.88)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), time.strftime(**"%m"**, time.localtime())  
 )  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 xs\_by\_win.mainloop()

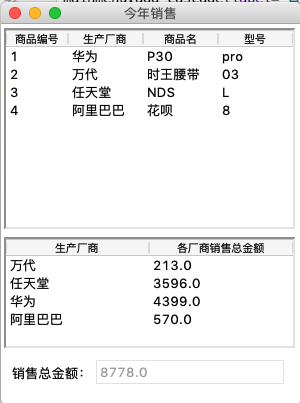
③本季度销售量



本季度销售

**def** \_jdxs():  
 xs\_jd\_win = Tk()  
 xs\_jd\_win.title(**"本季度销售"**)  
 xs\_jd\_win.geometry(**"300x380+300+250"**)  
  
 *# 查询结果窗口* tree\_by = ttk.Treeview(xs\_jd\_win, show=**"headings"**)  
 tree\_by.place(relx=0.01, rely=0.01)  
  
 tree\_by[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**)  
  
 tree\_by.column(**"商品编号"**, width=62)  
 tree\_by.column(**"生产厂商"**, width=75)  
 tree\_by.column(**"商品名"**, width=75)  
 tree\_by.column(**"型号"**, width=75)  
  
 tree\_by.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_by.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_by.heading(**"商品名"**, text=**"商品名"**)  
 tree\_by.heading(**"型号"**, text=**"型号"**)  
  
 A = 0  
 B = 0  
 C = int(time.strftime(**"%m"**, time.localtime()))  
 **if** C **in** range(1, 4):  
 A = 1  
 B = 3  
 **elif** C **in** range(4, 7):  
 A = 4  
 B = 6  
 **elif** C **in** range(7, 10):  
 A = 7  
 B = 9  
 **elif** C **in** range(10, 13):  
 A = 10  
 B = 12  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 >= '%s' AND 销售月 <= '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM sell  
 WHERE 销售年 = '%s' AND 销售月 >= '%s' AND 销售月 <= '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:4]  
 tree\_by.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(xs\_jd\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.56)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商销售总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=143)  
 tree\_cs.column(**"各厂商销售总金额"**, width=143)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商销售总金额"**, text=**"各厂商销售总金额"**)  
  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 >= '%s' AND 销售月 <= '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 >= '%s' AND 销售月 <= '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(xs\_jd\_win, text=**"销售总金额："**).place(relx=0.02, rely=0.89)  
 text\_jhzje = Entry(xs\_jd\_win)  
 text\_jhzje.place(relx=0.31, rely=0.88)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s' AND 销售月 >= '%s' AND 销售月 <= '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()), A, B)  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 xs\_jd\_win.mainloop()

④今年销售量



今年销售

**def** \_jnxs():  
 xs\_jn\_win = Tk()  
 xs\_jn\_win.title(**"今年销售"**)  
 xs\_jn\_win.geometry(**"300x380+300+250"**)  
  
 *# 查询结果窗口* tree\_jr = ttk.Treeview(xs\_jn\_win, show=**"headings"**)  
 tree\_jr.place(relx=0.01, rely=0.01)  
  
 tree\_jr[**"columns"**] = (**"商品编号"**, **"生产厂商"**, **"商品名"**, **"型号"**)  
  
 tree\_jr.column(**"商品编号"**, width=62)  
 tree\_jr.column(**"生产厂商"**, width=75)  
 tree\_jr.column(**"商品名"**, width=75)  
 tree\_jr.column(**"型号"**, width=75)  
  
 tree\_jr.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_jr.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_jr.heading(**"商品名"**, text=**"商品名"**)  
 tree\_jr.heading(**"型号"**, text=**"型号"**)  
  
 char1 = **"""SELECT 商品编号, 生产厂商, 商品名, 型号  
 FROM sell  
 WHERE 销售年 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 find1 = excuate.cal\_r(char1)  
  
 char2 = **"""SELECT count(商品编号) FROM sell  
 WHERE 销售年 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 num\_col = excuate.cal\_r(char2)[0][0]  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:4]  
 tree\_jr.insert(**""**, i, values=z[i])  
  
 *# 分类查询结果窗口* tree\_cs = ttk.Treeview(xs\_jn\_win, show=**"headings"**, height=5)  
 tree\_cs.place(relx=0.01, rely=0.56)  
  
 tree\_cs[**"columns"**] = (**"生产厂商"**, **"各厂商销售总金额"**)  
  
 tree\_cs.column(**"生产厂商"**, width=143)  
 tree\_cs.column(**"各厂商销售总金额"**, width=144)  
  
 tree\_cs.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_cs.heading(**"各厂商销售总金额"**, text=**"各厂商销售总金额"**)  
  
 char3 = **"""SELECT 生产厂商, SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s'  
 GROUP BY 生产厂商"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 find2 = excuate.cal\_r(char3)  
  
 char4 = **"""SELECT COUNT(\*) FROM  
 (SELECT COUNT(DISTINCT 生产厂商)  
 FROM sell  
 WHERE 销售年 = '%s'  
 GROUP BY 生产厂商)a"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 num\_c = excuate.cal\_r(char4)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:2]  
 tree\_cs.insert(**""**, i, values=z[i])  
  
 *# 计算结果文本框* Label(xs\_jn\_win, text=**"销售总金额："**).place(relx=0.02, rely=0.89)  
 text\_jhzje = Entry(xs\_jn\_win)  
 text\_jhzje.place(relx=0.31, rely=0.88)  
  
 char5 = **"""SELECT SUM(总金额)  
 FROM sell  
 WHERE 销售年 = '%s'"""** % (  
 time.strftime(**"%Y"**, time.localtime()))  
 sum\_gcszje = excuate.cal\_r(char5)[0][0]  
  
 **if** num\_col:  
 text\_jhzje.insert(0, sum\_gcszje)  
 text\_jhzje.config(state=DISABLED)  
 **else**:  
 text\_jhzje.insert(0, 0)  
 text\_jhzje.config(state=DISABLED)  
  
 xs\_jn\_win.mainloop()

1. 实现业绩查看

需要在文件开头声明导入以下文件与模块：

*# -\*-coding:gbk-\*-***from** tkinter **import** \*  
**import** tkinter.ttk **as** ttk  
**import** excuate  
**import** tkinter.messagebox **as** message



业绩查看

代码实现如下：

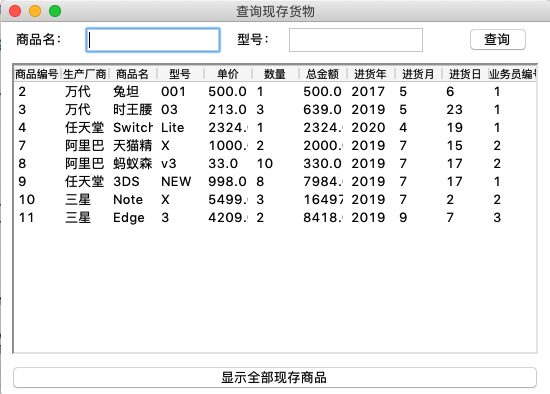
**def** \_yjck():  
 **def** \_yghcx():  
 char1 = **"SELECT 员工姓名 FROM employee WHERE 员工编号 = '%s'"** % text\_ygh.get()  
 find1 = excuate.cal\_r(char1)  
  
 **if** find1:  
 tree\_ygh = ttk.Treeview(yjck\_win, show=**"headings"**, height=15)  
 tree\_ygh.place(relx=0.02, rely=0.11)  
  
 tree\_ygh[**"columns"**] = (**"业务员编号"**, **"员工姓名"**, **"销售总金额"**)  
 tree\_ygh.column(**"业务员编号"**, width=102)  
 tree\_ygh.column(**"员工姓名"**, width=102)  
 tree\_ygh.column(**"销售总金额"**, width=102)  
  
 tree\_ygh.heading(**"业务员编号"**, text=**"业务员编号"**)  
 tree\_ygh.heading(**"员工姓名"**, text=**"员工姓名"**)  
 tree\_ygh.heading(**"销售总金额"**, text=**"销售总金额"**)  
  
 char4 = **"""SELECT 员工编号, 员工姓名, SUM(总金额)  
 FROM employee, sell  
 WHERE 员工编号 = 业务员编号 AND 员工编号 = '%s'  
 GROUP BY 员工编号, 员工姓名  
 ORDER BY 员工编号"""** % text\_ygh.get()  
 find4 = excuate.cal\_r(char4)  
  
 char5 = **"""SELECT COUNT(\*) FROM  
 (SELECT 员工编号, 员工姓名, SUM(总金额)  
 FROM employee, sell  
 WHERE 员工编号 = 业务员编号 AND 员工编号 = '%s'  
 GROUP BY 员工编号, 员工姓名  
 ORDER BY 员工编号)a"""** % text\_ygh.get()  
 num\_c = excuate.cal\_r(char5)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find4[i][0:3]  
 tree\_ygh.insert(**""**, i, values=z[i])  
  
 **else**:  
 message.showerror(**'错误'**, **'无该员工信息！'**)  
  
 **def** \_cxqb():  
 char2 = **"""SELECT 员工编号, 员工姓名, SUM(总金额)  
 FROM employee, sell  
 WHERE 员工编号 = 业务员编号  
 GROUP BY 员工编号, 员工姓名  
 ORDER BY 员工编号"""** find2 = excuate.cal\_r(char2)  
  
 char3 = **"""SELECT COUNT(\*) FROM  
 (SELECT 员工编号, 员工姓名, SUM(总金额)  
 FROM employee, sell  
 WHERE 员工编号 = 业务员编号  
 GROUP BY 员工编号, 员工姓名  
 ORDER BY 员工编号)a"""** num\_c = excuate.cal\_r(char3)[0][0]  
  
 tree\_yg = ttk.Treeview(yjck\_win, show=**"headings"**, height=15)  
 tree\_yg.place(relx=0.02, rely=0.11)  
  
 tree\_yg[**"columns"**] = (**"业务员编号"**, **"员工姓名"**, **"销售总金额"**)  
 tree\_yg.column(**"业务员编号"**, width=102)  
 tree\_yg.column(**"员工姓名"**, width=102)  
 tree\_yg.column(**"销售总金额"**, width=102)  
  
 tree\_yg.heading(**"业务员编号"**, text=**"业务员编号"**)  
 tree\_yg.heading(**"员工姓名"**, text=**"员工姓名"**)  
 tree\_yg.heading(**"销售总金额"**, text=**"销售总金额"**)  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:3]  
 tree\_yg.insert(**""**, i, values=z[i])  
  
 yjck\_win = Tk()  
 yjck\_win.geometry(**"325x375+550+320"**)  
 yjck\_win.title(**'查询员工销售额'**)  
  
 Label(yjck\_win, text=**"员工号："**).place(relx=0.02, rely=0.015)  
  
 text\_ygh = Entry(yjck\_win)  
 text\_ygh.place(relx=0.2, rely=0.01)  
  
 yghcx\_but = Button(yjck\_win, text=**"查询"**, width=5, command=\_yghcx)  
 yghcx\_but.place(relx=0.8, rely=0.018)  
  
 cxqb\_but = Button(yjck\_win, text=**"显示全部员工销售额"**, width=34, command=\_cxqb)  
 cxqb\_but.place(relx=0.02, rely=0.92)  
  
 yjck\_win.mainloop()

1. 实现查看数据表

需要在文件开头声明导入以下文件与模块：

*# -\*-coding:gbk-\*-***from** tkinter **import** \*  
**import** tkinter.messagebox **as** message  
**import** tkinter.ttk **as** ttk  
**import** excuate

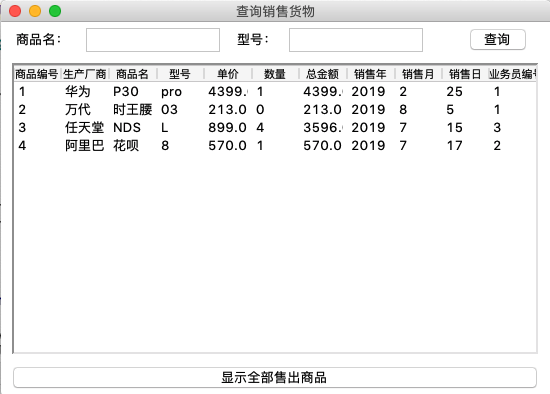
①现存货表



现存货表

**def** \_xchb():  
 **def** \_tjcx():  
 char1 = **"SELECT \* FROM goods WHERE 商品名 = '%s' AND 型号 = '%s'"** % (text\_spm.get(), text\_xh.get())  
 find1 = excuate.cal\_r(char1)  
 **if** find1:  
 char2 = **"SELECT \* FROM goods WHERE 商品名 = '%s' AND 型号 = '%s'"** % (text\_spm.get(), text\_xh.get())  
 find2 = excuate.cal\_r(char2)  
  
 char3 = **"""SELECT COUNT(\*) FROM  
 (SELECT \* FROM goods WHERE 商品名 = '%s' AND 型号 = '%s')a"""** % (text\_spm.get(), text\_xh.get())  
 num\_c = excuate.cal\_r(char3)[0][0]  
  
 tree\_h = ttk.Treeview(yjck\_win, show=**"headings"**, height=15)  
 tree\_h.place(relx=0.02, rely=0.11)  
  
 tree\_h[**"columns"**] = (**"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"**)  
 tree\_h.column(**"商品编号"**, width=47)  
 tree\_h.column(**"生产厂商"**, width=48)  
 tree\_h.column(**"商品名"**, width=48)  
 tree\_h.column(**"型号"**, width=47)  
 tree\_h.column(**"单价"**, width=48)  
 tree\_h.column(**"数量"**, width=47)  
 tree\_h.column(**"总金额"**, width=48)  
 tree\_h.column(**"进货年"**, width=48)  
 tree\_h.column(**"进货月"**, width=47)  
 tree\_h.column(**"进货日"**, width=47)  
 tree\_h.column(**"业务员编号"**, width=47)  
  
 tree\_h.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_h.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_h.heading(**"商品名"**, text=**"商品名"**)  
 tree\_h.heading(**"型号"**, text=**"型号"**)  
 tree\_h.heading(**"单价"**, text=**"单价"**)  
 tree\_h.heading(**"数量"**, text=**"数量"**)  
 tree\_h.heading(**"总金额"**, text=**"总金额"**)  
 tree\_h.heading(**"进货年"**, text=**"进货年"**)  
 tree\_h.heading(**"进货月"**, text=**"进货月"**)  
 tree\_h.heading(**"进货日"**, text=**"进货日"**)  
 tree\_h.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:11]  
 tree\_h.insert(**""**, i, values=z[i])  
 **else**:  
 message.showerror(**'错误'**, **'无现存货物信息！'**)  
  
 **def** \_cxqb():  
 char2 = **"SELECT \* FROM goods"** find2 = excuate.cal\_r(char2)  
  
 char3 = **"""SELECT COUNT(\*) FROM  
 (SELECT \* FROM goods)a"""** num\_c = excuate.cal\_r(char3)[0][0]  
  
 tree\_h = ttk.Treeview(yjck\_win, show=**"headings"**, height=15)  
 tree\_h.place(relx=0.02, rely=0.11)  
  
 tree\_h[**"columns"**] = (**"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"**)  
 tree\_h.column(**"商品编号"**, width=47)  
 tree\_h.column(**"生产厂商"**, width=48)  
 tree\_h.column(**"商品名"**, width=48)  
 tree\_h.column(**"型号"**, width=47)  
 tree\_h.column(**"单价"**, width=48)  
 tree\_h.column(**"数量"**, width=47)  
 tree\_h.column(**"总金额"**, width=48)  
 tree\_h.column(**"进货年"**, width=48)  
 tree\_h.column(**"进货月"**, width=47)  
 tree\_h.column(**"进货日"**, width=47)  
 tree\_h.column(**"业务员编号"**, width=47)  
  
 tree\_h.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_h.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_h.heading(**"商品名"**, text=**"商品名"**)  
 tree\_h.heading(**"型号"**, text=**"型号"**)  
 tree\_h.heading(**"单价"**, text=**"单价"**)  
 tree\_h.heading(**"数量"**, text=**"数量"**)  
 tree\_h.heading(**"总金额"**, text=**"总金额"**)  
 tree\_h.heading(**"进货年"**, text=**"进货年"**)  
 tree\_h.heading(**"进货月"**, text=**"进货月"**)  
 tree\_h.heading(**"进货日"**, text=**"进货日"**)  
 tree\_h.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:11]  
 tree\_h.insert(**""**, i, values=z[i])  
  
 yjck\_win = Tk()  
 yjck\_win.geometry(**"550x375+550+320"**)  
 yjck\_win.title(**'查询现存货物'**)  
  
 Label(yjck\_win, text=**"商品名："**).place(relx=0.019, rely=0.015)  
 Label(yjck\_win, text=**"型号："**).place(relx=0.42, rely=0.015)  
  
 text\_spm = Entry(yjck\_win, width=14)  
 text\_spm.place(relx=0.15, rely=0.01)  
 text\_xh = Entry(yjck\_win, width=14)  
 text\_xh.place(relx=0.52, rely=0.01)  
  
 yghcx\_but = Button(yjck\_win, text=**"查询"**, width=6, command=\_tjcx)  
 yghcx\_but.place(relx=0.85, rely=0.018)  
  
 cxqb\_but = Button(yjck\_win, text=**"显示全部现存商品"**, width=58, command=\_cxqb)  
 cxqb\_but.place(relx=0.02, rely=0.92)  
  
 yjck\_win.mainloop()

②已售商品表



已售商品表

**def** \_scys():  
 **def** \_tjcx():  
 char1 = **"SELECT \* FROM sell WHERE 商品名 = '%s' AND 型号 = '%s'"** % (text\_spm.get(), text\_xh.get())  
 find1 = excuate.cal\_r(char1)  
 **if** find1:  
 char2 = **"SELECT \* FROM sell WHERE 商品名 = '%s' AND 型号 = '%s'"** % (text\_spm.get(), text\_xh.get())  
 find2 = excuate.cal\_r(char2)  
  
 char3 = **"""SELECT COUNT(\*) FROM  
 (SELECT \* FROM sell WHERE 商品名 = '%s' AND 型号 = '%s')a"""** % (text\_spm.get(), text\_xh.get())  
 num\_c = excuate.cal\_r(char3)[0][0]  
  
 tree\_h = ttk.Treeview(yjck\_win, show=**"headings"**, height=15)  
 tree\_h.place(relx=0.02, rely=0.11)  
  
 tree\_h[**"columns"**] = (**"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"**)  
 tree\_h.column(**"商品编号"**, width=47)  
 tree\_h.column(**"生产厂商"**, width=48)  
 tree\_h.column(**"商品名"**, width=48)  
 tree\_h.column(**"型号"**, width=47)  
 tree\_h.column(**"单价"**, width=48)  
 tree\_h.column(**"数量"**, width=47)  
 tree\_h.column(**"总金额"**, width=48)  
 tree\_h.column(**"销售年"**, width=48)  
 tree\_h.column(**"销售月"**, width=47)  
 tree\_h.column(**"销售日"**, width=47)  
 tree\_h.column(**"业务员编号"**, width=47)  
  
 tree\_h.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_h.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_h.heading(**"商品名"**, text=**"商品名"**)  
 tree\_h.heading(**"型号"**, text=**"型号"**)  
 tree\_h.heading(**"单价"**, text=**"单价"**)  
 tree\_h.heading(**"数量"**, text=**"数量"**)  
 tree\_h.heading(**"总金额"**, text=**"总金额"**)  
 tree\_h.heading(**"销售年"**, text=**"销售年"**)  
 tree\_h.heading(**"销售月"**, text=**"销售月"**)  
 tree\_h.heading(**"销售日"**, text=**"销售日"**)  
 tree\_h.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:11]  
 tree\_h.insert(**""**, i, values=z[i])  
 **else**:  
 message.showerror(**'错误'**, **'无售出货物信息！'**)  
  
 **def** \_cxqb():  
 char2 = **"SELECT \* FROM sell"** find2 = excuate.cal\_r(char2)  
  
 char3 = **"""SELECT COUNT(\*) FROM  
 (SELECT \* FROM sell)a"""** num\_c = excuate.cal\_r(char3)[0][0]  
  
 tree\_h = ttk.Treeview(yjck\_win, show=**"headings"**, height=15)  
 tree\_h.place(relx=0.02, rely=0.11)  
  
 tree\_h[**"columns"**] = (**"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"**)  
 tree\_h.column(**"商品编号"**, width=47)  
 tree\_h.column(**"生产厂商"**, width=48)  
 tree\_h.column(**"商品名"**, width=48)  
 tree\_h.column(**"型号"**, width=47)  
 tree\_h.column(**"单价"**, width=48)  
 tree\_h.column(**"数量"**, width=47)  
 tree\_h.column(**"总金额"**, width=48)  
 tree\_h.column(**"销售年"**, width=48)  
 tree\_h.column(**"销售月"**, width=47)  
 tree\_h.column(**"销售日"**, width=47)  
 tree\_h.column(**"业务员编号"**, width=47)  
  
 tree\_h.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_h.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_h.heading(**"商品名"**, text=**"商品名"**)  
 tree\_h.heading(**"型号"**, text=**"型号"**)  
 tree\_h.heading(**"单价"**, text=**"单价"**)  
 tree\_h.heading(**"数量"**, text=**"数量"**)  
 tree\_h.heading(**"总金额"**, text=**"总金额"**)  
 tree\_h.heading(**"销售年"**, text=**"销售年"**)  
 tree\_h.heading(**"销售月"**, text=**"销售月"**)  
 tree\_h.heading(**"销售日"**, text=**"销售日"**)  
 tree\_h.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:11]  
 tree\_h.insert(**""**, i, values=z[i])  
  
 yjck\_win = Tk()  
 yjck\_win.geometry(**"550x375+550+320"**)  
 yjck\_win.title(**'查询销售货物'**)  
  
 Label(yjck\_win, text=**"商品名："**).place(relx=0.019, rely=0.015)  
 Label(yjck\_win, text=**"型号："**).place(relx=0.42, rely=0.015)  
  
 text\_spm = Entry(yjck\_win, width=14)  
 text\_spm.place(relx=0.15, rely=0.01)  
 text\_xh = Entry(yjck\_win, width=14)  
 text\_xh.place(relx=0.52, rely=0.01)  
  
 yghcx\_but = Button(yjck\_win, text=**"查询"**, width=6, command=\_tjcx)  
 yghcx\_but.place(relx=0.85, rely=0.018)  
  
 cxqb\_but = Button(yjck\_win, text=**"显示全部售出商品"**, width=58, command=\_cxqb)  
 cxqb\_but.place(relx=0.02, rely=0.92)  
  
 yjck\_win.mainloop()

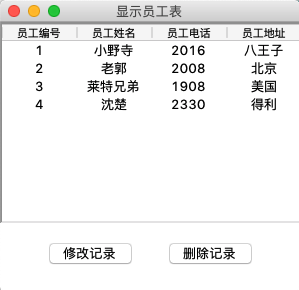
③退货表



退货表

**def** \_xsth():  
 xsth\_win = Tk()  
 xsth\_win.geometry(**"550x310+525+275"**)  
 xsth\_win.title(**'显示退货表'**)  
  
 tree\_h = ttk.Treeview(xsth\_win, show=**"headings"**, height=15)  
 tree\_h.place(relx=0.02, rely=0.01)  
  
 tree\_h[**"columns"**] = (**"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"**)  
 tree\_h.column(**"商品编号"**, width=47)  
 tree\_h.column(**"生产厂商"**, width=48)  
 tree\_h.column(**"商品名"**, width=48)  
 tree\_h.column(**"型号"**, width=47)  
 tree\_h.column(**"单价"**, width=48)  
 tree\_h.column(**"数量"**, width=47)  
 tree\_h.column(**"总金额"**, width=50)  
 tree\_h.column(**"退货年"**, width=48)  
 tree\_h.column(**"退货月"**, width=47)  
 tree\_h.column(**"退货日"**, width=47)  
 tree\_h.column(**"业务员编号"**, width=50)  
  
 tree\_h.heading(**"商品编号"**, text=**"商品编号"**)  
 tree\_h.heading(**"生产厂商"**, text=**"生产厂商"**)  
 tree\_h.heading(**"商品名"**, text=**"商品名"**)  
 tree\_h.heading(**"型号"**, text=**"型号"**)  
 tree\_h.heading(**"单价"**, text=**"单价"**)  
 tree\_h.heading(**"数量"**, text=**"数量"**)  
 tree\_h.heading(**"总金额"**, text=**"总金额"**)  
 tree\_h.heading(**"退货年"**, text=**"退货年"**)  
 tree\_h.heading(**"退货月"**, text=**"退货月"**)  
 tree\_h.heading(**"退货日"**, text=**"退货日"**)  
 tree\_h.heading(**"业务员编号"**, text=**"业务员编号"**)  
  
 char2 = **"SELECT \* FROM retreat"** find2 = excuate.cal\_r(char2)  
  
 char3 = **"""SELECT COUNT(\*) FROM  
 (SELECT \* FROM retreat)a"""** num\_c = excuate.cal\_r(char3)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_c):  
 z[i] = find2[i][0:11]  
 tree\_h.insert(**""**, i, values=z[i])

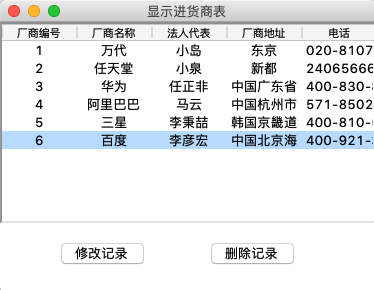
④员工表



员工表

**def** \_xsyg():  
 xsyg\_win = Tk()  
 xsyg\_win.title(**'显示员工表'**)  
 xsyg\_win.geometry(**"300x270+300+250"**)  
  
 tree\_yg = ttk.Treeview(xsyg\_win, height=10, show=**"headings"**)  
 tree\_yg.place(relx=0, rely=0)  
  
 tree\_yg[**"columns"**] = (**"员工编号"**, **"员工姓名"**, **"员工电话"**, **"员工地址"**)  
  
 tree\_yg.column(**"员工编号"**, width=75, anchor=**'center'**)  
 tree\_yg.column(**"员工姓名"**, width=75, anchor=**'center'**)  
 tree\_yg.column(**"员工电话"**, width=75, anchor=**'center'**)  
 tree\_yg.column(**"员工地址"**, width=75, anchor=**'center'**)  
  
 tree\_yg.heading(**"员工编号"**, text=**"员工编号"**)  
 tree\_yg.heading(**"员工姓名"**, text=**"员工姓名"**)  
 tree\_yg.heading(**"员工电话"**, text=**"员工电话"**)  
 tree\_yg.heading(**"员工地址"**, text=**"员工地址"**)  
  
 char1 = **"SELECT \* FROM employee"** find1 = excuate.cal\_r(char1)  
  
 char2 = **"SELECT count(\*) FROM (SELECT \* FROM employee)a"** num\_col = excuate.cal\_r(char2)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:4]  
 tree\_yg.insert(**""**, i, values=z[i])  
  
  
 **def** \_change():  
 **def** \_changed():  
 message.showinfo(**'提示'**, **'修改成功！'**)  
 changed\_but.destroy()  
  
 changed\_but = Button(xsyg\_win, text=**"确认修改"**, width=9, command=\_changed)  
 changed\_but.place(relx=0.16, rely=0.82)  
  
 **def** set\_cell\_value(event):  
 column = tree\_yg.identify\_column(event.x)  
 row = tree\_yg.identify\_row(event.y)  
 cn = int(str(column).replace(**'#'**, **''**))  
 rn = int(str(row).replace(**'I'**, **''**))  
 val = z[rn - 1][cn - 1]  
 c\_val = tree\_yg[**"columns"**][cn - 1]  
  
 text\_change = Entry(xsyg\_win, width=7)  
 text\_change.place(x=(cn-1)\*75, y=(rn-1)\*20+10)  
 text\_change.insert(0, val)  
  
 **def** \_set(event=**None**):  
 char = **"UPDATE employee SET %s = '%s' WHERE 员工编号 = '%s'"** % (  
 c\_val, text\_change.get(), rn  
 )  
 excuate.cal\_nr(char)  
  
 text\_change.destroy()  
  
 x = tree\_yg.get\_children()  
 **for** item **in** x:  
 tree\_yg.delete(item)  
 tree\_yg.selection\_clear()  
  
 char1 = **"SELECT \* FROM employee"** find1 = excuate.cal\_r(char1)  
  
 char2 = **"SELECT count(\*) FROM (SELECT \* FROM employee)a"** num\_col = excuate.cal\_r(char2)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:4]  
 tree\_yg.insert(**""**, i, values=z[i])  
  
 text\_change.bind(**'<Return>'**, \_set)  
  
 tree\_yg.bind(**'<Double-1>'**, set\_cell\_value)  
  
 **def** \_del():  
 char1 = **"SELECT \* FROM employee"** find1 = excuate.cal\_r(char1)  
  
 **global** rn  
 char = **"DELETE FROM employee WHERE 员工编号 = '%s'"** % find1[rn-1][0]  
 excuate.cal\_nr(char)  
  
 char2 = **"SELECT \* FROM employee"** find2 = excuate.cal\_r(char2)  
  
 char3 = **"SELECT count(\*) FROM (SELECT \* FROM employee)a"** num\_col = excuate.cal\_r(char3)[0][0]  
  
 x = tree\_yg.get\_children()  
 **for** item **in** x:  
 tree\_yg.delete(item)  
  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find2[i][0:4]  
 tree\_yg.insert(**""**, i, values=z[i])  
  
 **def** \_getrn(event):  
 row = tree\_yg.identify\_row(event.y)  
 **global** rn  
 rn = int(str(row).replace(**'I'**, **''**))  
 tree\_yg.bind(**'<Button-1>'**, \_getrn)  
  
 change\_but = Button(xsyg\_win, text=**"修改记录"**, width=9, command=\_change)  
 change\_but.place(relx=0.16, rely=0.82)  
 delete\_but = Button(xsyg\_win, text=**"删除记录"**, width=9, command=\_del)  
 delete\_but.place(relx=0.56, rely=0.82)

⑤进货商表



进货商表

**def** \_jhsb():  
 xsyg\_win = Tk()  
 xsyg\_win.title(**'显示进货商表'**)  
 xsyg\_win.geometry(**"375x270+300+250"**)  
  
 tree\_yg = ttk.Treeview(xsyg\_win, height=10, show=**"headings"**)  
 tree\_yg.place(relx=0, rely=0)  
  
 tree\_yg[**"columns"**] = (**"厂商编号"**, **"厂商名称"**, **"法人代表"**, **"厂商地址"**, **"电话"**)  
  
 tree\_yg.column(**"厂商编号"**, width=75, anchor=**'center'**)  
 tree\_yg.column(**"厂商名称"**, width=75, anchor=**'center'**)  
 tree\_yg.column(**"法人代表"**, width=75, anchor=**'center'**)  
 tree\_yg.column(**"厂商地址"**, width=75, anchor=**'center'**)  
 tree\_yg.column(**"电话"**, width=75, anchor=**'center'**)  
  
 tree\_yg.heading(**"厂商编号"**, text=**"厂商编号"**)  
 tree\_yg.heading(**"厂商名称"**, text=**"厂商名称"**)  
 tree\_yg.heading(**"法人代表"**, text=**"法人代表"**)  
 tree\_yg.heading(**"厂商地址"**, text=**"厂商地址"**)  
 tree\_yg.heading(**"电话"**, text=**"电话"**)  
  
 char1 = **"SELECT \* FROM manufacturer"** find1 = excuate.cal\_r(char1)  
  
 char2 = **"SELECT count(\*) FROM (SELECT \* FROM manufacturer)a"** num\_col = excuate.cal\_r(char2)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:5]  
 tree\_yg.insert(**""**, i, values=z[i])  
  
 **def** \_change():  
 **def** \_changed():  
 message.showinfo(**'提示'**, **'修改成功！'**)  
 changed\_but.destroy()  
  
 changed\_but = Button(xsyg\_win, text=**"确认修改"**, width=9, command=\_changed)  
 changed\_but.place(relx=0.16, rely=0.82)  
  
 **def** set\_cell\_value(event):  
 column = tree\_yg.identify\_column(event.x)  
 row = tree\_yg.identify\_row(event.y)  
 cn = int(str(column).replace(**'#'**, **''**))  
 rn = int(str(row).replace(**'I'**, **''**))  
 val = z[rn - 1][cn - 1]  
 c\_val = tree\_yg[**"columns"**][cn - 1]  
  
 text\_change = Entry(xsyg\_win, width=7)  
 text\_change.place(x=(cn - 1) \* 75, y=(rn - 1) \* 20 + 10)  
 text\_change.insert(0, val)  
  
 **def** \_set(event=**None**):  
 char = **"UPDATE manufacturer SET %s = '%s' WHERE 厂商编号 = '%s'"** % (  
 c\_val, text\_change.get(), rn  
 )  
 excuate.cal\_nr(char)  
  
 text\_change.destroy()  
  
 x = tree\_yg.get\_children()  
 **for** item **in** x:  
 tree\_yg.delete(item)  
 tree\_yg.selection\_clear()  
  
 char1 = **"SELECT \* FROM manufacturer"** find1 = excuate.cal\_r(char1)  
  
 char2 = **"SELECT count(\*) FROM (SELECT \* FROM manufacturer)a"** num\_col = excuate.cal\_r(char2)[0][0]  
  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find1[i][0:4]  
 tree\_yg.insert(**""**, i, values=z[i])  
  
 text\_change.bind(**'<Return>'**, \_set)  
  
 tree\_yg.bind(**'<Double-1>'**, set\_cell\_value)  
  
 **def** \_del():  
 char1 = **"SELECT \* FROM manufacturer"** find1 = excuate.cal\_r(char1)  
  
 **global** rn  
 char = **"DELETE FROM manufacturer WHERE 厂商编号 = '%s'"** % find1[rn - 1][0]  
 excuate.cal\_nr(char)  
  
 char2 = **"SELECT \* FROM manufacturer"** find2 = excuate.cal\_r(char2)  
  
 char3 = **"SELECT count(\*) FROM (SELECT \* FROM manufacturer)a"** num\_col = excuate.cal\_r(char3)[0][0]  
  
 x = tree\_yg.get\_children()  
 **for** item **in** x:  
 tree\_yg.delete(item)  
  
 z = {}  
 **for** i **in** range(0, num\_col):  
 z[i] = find2[i][0:4]  
 tree\_yg.insert(**""**, i, values=z[i])  
  
 **def** \_getrn(event):  
 row = tree\_yg.identify\_row(event.y)  
 **global** rn  
 rn = int(str(row).replace(**'I'**, **''**))  
  
 tree\_yg.bind(**'<Button-1>'**, \_getrn)  
  
 change\_but = Button(xsyg\_win, text=**"修改记录"**, width=9, command=\_change)  
 change\_but.place(relx=0.16, rely=0.82)  
 delete\_but = Button(xsyg\_win, text=**"删除记录"**, width=9, command=\_del)  
 delete\_but.place(relx=0.56, rely=0.82)

# 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

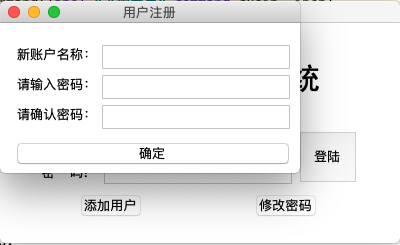
1. 系统登陆

在相应的框内输入用户名和密码，点击登录即可实现登陆



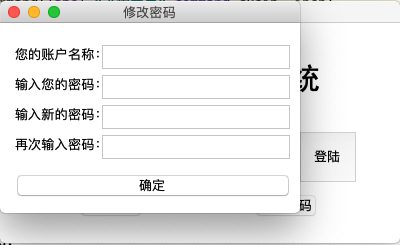
1. 用户注册

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-今日进货量即可弹出进货信息界面。

②本月进货量：在菜单上选择进货统计-本月进货量即可弹出进货信息界面。

③本季度进货量：在菜单上选择进货统计-本季度进货量即可弹出进货信息界面。

④今年进货量：在菜单上选择进货统计-今年进货量即可弹出进货信息界面。

1. 销售统计

①今日销售量：在菜单上选择销售统计-今日销售量即可弹出销售信息界面。

②本月销售量：在菜单上选择销售统计-本月销售量即可弹出销售信息界面。

③本季度销售量：在菜单上选择销售统计-本季度销售量即可弹出销售信息界面。

④今年销售量：在菜单上选择销售统计-今年销售量即可弹出销售信息界面。

1. 业绩查看

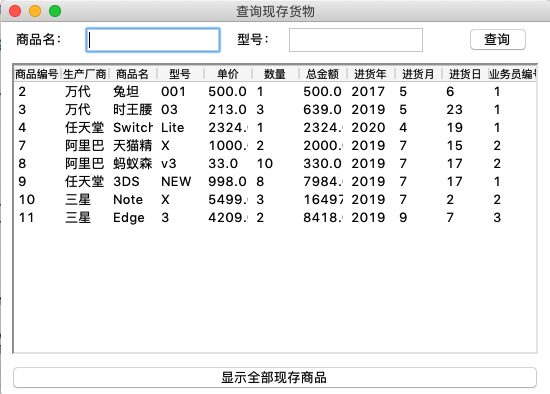
在菜单上选择业绩查看-业绩查看即可弹出业绩查看界面。



通过输入员工号查询特定员工的销售情况，也可点击底部的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。



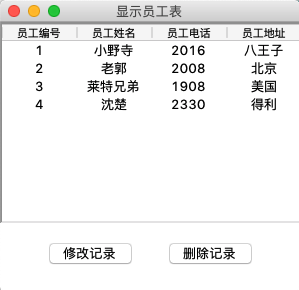
通过输入商品名和型号查询商品的销售情况，也可点击底部的显示全部现存商品来显示全部现存商品。

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，双击想要修改的数据，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

净时间约50小时，花了大约三天半写完的这份程序对我个人的意义非同凡响，它既是我独立一人写完的第一次超过两千行的代码文件，也是一次必不可少的思想锤炼。这也大抵上是体验了一把程序员的生活，每天定点起床，每天都去一样的地方把自己关在代码里，在实践的过程中无数次的在迷宫里寻找出路，循环着阴暗与光亮的日子是一份宝藏，打开这份宝藏的钥匙本就在我心里，只是需要这场试炼去让我得到它。