****

**数据库课程设计报告**

**-------苗木销售管理系统**

**计算机科学与技术 学院**

**计算机科学与技术 专业**

**2016 级 4 班**

**姓名： 赵光昊**

**学号： 201600130055**

**任课教师： 实验教师： 助教：**

前言 3

一、系统开发平台 4

二、数据库规划 5

2.1 任务陈述 5

2.2 任务目标 5

三、系统定义 7

3.1 系统边界 7

3.2 用户视图 8

四、需求分析 9

4.1 用户需求说明 9

4.1.1 数据需求 9

4.1.2 事务需求 9

4.2 系统需求说明 10

五、数据库逻辑设计 12

5.1 ER图【3】： 12

5.2 数据字典 12

六、数据库物理设计 14

6.1 索引 14

6.2 视图 14

6.3 安全机制 15

6.4 其他 15

七、应用程序设计 16

7.1 功能模块 17

7.2 界面设计 18

7.3 事务设计 28

八、测试和运行 43

九、总结 54

9.1 系统优点 54

9.2 系统不足 54

9.3 系统改进 54

9.4 经验与收获 55

前言

进入二十一世纪以来，随着计算机技术的发展，计算机行业已经不断融入其他行业并提供服务，计算机在信息管理方面的优势尽显。与此同时，随着我国提出可持续发展战略和保护环境国策，越来越多的人开始关注环保问题，近年更是明确指出将环保纳入政绩考核之中，环保有关的行业高速发展，其中就有苗木行业。苗木行业有单体价格高，单笔成交金额高，成交次数有限的特点，在过去对计算机技术的需求不高，但是，随着近年的发展，苗木行业的需求越来越大，销售相关的数据暴涨，传统的信息记录方式已经无法满足要求，所以，为公司开发苗木库存销售管理系统刻不容缓。

本项目就是研究苗木库存销售管理系统的数据库构件过程。该数据库展示了苗木销售过程中的基本信息与基本流程，主要包括苗木的基本信息资料与买家的基本资料，记录苗木入库、销售出库、买家记录、收益记录。

# 一、系统开发平台

**1.1 开发平台简介**

本苗木销售管理系统采用C-S架构，使用Java的技术路线，遵守MVC Model模型。

Model模型层使用数据库作为持久性储存结构，选用的后台数据库是MySQL5.7。这是业界领先的开源数据库，在开源产品中具有仅次于Apache服务器的市场占有率。本数据库开放源代码，具有免费使用，比较稳定的特点，适合于小型系统的持久性存储。

View视图层使用JAVA GUI技术。这是Java 本身的图形界面技术。住院管理系统中严格控制界面代码中的业务逻辑部分，实现了很好的业务和表现的分离。

Controller控制层实现了对数据库中数据的增删查改的具体方法。界面获取的数据通过调用相应方法进行数据的封装和与数据库的交互，从而实现业务逻辑和表现的完全分离。

**1.2 开发语言：Java**

用Java做设计流程清晰、结构合理，有良好的可扩充性和耦合性。

**1.3 开发工具：Eclipse、 JDK、 Navicat for MySQL**

**1.4 数据库: MySQL 5.7**

MySQL是业界领先的开源数据库，在开源产品中具有仅次于Apache服务器的市场占有率。本数据库开放源代码，具有免费使用，比较稳定的特点，适合于小型系统的持久性存储。

**1.5 操作系统:Microsoft Windows 10**

# 二、数据库规划

## 2.1 任务陈述

苗木销售管理系统是一个使整个销售信息化的关键。随着环保意识的普及，苗木销售数量急剧增加，有关销售的各种信息量也成倍增长。【1】面对庞大的信息量需要有苗木管理系统来提高销售记录管理工作的效率。一个完善的苗木销售管理系统能够极大地提高苗木信息管理的效率,具有检索迅速、查找方便、可靠性高、存储量大、更新快、寿命长、成本低等优点。【2】

本苗木销售管理系统包括库存明细管理、销售明细管理、收益明细管理、买家信息管理四大模块并且及时记录存储各个环节信息的变更，以便管理、查询、显示、输出。本系统因为考虑到本身系统的特殊性，所以本系统只有管理员模式一种。管理员可以拥有完全的权限管理系统，可以完成对库存信息、销售信息、收益信息、买家信息的查询，并且完成苗木入库、苗木销售的操作。

## 2.2 任务目标

|  |  |  |
| --- | --- | --- |
| 模板 | 功能名称 | 具体描述 |
| 库存明细管理 | 记录苗木库存基本信息 | 编号（苗木编号）、日期、品种、规格（苗木分支数）、数量、树木-土球状态（根部带土情况）、来源（苗木原产地）、场圃（种在哪个场圃）、装车价（进价）、运费、卸车+种植、成本价（装车价+运费+卸车+种植）、预售价、实景图（现场苗木图片）、备注 |
|  | 能够输入、修改、查询（由品种查询） | 能够查询某一品种树苗的库存信息 |
| 销售明细管理 | 记录苗木销售信息 | 日期（重新填写的发货日期）、品种、规格（苗木分支数）、数量、编号、来源、销售价、购买方、配车信息（地点、车型、运费）、备注 |
|  | 能够输入、修改、查询 | 查询某一条或几条销售记录 |
| 收益明细管理 | 记录收益信息 | 记录编号、日期、成本价、销售价 |
|  | 能够输入、修改、查询 | 查询某一时间（月度）内的收益信息 |
| 买家信息管理 | 记录买家信息 | 买家编号、姓名、公司、电话 |
|  | 能够输入、修改、查询 | 能都查询买家信息 |

# 三、系统定义

## 3.1 系统边界

系统边界描述数据库系统和企业信息系统的其他部分的接口，是信息系统内部构成元素与外部有联系实体之间的信息关系的描述与分割。它并不需要在它们之间划一条物理边界，而只需要弄清它们之间信息输入与输出的分割。

本数据库系统共包括苗木信息、销售信息、收益信息、买家信息四个模块。该信息系统的其他部分包括苗木入库、苗木销售、买家信息录入、月度收益信息等。

数据库系统的内部构成元素与外部其他部分之间的信息关系如下图所示：

苗木信息

收益信息

苗木入库

苗木销售

销售信息

买家信息

买家信息录入

月度收益信息

## 3.2 用户视图

**管理员（Administrator）用户视图**

1. 库存明细管理：

输入、修改、查询。

1. 销售明细管理：

输入、修改、查询。

1. 收益明细管理：

输入、修改、查询。

1. 买家信息管理：

输入、修改、查询。

# 四、需求分析

## 4.1 用户需求说明

### 4.1.1 数据需求

其中需求数据为：

1. 苗木入库基本信息记录：

编号（苗木编号）、日期、品种、规格、数量、树木-土球状态（根部带土情况）、来源、场圃（种在哪个场圃）、装车价（进价）、运费、卸车+种植、成本价（装车价+运费+卸车+种植）、预售价、实景图（现场苗木图片）、备注

1. 苗木销售基本信息记录：

日期（重新填写的发货日期）、品种、规格、数量、编号、来源、销售价、购买方、配车信息（地点、车型、运费）、备注

1. 买家基本信息记录：

姓名、公司、电话

### 4.1.2 事务需求

1. **数据录入：**
2. 录入苗木的基本信息：

日期、品种、规格、数量、树木-土球状态（根部带土情况）、编号（苗木编号）、来源、场圃（种在哪个场圃）、装车价（进价）、运费、卸车+种植、成本价（装车价+运费+卸车+种植）、预售价、实景图（现场苗木图片）、备注

1. 录入苗木的销售信息：

日期（重新填写的发货日期）、品种、规格、数量、编号、来源、销售价、购买方、配车信息（地点、车型、运费）、备注

1. 录入买家信息：

姓名、公司、电话

1. **数据更新/删除：**

（1） 苗木信息的更新

（2） 销售信息的更新/ 删除

1. **数据查看：**

**管理员用户视图下：**

（1） 按照苗木品种列出库存苗木记录：编号、日期、品种、规格、数量、树木-土球状态、来源、场圃、装车价、运费、卸车+种植、成本价、预售价、实景图、备注

（2） 按照买家列出销售记录：日期、品种、规格、数量、编号、来源、销售价、购买方、配车信息、备注

（3） 按照月份列出月收益状况

## 4.2 系统需求说明

该苗木销售管理系统需要较强的数据处理功能，理论上应该能够容纳上万苗木的数据资料，并且在搜索方面理应具有较快的响应速度，能够处理多方面的数据请求。系统能够有效的处理各种异常，具有较好的健壮性。

**4.2.1 初始数据库大小**

（1） 库存苗木大约有百棵苗木。

（2） 销售记录留存百条。

**4.2.2 网络和共享需求**

必须能够支持唯一管理员用户访问，不需要考虑这么大数量并发访问的许可需求。

**4.2.3 性能**

（1） 单个记录查询时间少于1秒

（2） 更新/保存记录时间少于1秒

**4.2.4 安全性**

（1） 数据库必须有口令保护

（2） 用户只能在适合他们完成工作需要的窗口中看到需要的数据

**4.2.5 备份和恢复**

每天24点备份

**4.2.6 用户界面**

菜单驱动

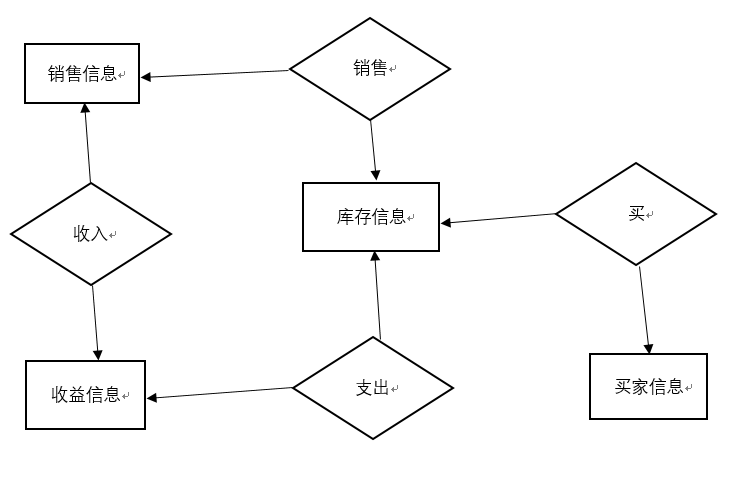
**4.2.7 法律问题**

对用户信息管理，遵守法律

五、数据库逻辑设计

## 5.1 ER图【3】：

本系统的ER图如下：



## 5.2 数据字典

**5.2.1 从数据字典中抽取出来的系统实体描述：**

库存明细管理实体表：



销售明细管理实体表：



收益明细管理实体表：



买家信息管理实体表：



**5.2.2 从数据字典中抽取出来的联系的描述：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实体 | 多样性 | 联系 | 多样性 | 实体 |
| 库存信息 | 1 | 支出 | 1 | 收益信息 |
| 库存信息 | M | 买 | 1 | 买家信息 |
| 库存信息 | 1 | 销售 | 1 | 销售信息 |
| 收益信息 | 1 | 收入 | 1 | 销售信息 |

# 六、数据库物理设计

## 6.1 索引

数据库中的索引：

|  |  |  |
| --- | --- | --- |
| 表名 | 主键 | 外键 |
| 库存信息 | 苗木ID |  |
| 销售信息 | 苗木ID | 买家ID |
| 收益信息 | 交易ID |  |
| 买家信息 | 买家ID |  |

## 6.2 视图

此环节设计在数据库应用生命周期的需求分析和收集阶段标识的用户视图。通常，视图使用SQL或类似QBE的工具创建。例如查询id为1苗木的全部信息：

CREATE VIEW Tree\_View

AS SELECT

treeID, date, tree, size,num, tree\_soil, source, field, on\_trucks, transport, off\_trucks, cost\_price, sale\_price, img, notes

FROM tree

WHERE treenum = 1；

## 6.3 安全机制

**6.3.1 系统安全**

1、本苗木销售管理系统提供了充足的异常处理机制，能够捕获由各种错误引发的异常(如：输入数据类型与数据库要求类型不一致、查询过程中出现的错误、操作不规范等等)。

2、管理员输入信息如果与数据库已有信息重复，则提示管理者重新输入

**6.3.2 数据安全**

1、在登陆界面输入管理ID与密码。只有管理员才进入本系统实现苗木销售及资金有关的一系列流程，以确保数据库不被随意更改，保证数据安全。

2、本系统只有管理员身份，在完成苗木入库、苗木销售、收支项目记录、买家信息记录等操作后需要更新其他的有关的联系表，这些都是通过事务实现的，很好的保护了关系型数据库的强一致性，体现了数据安全。

## 6.4 其他

规范化产生一个结构上一致且最小冗余的逻辑数据库设计，但是，规范化的数据库设计有时不能提供最大的处理效率。所以我们愿意接收规范化设计方面的一些损失而实现更好的性能。

# 七、应用程序设计

此苗木管理系统的主要功能如下图所示：

**苗木管理系统**

**登录系统**

**苗木入库登记**

苗木信息查询

苗木信息登记

**买家登记**

买家信息查询

买家信息登记

**收益记录**

指定时段内收支信息查询

指定时段内收益计算

**退出系统**

**销售记录**

查找预售苗木

查找买家

销售苗木操作

销售信息查询

菜单

## 7.1 功能模块

该住院管理信息系统共有苗木入库登记、买家登记、销售记录、收益记录四大功能模块，功能模块具体实现如下：

**7.1.1 苗木登记**

（1）填写苗木基本信息

（2）完成苗木登记以及相关表的更新

* + 1. **苗木查询**

（1）查询苗木信息

（2）输入苗木品种

* + 1. **买家登记**

（1）填写买家基本信息

（2）完成买家登记以及相关表的更新

* + 1. **买家查询**

（1）查询买家信息

（2）输入买家姓名

* + 1. **苗木销售**

（1）查询库存苗木信息

（2）查询买家信息

（3）输入预售数量

（4）检测库存是否充足

（5）完成销售操作，更新相关表

* + 1. **苗木销售记录查询**

（1）查询销售苗木信息

（2）输入查询起止时间

* + 1. **收益查询求和**

（1）查询收支信息

（2）计算收支总计

## 7.2 界面设计

**7.2.1 登录界面**

用户在登录界面输入用户名和密码并经过系统验证后，可以进入主服务界面进行相关操作（账号密码为初始设定好的值）



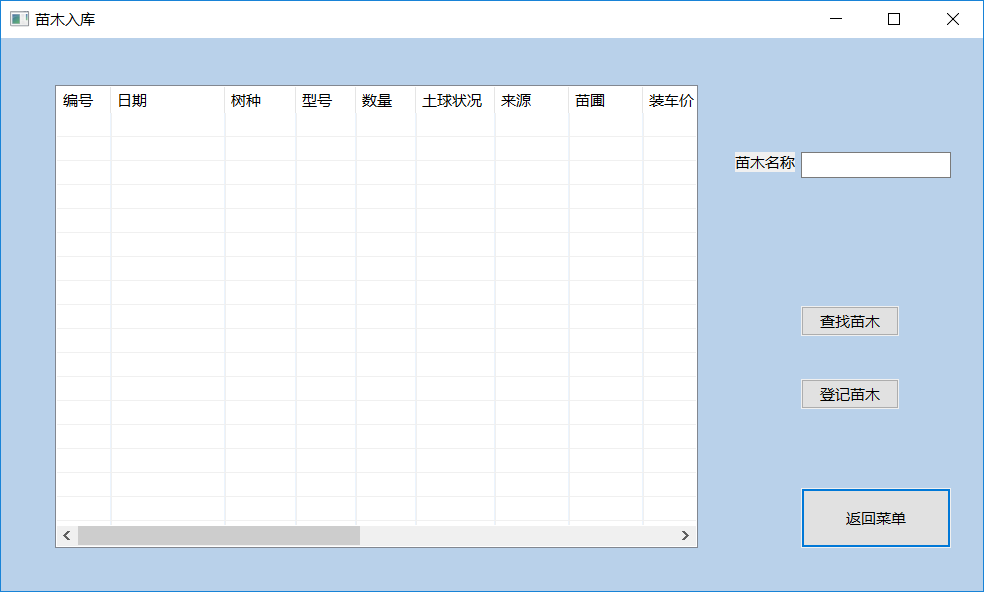
**7.2.2 主服务界面**

在输入正确的账号密码后进入主服务界面，主服务界面提供了通往目的功能的按钮，在按下按钮后主服务界面dispose，显示新的界面。

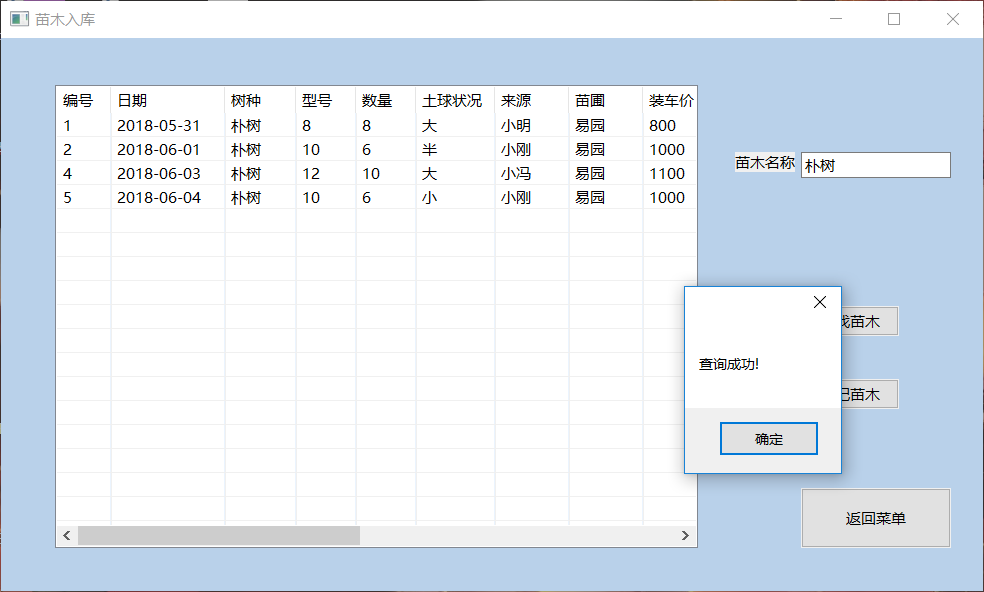


**7.2.3 苗木信息查询**

1. 苗木信息查询界面如下图所示：



1. 查询结果

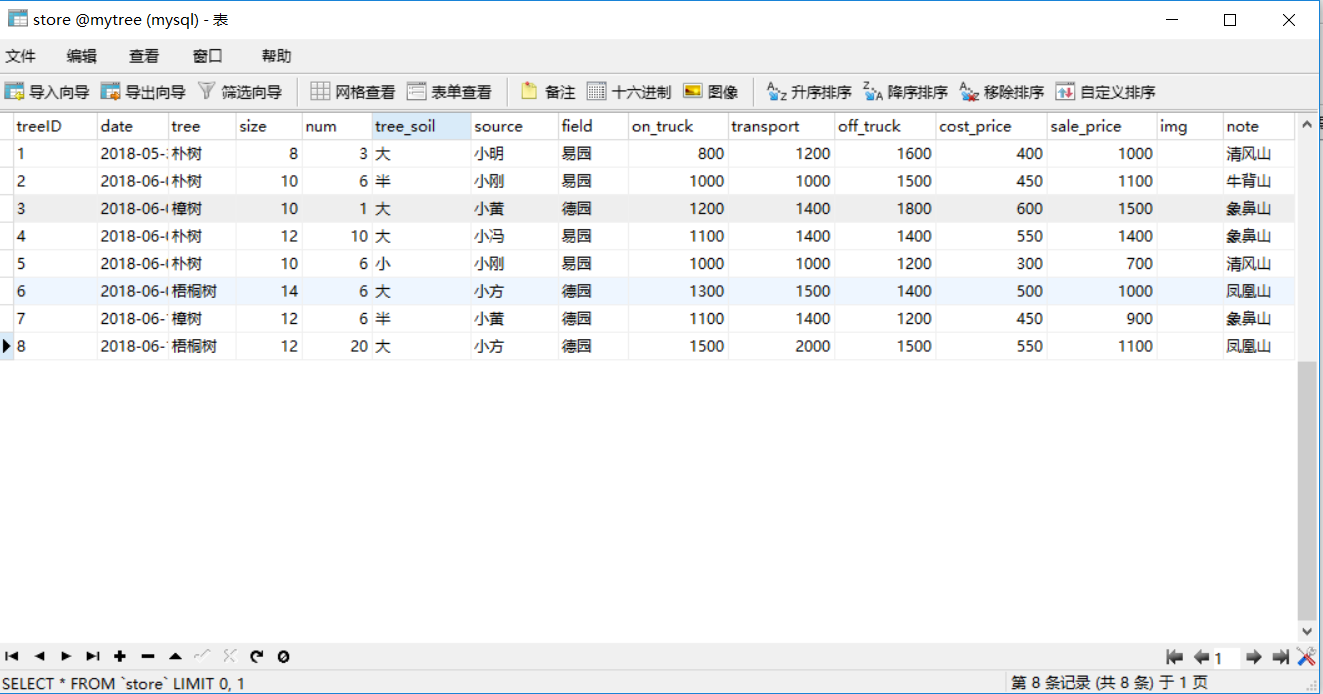


1. 苗木入库登记

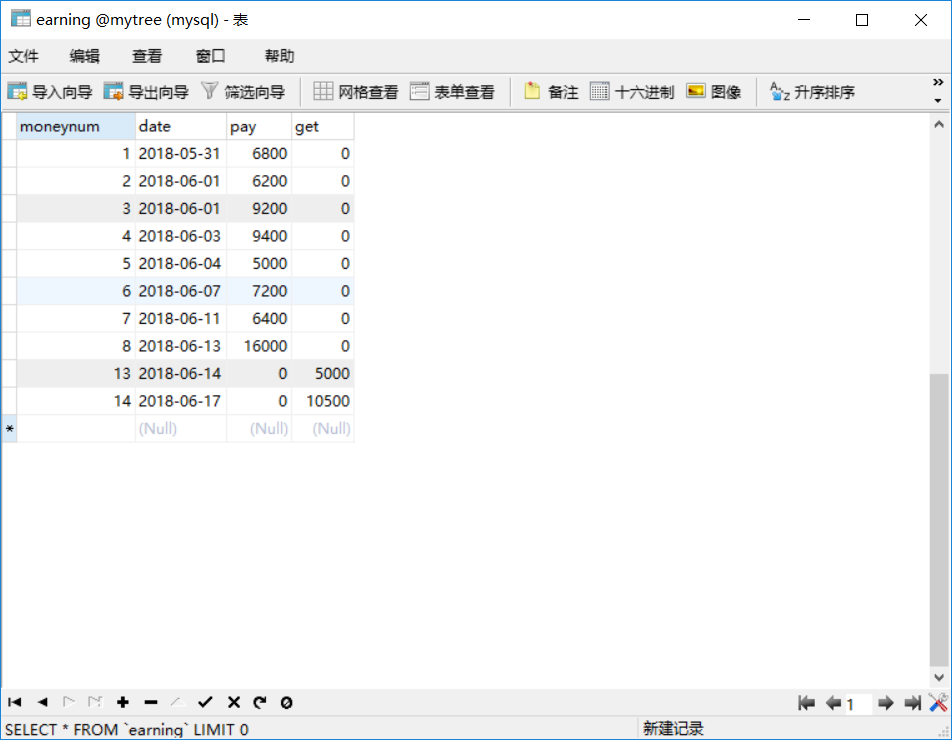


1. 完成登记后更新相关表

苗木库存信息表（编号8所在行）

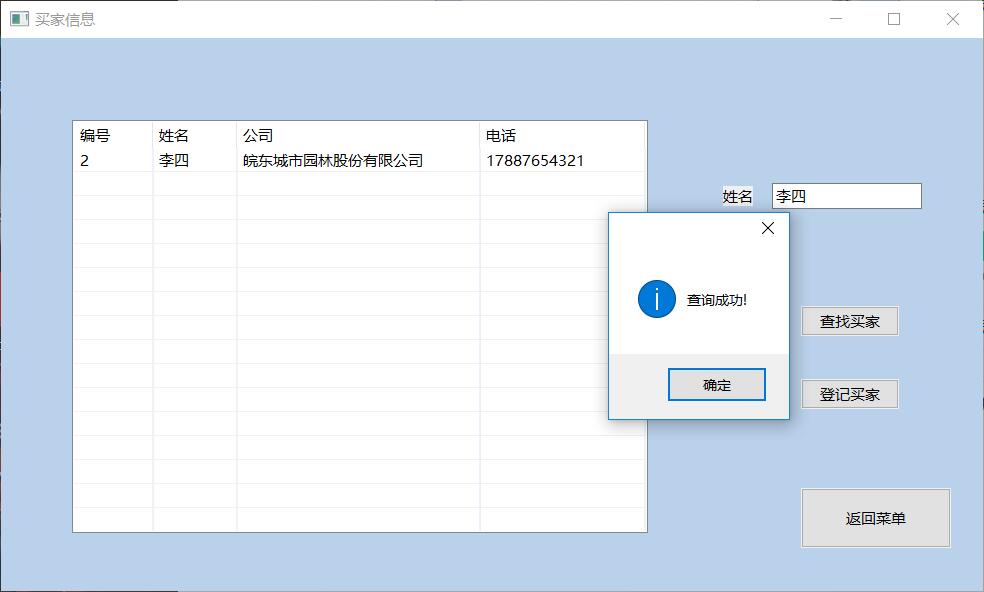


收支信息表（编号8所在行）

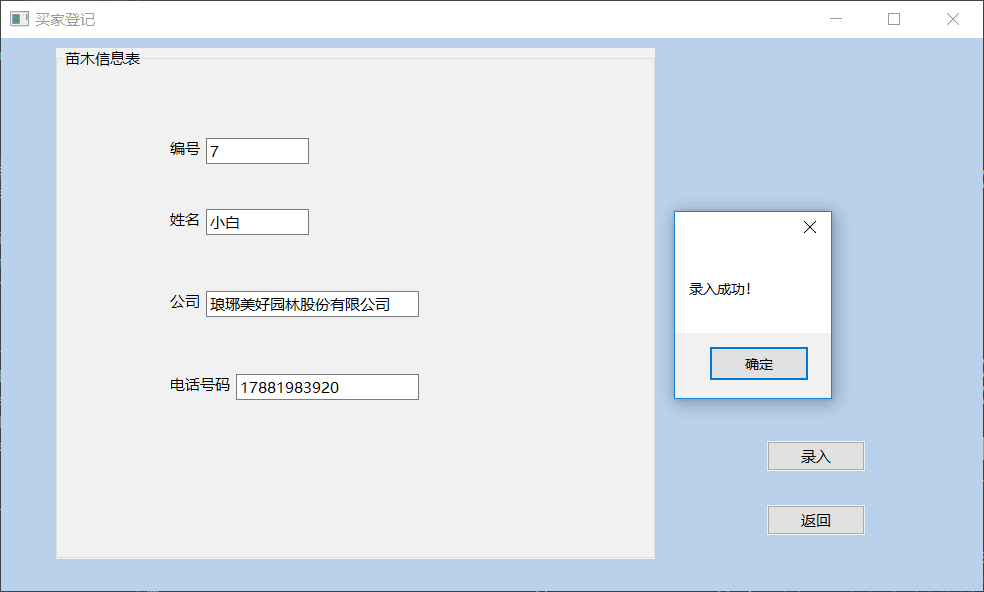


**7.2.4 买家信息查询**

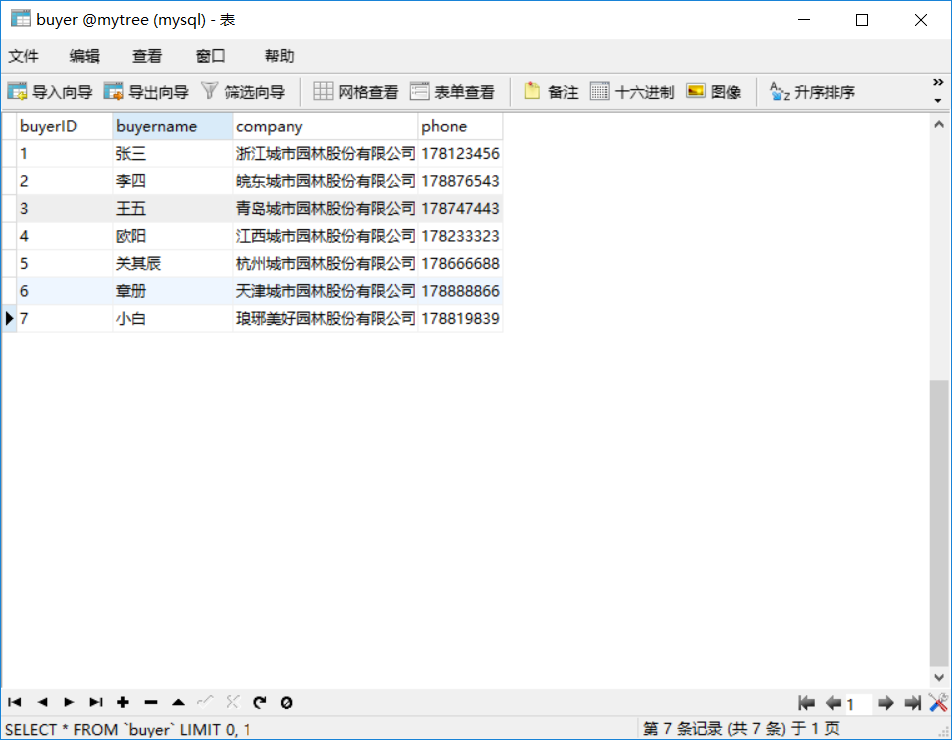
1. 买家信息查询界面如下图所示：



1. 买家信息登记界面如下图所示：

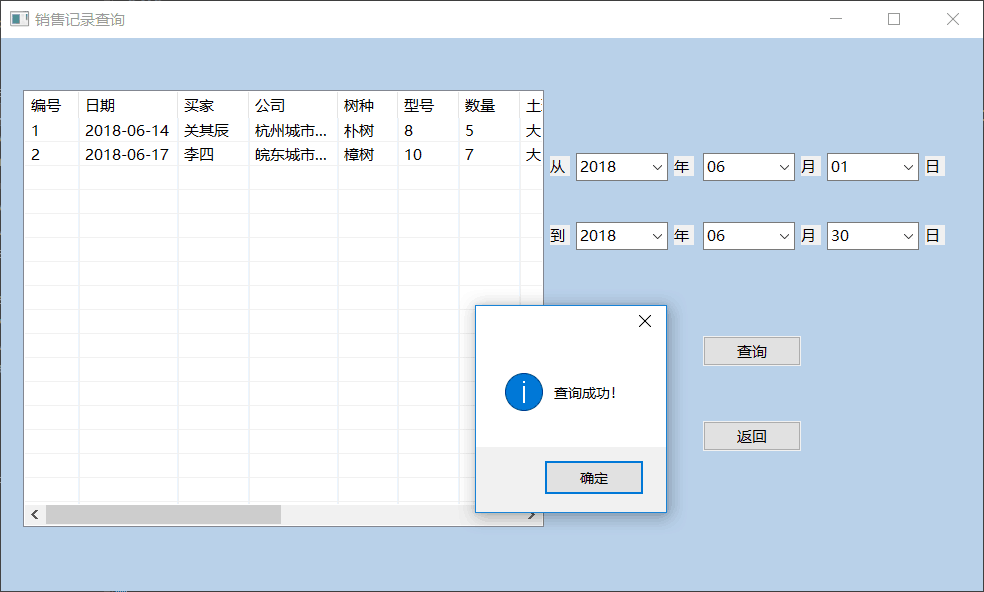
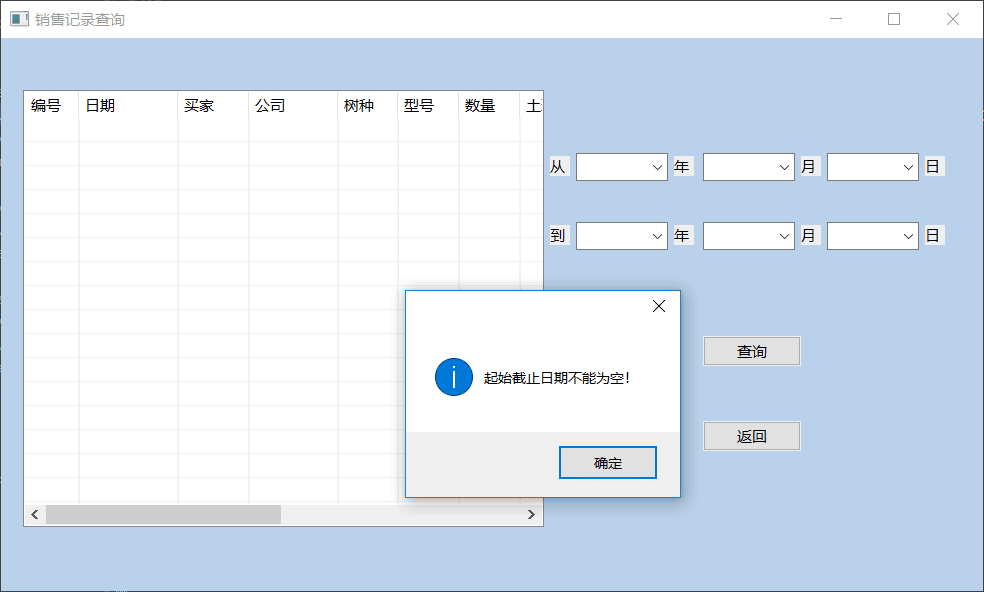


1. 完成登记后更新相关表如下图所示：

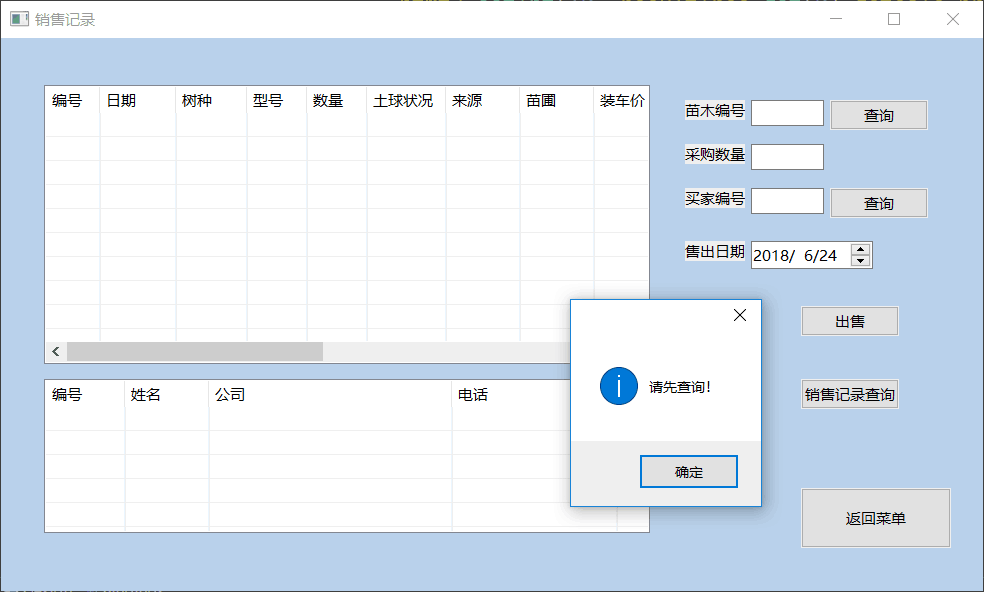


**7.2.5 销售信息查询**

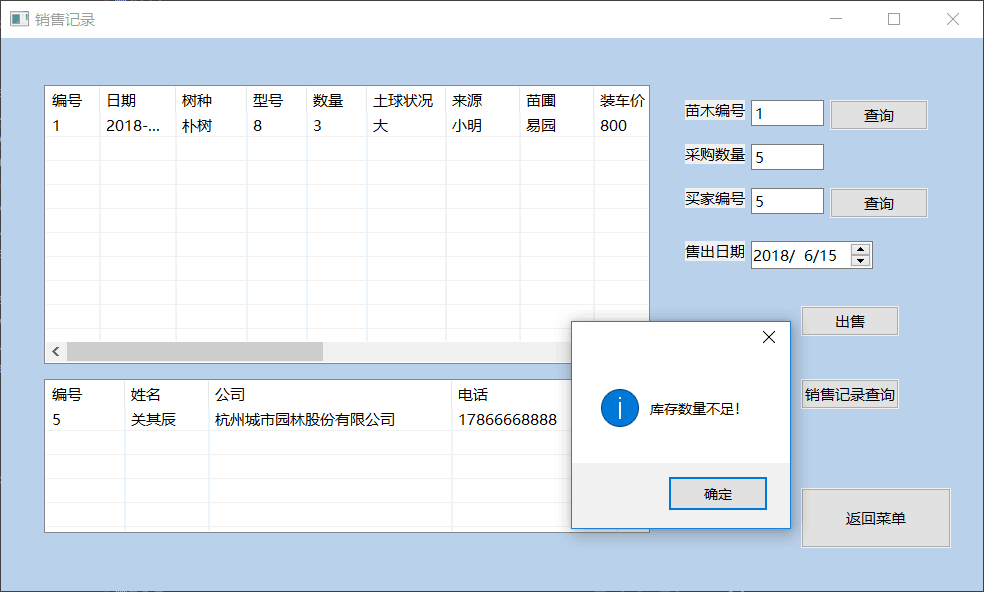
（1）销售信息查询界面如下图所示：



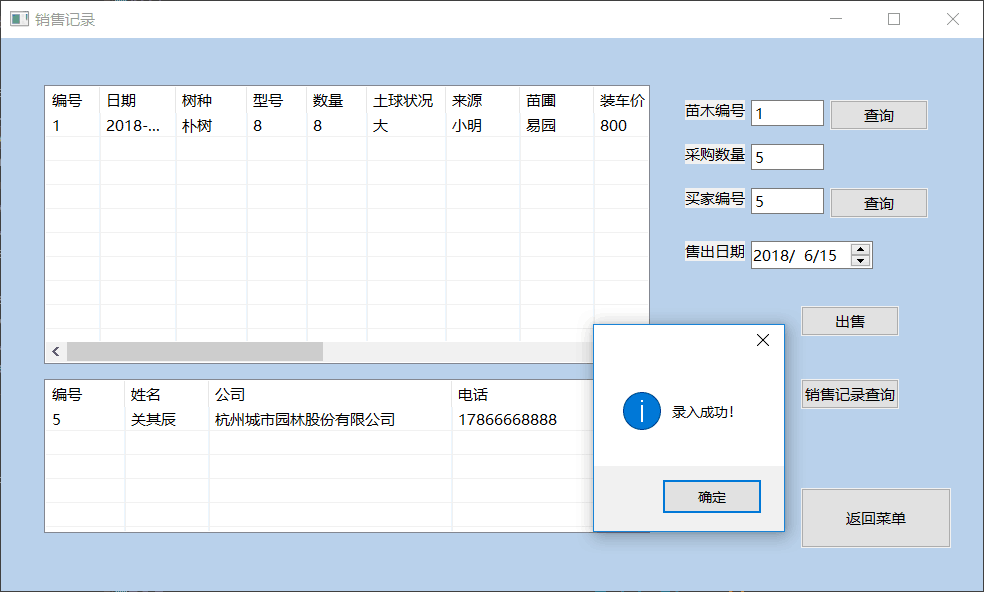
（2）销售过程界面如下图所示：



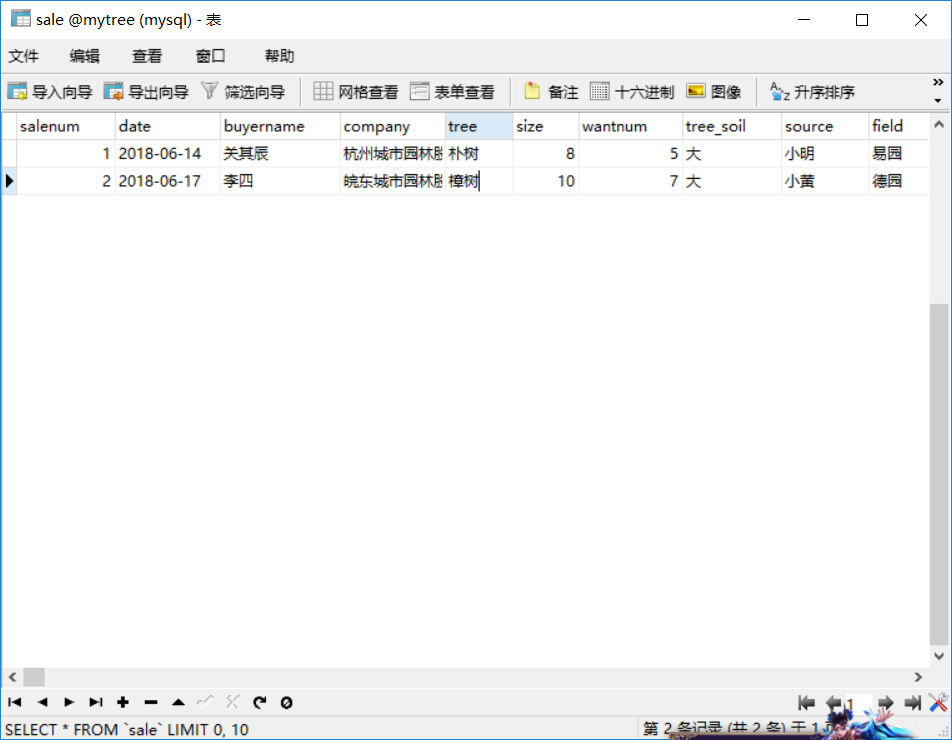
（3）销售库存数量判断界面如下图所示：

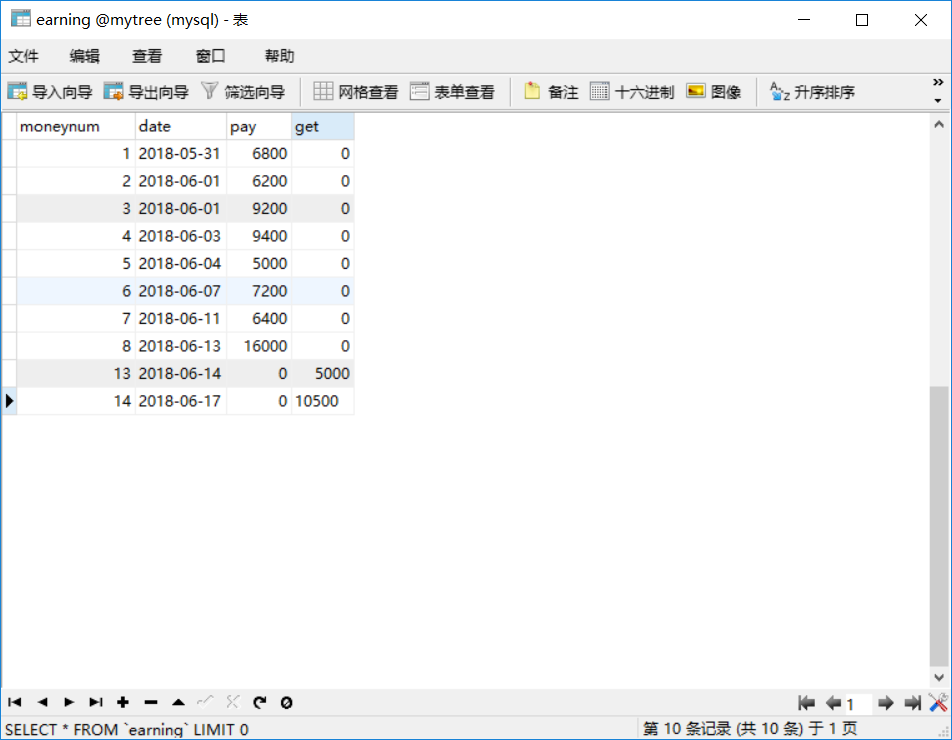


（4）销售成功结果界面如下图所示：



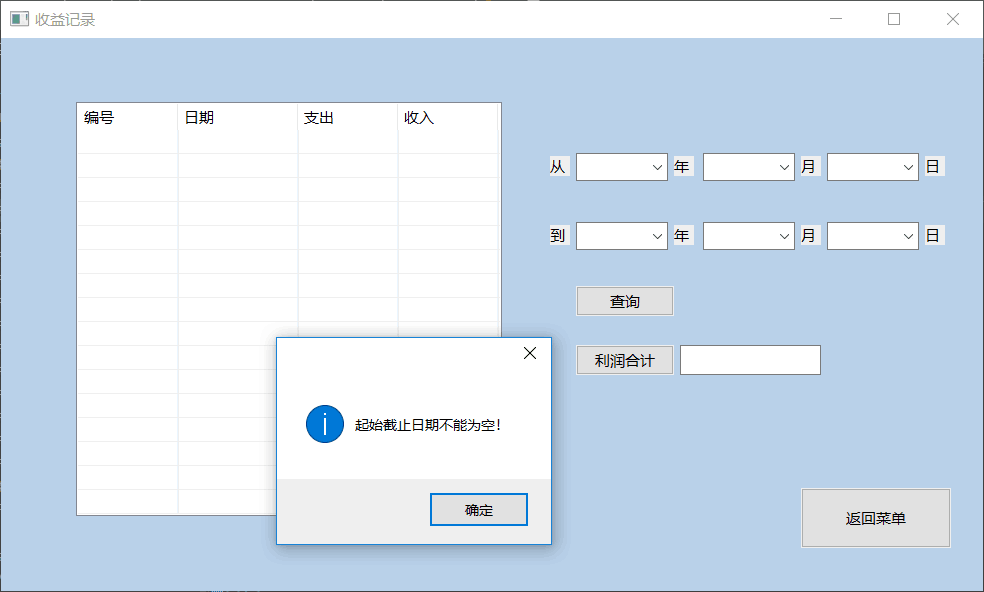
（5）完成登记后更新相关表如下图所示：



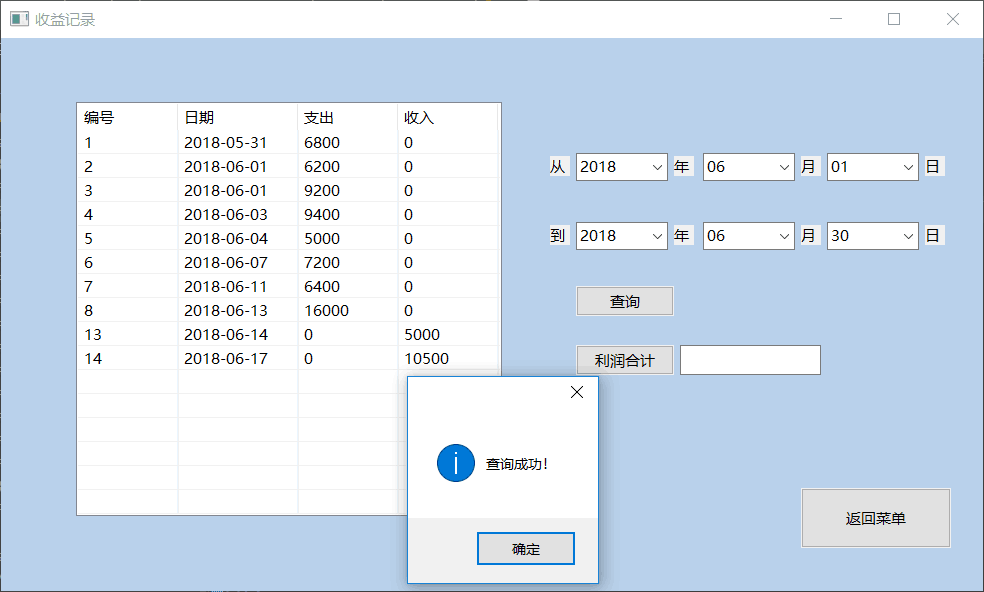


**7.2.6 收益信息查询**

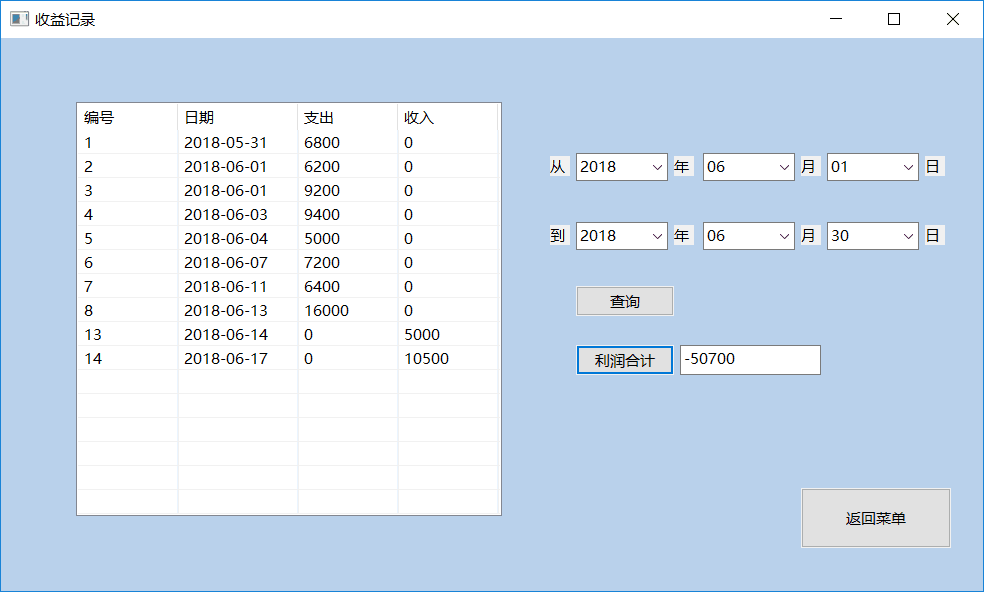
（1）收益信息查询界面



（2）收益信息查询结果：



（3）收益信息金额求和



## 7.3 事务设计

该系统中，所有关于查询的事务基本上都可以由Select语句实现，而更新的事务则用Update语句，删除的事务用Delete语句实现。

**7.3.1苗木入库事务**

入库登记：

Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/mytree?useSSL=false&serverTimezone=UTC","root","001215");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

int iday=dateTime.getDay();

int imonth=dateTime.getMonth()+1;

int iyear=dateTime.getYear();

String day,month,year;

if(iday<10) {

day = "0"+iday;

}else{

day = ""+iday;

}

if(imonth<10) {

month = "0"+imonth;

}else{

month = ""+imonth;

}

year = ""+iyear;

DateFormat df = new SimpleDateFormat("yyyy-MM-dd");

java.util.Date u\_date=new java.util.Date();

u\_date = df.parse(year+"-"+month+"-"+day);

java.sql.Date s\_date=new java.sql.Date(u\_date.getTime());

String treeID=text\_1.getText();//苗木编号

String tree=text\_2.getText();//苗木名称

int size=Integer.parseInt(text\_3.getText());

int num=Integer.parseInt(text\_4.getText());

String tree\_soil=text\_5.getText();

String source=text\_6.getText();

String field=text\_7.getText();

int on\_truck=Integer.parseInt(text\_8.getText());

int transport=Integer.parseInt(text\_9.getText());

int off\_truck=Integer.parseInt(text\_10.getText());

int cost\_price=Integer.parseInt(text\_11.getText());

int sale\_price=Integer.parseInt(text\_12.getText());

String img=text\_13.getText();

String note=text\_14.getText();

String sql ="" +

"insert into store " +

"(treeID,date,tree,size,num,tree\_soil,source,field," +

"on\_truck,transport,off\_truck,cost\_price,sale\_price,img,note) " +

"values(" +

"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

java.sql.PreparedStatement ptmt = connect.prepareStatement(sql);

ptmt.setString(1, treeID);

ptmt.setDate(2, s\_date);

ptmt.setString(3, tree);

ptmt.setInt(4, size);

ptmt.setInt(5, num);

ptmt.setString(6, tree\_soil);

ptmt.setString(7, source);

ptmt.setString(8, field);

ptmt.setInt(9, on\_truck);

ptmt.setInt(10, transport);

ptmt.setInt(11, off\_truck);

ptmt.setInt(12, cost\_price);

ptmt.setInt(13, sale\_price);

ptmt.setString(14, img);

ptmt.setString(15, note);

ptmt.execute();

String sql1 ="" +

"insert into earning " +

"(date,pay,`get`) " +

"values(" +

"?,?,0)";

java.sql.PreparedStatement ptmt1 = connect.prepareStatement(sql1);

ptmt1.setDate(1, s\_date);

ptmt1.setInt(2, cost\_price\*num+on\_truck+transport+off\_truck);

ptmt1.execute();

苗木查询：

Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/mytree?useSSL=false&serverTimezone=UTC","root","001215");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，MySQL8.0要添加useSSL和UTC，后面的2个参数分别是登陆用户名和密码

String fname=text.getText();//苗木名称

String sql = "select \* from store "+

"where tree = ?";

java.sql.PreparedStatement ptmt = connect.prepareStatement(sql);

ptmt.setString(1, fname);

ResultSet rs = ptmt.executeQuery();//创建数据对象

List<Tree> resultTree = new ArrayList<Tree>();

while(rs.next()){

Tree tree = new Tree();

tree.settreeID(rs.getString("treeID"));

tree.setdate(rs.getDate("date"));

tree.settree(rs.getString("tree"));

tree.setsize(rs.getInt("size"));

tree.setnum(rs.getInt("num"));

tree.settree\_soil(rs.getString("tree\_soil"));

tree.setsource(rs.getString("source"));

tree.setfield(rs.getString("field"));

tree.seton\_truck(rs.getInt("on\_truck"));

tree.settransport(rs.getInt("transport"));

tree.setoff\_truck(rs.getInt("off\_truck"));

tree.setcost\_price(rs.getInt("cost\_price"));

tree.setsale\_price(rs.getInt("sale\_price"));

tree.setimg(rs.getString("img"));

tree.setnote(rs.getString("note"));

resultTree.add(tree);

}

if(resultTree!=null) {

table\_1.removeAll();

for(int i = 0 ; i < resultTree.size() ; i++) {

TableItem item;

item = new TableItem(table\_1,SWT.NONE);

item.setText(new String[] {

resultTree.get(i).gettreeID(),

resultTree.get(i).getdate()+"",

resultTree.get(i).gettree(),

resultTree.get(i).getsize()+"",

resultTree.get(i).getnum()+"",

resultTree.get(i).gettree\_soil(),

resultTree.get(i).getsource(),

resultTree.get(i).getfield(),

resultTree.get(i).geton\_truck()+"",

resultTree.get(i).gettransport()+"",

resultTree.get(i).getoff\_truck()+"",

resultTree.get(i).getcost\_price()+"",

resultTree.get(i).getsale\_price()+"",

resultTree.get(i).getimg(),

resultTree.get(i).getnote(),

});

}

}

**7.3.2买家登记事务**

买家登记：

Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/mytree?useSSL=false&serverTimezone=UTC","root","001215");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String buyerID=text.getText();

String buyername=text\_1.getText();

String company=text\_2.getText();

String phone=text\_3.getText();

String sql ="" +

"insert into buyer " +

"(buyerID,buyername,company,phone) " +

"values(?,?,?,?)";

java.sql.PreparedStatement ptmt = connect.prepareStatement(sql);

ptmt.setString(1, buyerID);

ptmt.setString(2, buyername);

ptmt.setString(3, company);

ptmt.setString(4, phone);

ptmt.execute();

买家查询：

Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/mytree?useSSL=false&serverTimezone=UTC","root","001215");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String fname=text.getText();//姓名

String sql = "select \* from buyer "+

"where buyername = ?";

java.sql.PreparedStatement ptmt = connect.prepareStatement(sql);

ptmt.setString(1, fname);

ResultSet rs = ptmt.executeQuery();//创建数据对象

List<Buyer> resultBuyer = new ArrayList<Buyer>();

if(resultBuyer!=null) {

while(rs.next()){

Buyer buyer = new Buyer();

buyer.setbuyerID(rs.getString("buyerID"));

buyer.setbuyername(rs.getString("buyername"));

buyer.setcompany(rs.getString("company"));

buyer.setphone(rs.getString("phone"));

resultBuyer.add(buyer);

}

}

if(resultBuyer!=null) {

table.removeAll();

for(int i = 0 ; i < resultBuyer.size() ; i++) {

TableItem item;

item = new TableItem(table,SWT.NONE);

item.setText(new String[] {

resultBuyer.get(i).getbuyerID(),

resultBuyer.get(i).getbuyername(),

resultBuyer.get(i).getcompany(),

resultBuyer.get(i).getphone(),

});

}

}

**7.3.3苗木销售事务**

销售苗木：

Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/mytree?useSSL=false&serverTimezone=UTC","root","001215");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String treeID=text.getText();

int wantnum=Integer.parseInt(text\_1.getText());

String buyerID=text\_2.getText();

int iday=dateTime.getDay();

int imonth=dateTime.getMonth()+1;

int iyear=dateTime.getYear();

String day,month,year;

if(iday<10) {

day = "0"+iday;

}else{

day = ""+iday;

}

if(imonth<10) {

month = "0"+imonth;

}else{

month = ""+imonth;

}

year = ""+iyear;

DateFormat df = new SimpleDateFormat("yyyy-MM-dd");

java.util.Date u\_date=new java.util.Date();

u\_date = df.parse(year+"-"+month+"-"+day);

java.sql.Date s\_date=new java.sql.Date(u\_date.getTime());

if(wantnum<1) {

MessageBox messageBox = new MessageBox(shell);

messageBox.setMessage("请输入售出数量！");

messageBox.open();

}else {

String sql1 ="" +

"select tree,size,num,tree\_soil,source,field,on\_truck,transport,off\_truck,cost\_price,sale\_price from store " +

"where treeID = ?";

java.sql.PreparedStatement ptmt1 = connect.prepareStatement(sql1);

ptmt1.setString(1, treeID);

ResultSet rs1 = ptmt1.executeQuery();//创建数据对象

List<Tree> resultTree = new ArrayList<Tree>();

while(rs1.next()){

Tree tree = new Tree();

tree.setdate(s\_date);

tree.settree(rs1.getString("tree"));

tree.setsize(rs1.getInt("size"));

tree.setnum(rs1.getInt("num"));

tree.settree\_soil(rs1.getString("tree\_soil"));

tree.setsource(rs1.getString("source"));

tree.setfield(rs1.getString("field"));

tree.seton\_truck(rs1.getInt("on\_truck"));

tree.settransport(rs1.getInt("transport"));

tree.setoff\_truck(rs1.getInt("off\_truck"));

tree.setcost\_price(rs1.getInt("cost\_price"));

tree.setsale\_price(rs1.getInt("sale\_price"));

resultTree.add(tree);

}

String sql2 ="" +

"select buyername,company from buyer " +

"where buyerID =?";

java.sql.PreparedStatement ptmt2 = connect.prepareStatement(sql2);

ptmt2.setString(1, buyerID);

ResultSet rs2 = ptmt2.executeQuery();//创建数据对象

List<Buyer> resultBuyer = new ArrayList<Buyer>();

while(rs2.next()){

Buyer buyer = new Buyer();

buyer.setbuyername(rs2.getString("buyername"));

buyer.setcompany(rs2.getString("company"));

resultBuyer.add(buyer);

}

if(wantnum>resultTree.get(0).getnum()) {

MessageBox messageBox = new MessageBox(shell);

messageBox.setMessage("库存数量不足！");

messageBox.open();

}else {

String sql3 ="" +

"insert into sale " +

"(date,buyername,company,tree,size,wantnum," +

"tree\_soil,source,field,out\_price) " +

"values(" +

"?,?,?,?,?,?,?,?,?,?)";

java.sql.PreparedStatement ptmt3 = connect.prepareStatement(sql3);

ptmt3.setDate(1, s\_date);

ptmt3.setString(2, resultBuyer.get(0).getbuyername());

ptmt3.setString(3, resultBuyer.get(0).getcompany());

ptmt3.setString(4, resultTree.get(0).gettree());

ptmt3.setInt(5, resultTree.get(0).getsize());

ptmt3.setInt(6, wantnum);

ptmt3.setString(7, resultTree.get(0).gettree\_soil());

ptmt3.setString(8, resultTree.get(0).getsource());

ptmt3.setString(9, resultTree.get(0).getfield());

ptmt3.setInt(10, resultTree.get(0).getsale\_price()\*wantnum);

ptmt3.execute();

String sql4 ="" +

"insert into earning " +

"(date,pay,`get`) " +

"values(?,?,?)";

java.sql.PreparedStatement ptmt4 = connect.prepareStatement(sql4);

ptmt4.setDate(1, s\_date);

ptmt4.setInt(2, 0);

ptmt4.setInt(3, resultTree.get(0).getsale\_price()\*wantnum);

ptmt4.execute();

if(wantnum==resultTree.get(0).getnum()) {

String sql5 ="delete from store" +

"where treeID =?";

java.sql.PreparedStatement ptmt5 = connect.prepareStatement(sql5);

ptmt5.setString(1, resultTree.get(1).gettreeID());

ptmt5.execute();

}else {

String sql6 ="" +

"update store " +

"set num =? " +

"where treeID =?";

java.sql.PreparedStatement ptmt6 = connect.prepareStatement(sql6);

ptmt6.setInt(1, resultTree.get(0).getnum()-wantnum);

ptmt6.setString(2, treeID);

ptmt6.execute();

}

MessageBox messageBox = new MessageBox(shell);

messageBox.setMessage("录入成功！");

messageBox.open();

buyerflag = 1;

}

}

销售记录查询：

Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/mytree?useSSL=false&serverTimezone=UTC","root","001215");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String year1=combo.getText();

String month1=Integer.parseInt(combo\_1.getText())+"";

String day1=combo\_2.getText();

String year2=combo\_3.getText();

String month2=Integer.parseInt(combo\_4.getText())+"";

String day2=combo\_5.getText();

DateFormat df = new SimpleDateFormat("yyyy-MM-dd");

java.util.Date u\_date1=new java.util.Date();

java.util.Date u\_date2=new java.util.Date();

u\_date1 = df.parse(year1+"-"+month1+"-"+day1);

u\_date2 = df.parse(year2+"-"+month2+"-"+day2);

java.sql.Date s\_date1=new java.sql.Date(u\_date1.getTime());

java.sql.Date s\_date2=new java.sql.Date(u\_date2.getTime());

String sql ="" +

"select \* from sale " +

"where date between ? and ?";

java.sql.PreparedStatement ptmt = connect.prepareStatement(sql);

ptmt.setDate(1, s\_date1);

ptmt.setDate(2, s\_date2);

ResultSet rs = ptmt.executeQuery();//创建数据对象

List<Trade> resultTrade = new ArrayList<Trade>();

while(rs.next()){

Trade trade = new Trade();

trade.setsalenum(Integer.parseInt(rs.getString("salenum")));

trade.setbuyername(rs.getString("buyername"));

trade.setcompany(rs.getString("company"));

trade.setdate(rs.getDate("date"));

trade.settree(rs.getString("tree"));

trade.setsize(rs.getInt("size"));

trade.setwantnum(rs.getInt("wantnum"));

trade.settree\_soil(rs.getString("tree\_soil"));

trade.setsource(rs.getString("source"));

trade.setfield(rs.getString("field"));

trade.setout\_price(rs.getInt("out\_price"));

trade.setnote(rs.getString("note"));

resultTrade.add(trade);

}

if(resultTrade!=null) {

table.removeAll();

for(int i = 0 ; i < resultTrade.size() ; i++) {

TableItem item;

item = new TableItem(table,SWT.NONE);

item.setText(new String[] {

resultTrade.get(i).getsalenum()+"",

resultTrade.get(i).getdate()+"",

resultTrade.get(i).getbuyername(),

resultTrade.get(i).getcompany(),

resultTrade.get(i).gettree(),

resultTrade.get(i).getsize()+"",

resultTrade.get(i).getwantnum()+"",

resultTrade.get(i).gettree\_soil(),

resultTrade.get(i).getsource(),

resultTrade.get(i).getfield(),

resultTrade.get(i).getout\_price()+"",

resultTrade.get(i).getnote(),

});

}

}

**7.3.4收益合计事务**

收益查询统计：

Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/mytree?useSSL=false&serverTimezone=UTC","root","001215");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String year1=combo.getText();

String month1=Integer.parseInt(combo\_1.getText())+"";

String day1=combo\_2.getText();

String year2=combo\_3.getText();

String month2=Integer.parseInt(combo\_4.getText())+"";

String day2=combo\_5.getText();

DateFormat df = new SimpleDateFormat("yyyy-MM-dd");

java.util.Date u\_date1=new java.util.Date();

java.util.Date u\_date2=new java.util.Date();

u\_date1 = df.parse(year1+"-"+month1+"-"+day1);

u\_date2 = df.parse(year2+"-"+month2+"-"+day2);

java.sql.Date s\_date1=new java.sql.Date(u\_date1.getTime());

java.sql.Date s\_date2=new java.sql.Date(u\_date2.getTime());

String sql ="" +

"select \* from earning " +

"where date between ? and ?";

java.sql.PreparedStatement ptmt = connect.prepareStatement(sql);

ptmt.setDate(1, s\_date1);

ptmt.setDate(2, s\_date2);

ResultSet rs = ptmt.executeQuery();//创建数据对象

List<Money> resultmoney = new ArrayList<Money>();

while(rs.next()){

Money money = new Money();

money.setmoneynum(rs.getInt("moneynum"));

money.setdate(rs.getDate("date"));

money.setpay(rs.getInt("pay"));

money.setget(rs.getInt("get"));

resultmoney.add(money);

}

if(resultmoney!=null) {

table.removeAll();

for(int i = 0 ; i < resultmoney.size() ; i++) {

TableItem item;

item = new TableItem(table,SWT.NONE);

item.setText(new String[] {

resultmoney.get(i).getmoneynum()+"",

resultmoney.get(i).getdate()+"",

resultmoney.get(i).getpay()+"",

resultmoney.get(i).getget()+"",

});

a.add(i, resultmoney.get(i).getpay()+"");

b.add(i, resultmoney.get(i).getget()+"");

}

}

flag = 1;

求和：

public void widgetSelected(SelectionEvent event){

if(flag==0) {

MessageBox messageBox = new MessageBox(shell);

messageBox.setMessage("请先查询！");

messageBox.open();

}else {

int ii=0,jj=0,heji;

for(int i=0;i<a.size();i++) {

ii+=Integer.parseInt(a.get(i));

}

for(int j=0;j<b.size();j++) {

jj+=Integer.parseInt(b.get(j));

}

if(ii>jj) {

heji=ii-jj;

text.setText("-"+heji);

}else {

heji=jj-ii;

text.setText(""+heji);

}

}

}

# 八、测试和运行

**8.1.1 登录界面**

用户在登录界面输入用户名和密码并经过系统验证后，可以进入主服务界面进行相关操作（账号密码为初始设定好的值）



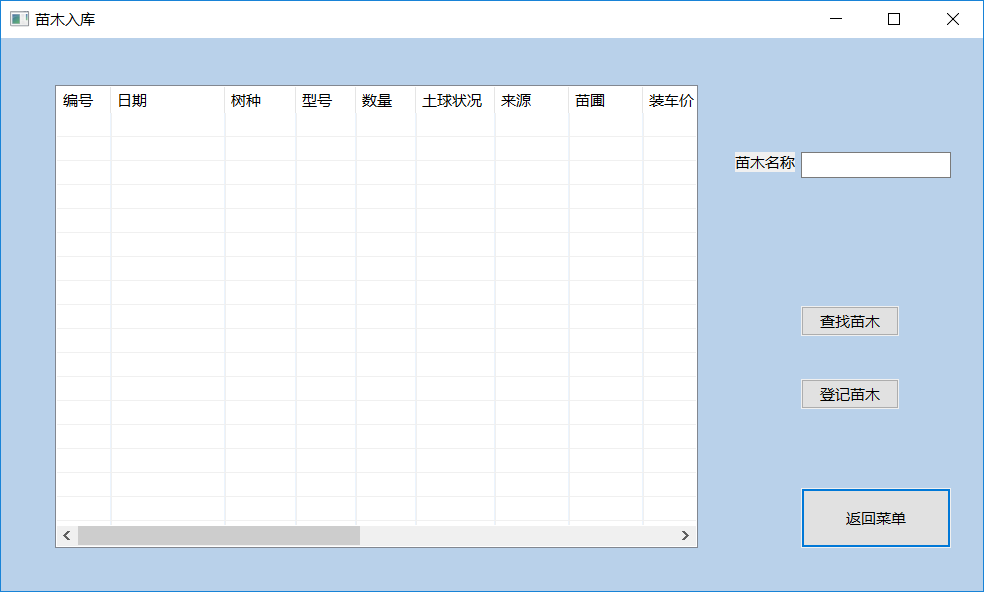
**8.1.2 主服务界面**

在输入正确的账号密码后进入主服务界面，主服务界面提供了通往目的功能的按钮，在按下按钮后主服务界面dispose，显示新的界面。

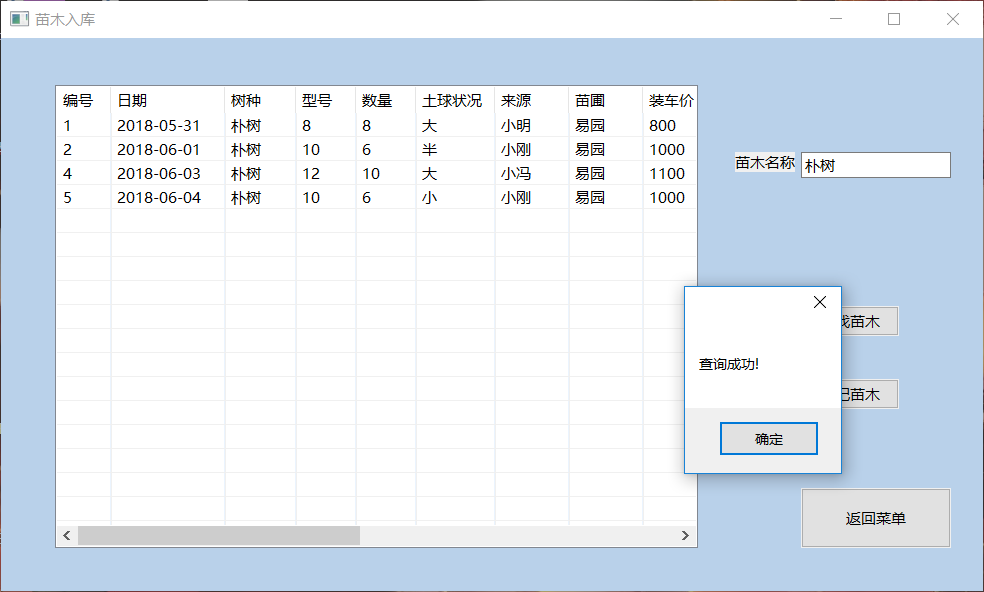


* + 1. **苗木信息查询**

（1）苗木信息查询界面如下图所示：



（2）查询结果

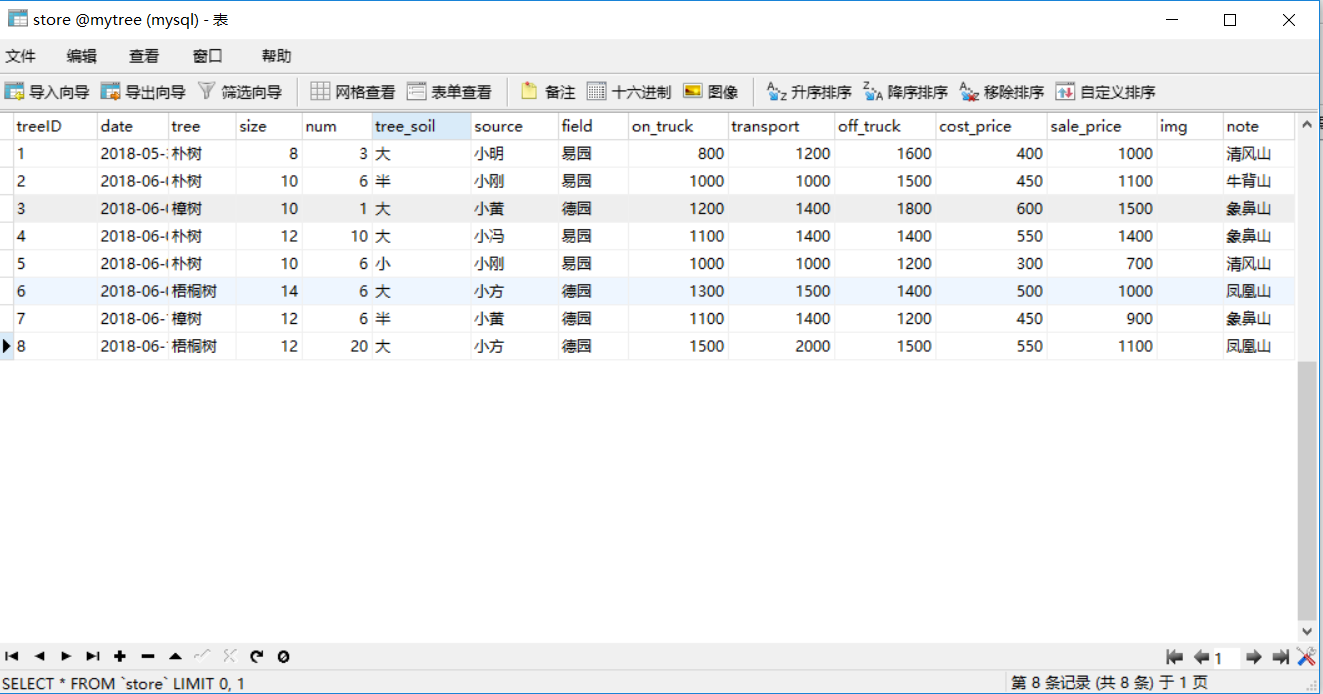


（3）苗木入库登记

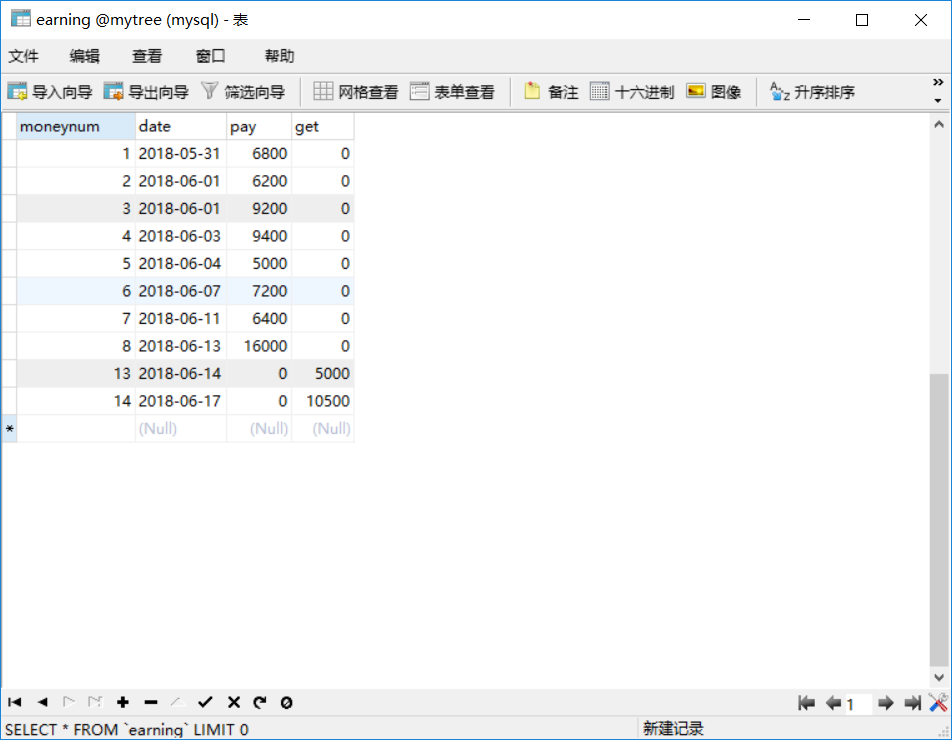


（4）完成登记后更新相关表

苗木库存信息表（编号8所在行）

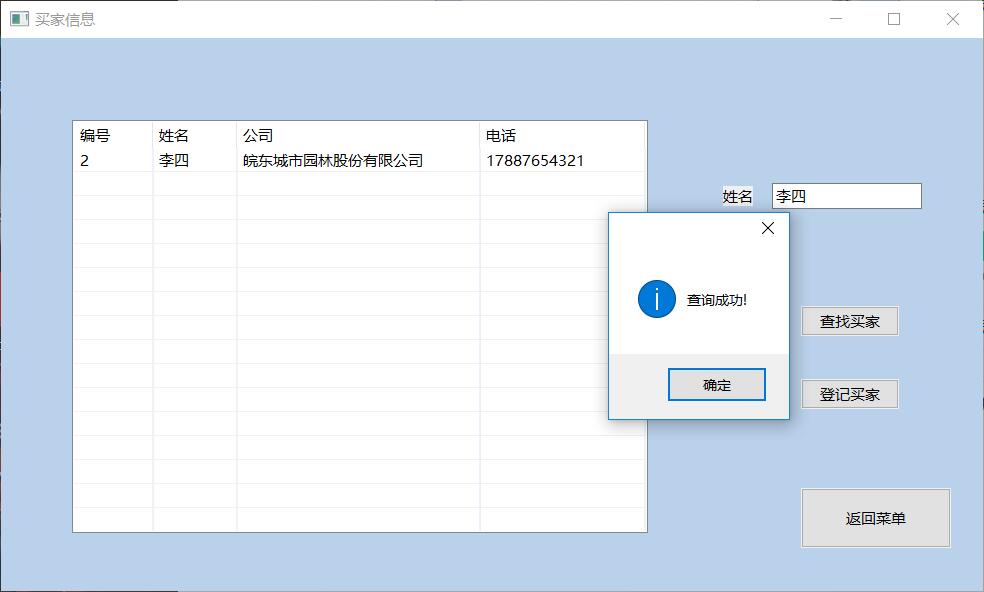


收支信息表（编号8所在行）

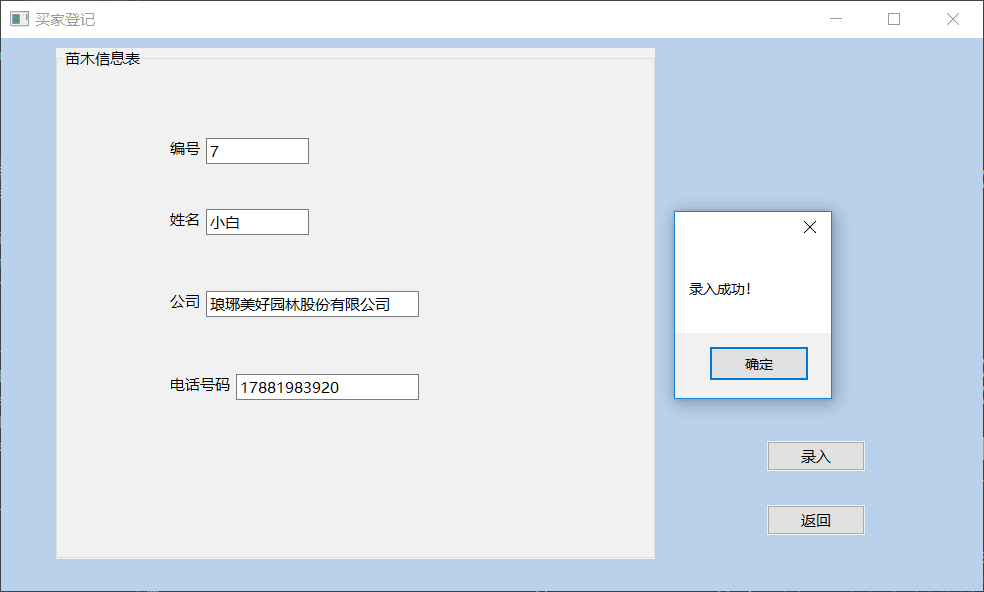


* + 1. **买家信息查询**

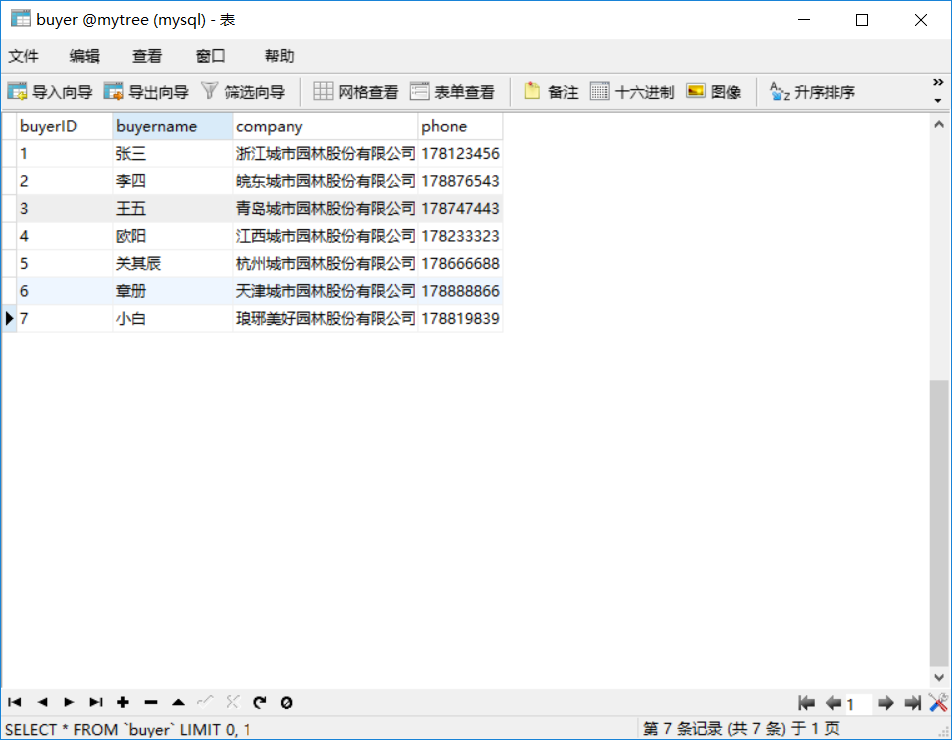
（1）买家信息查询界面如下图所示：



（2）买家信息登记界面如下图所示：

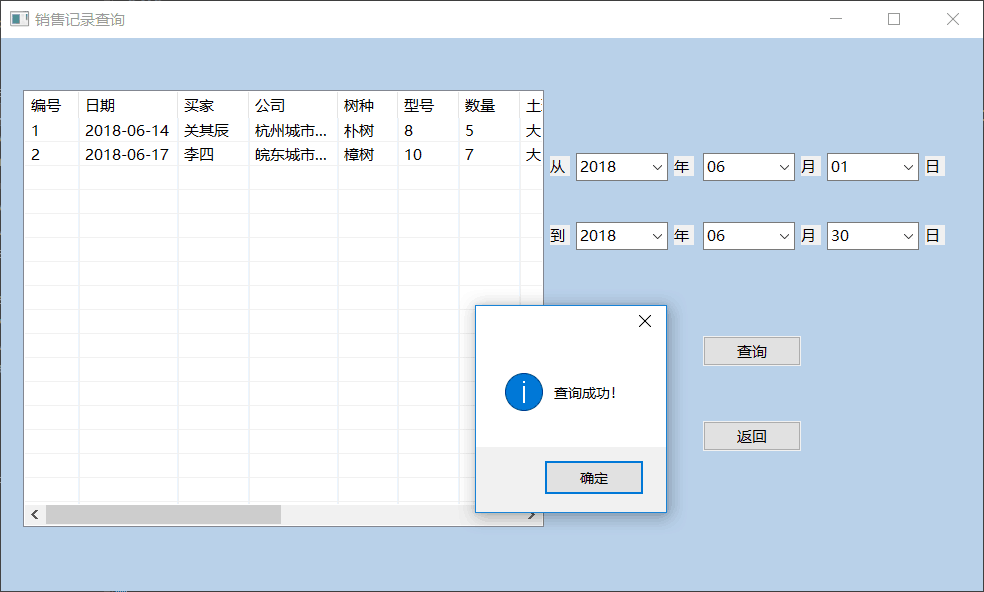
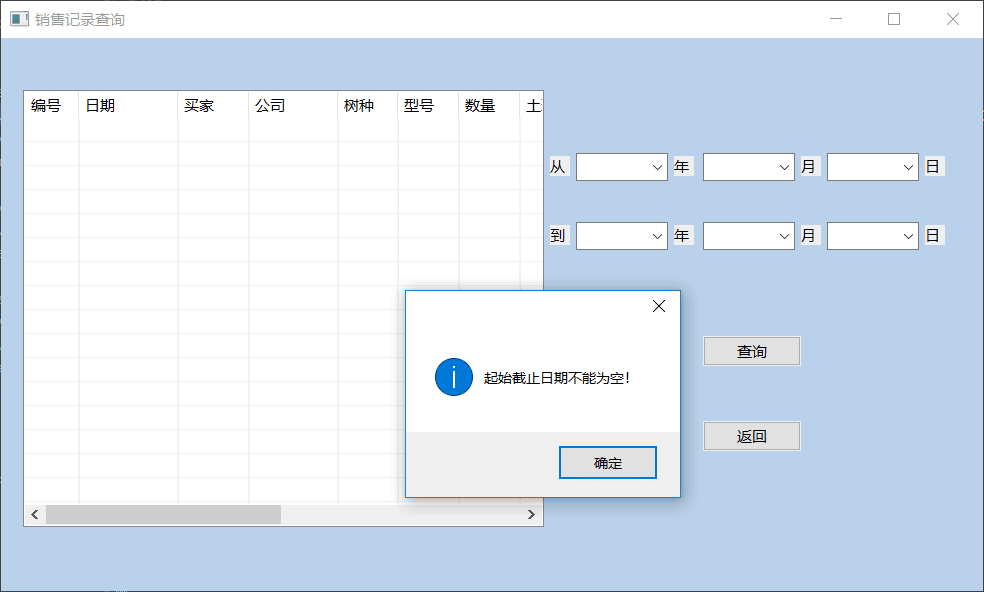


（3）完成登记后更新相关表如下图所示：

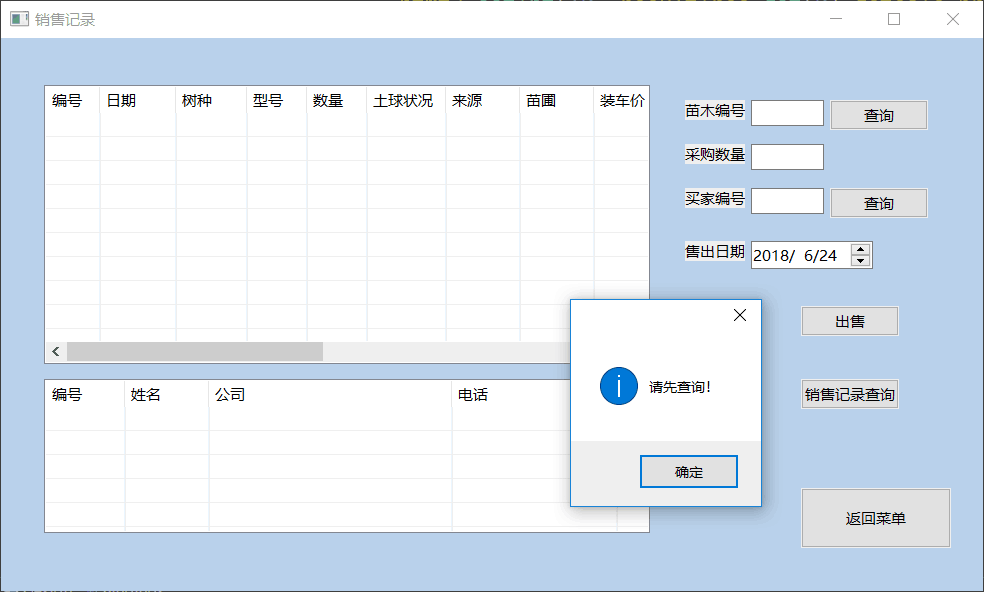


**8.1.5 销售信息查询**

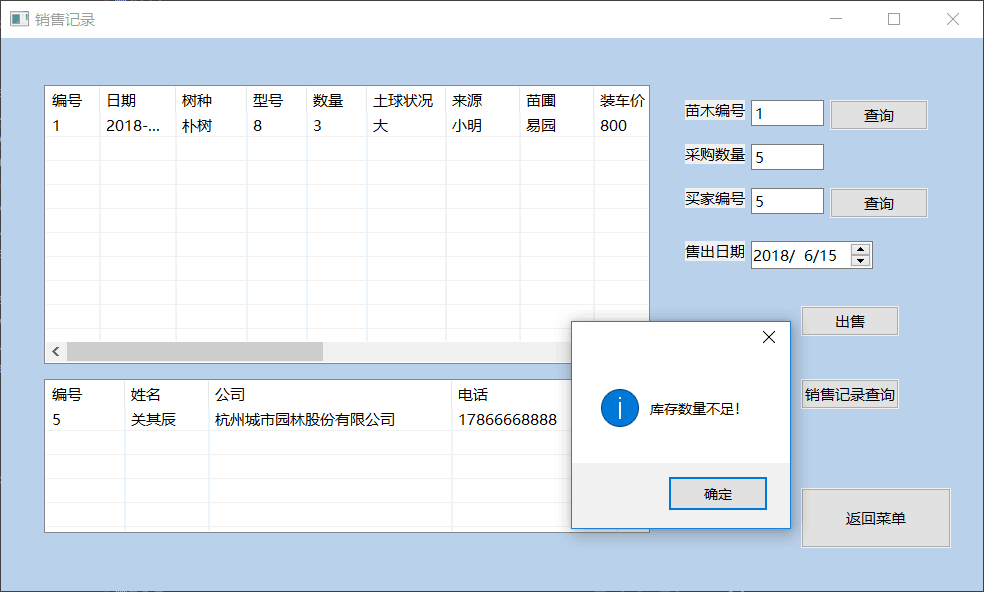
（1）销售信息查询界面如下图所示：



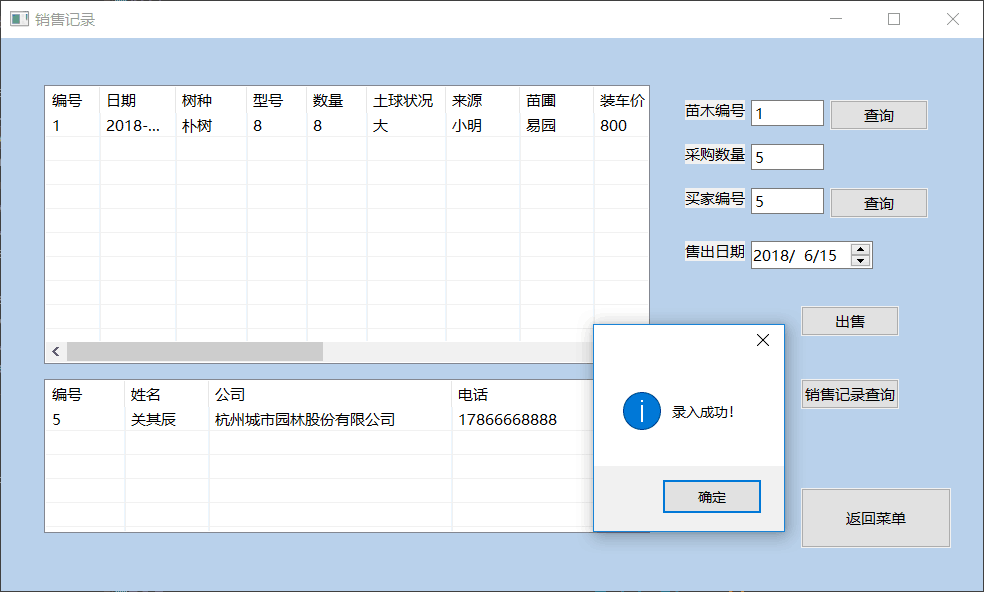
（2）销售过程界面如下图所示：



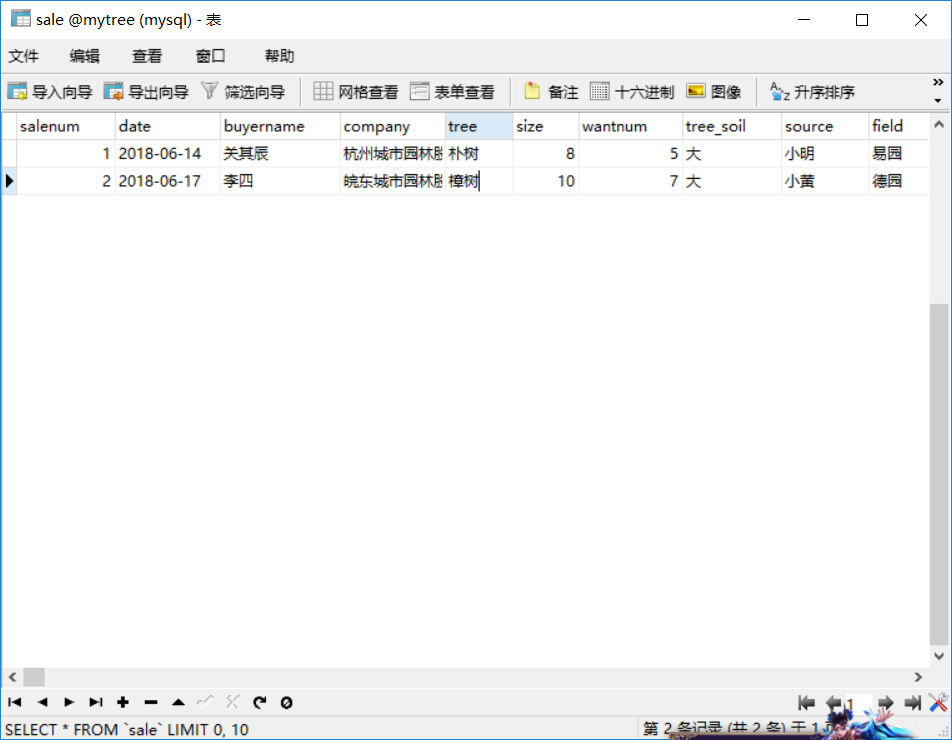
（3）销售库存数量判断界面如下图所示：

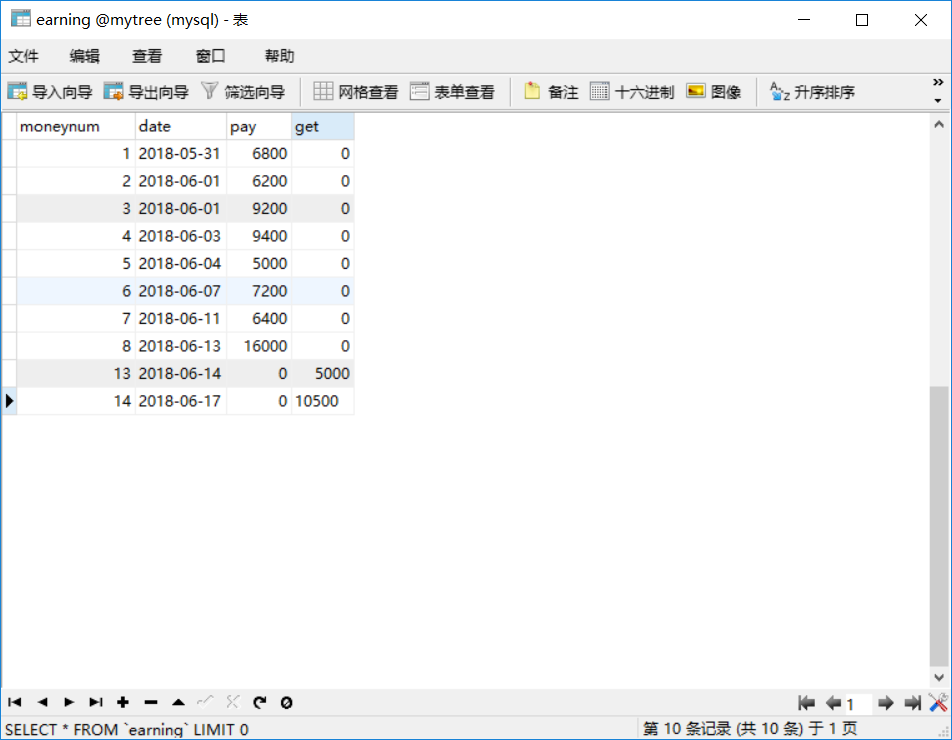


（4）销售成功结果界面如下图所示：



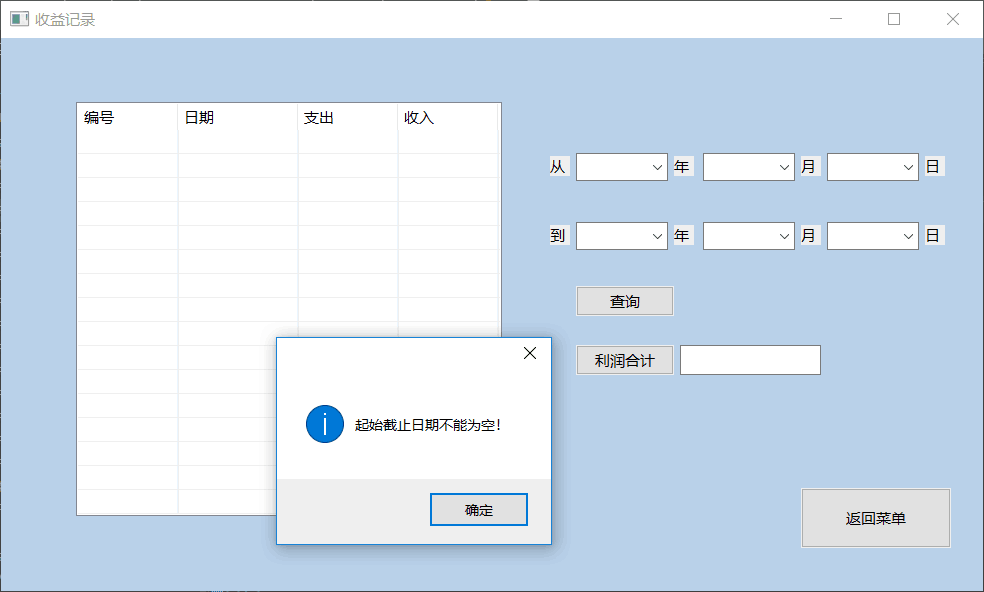
（5）完成登记后更新相关表如下图所示：



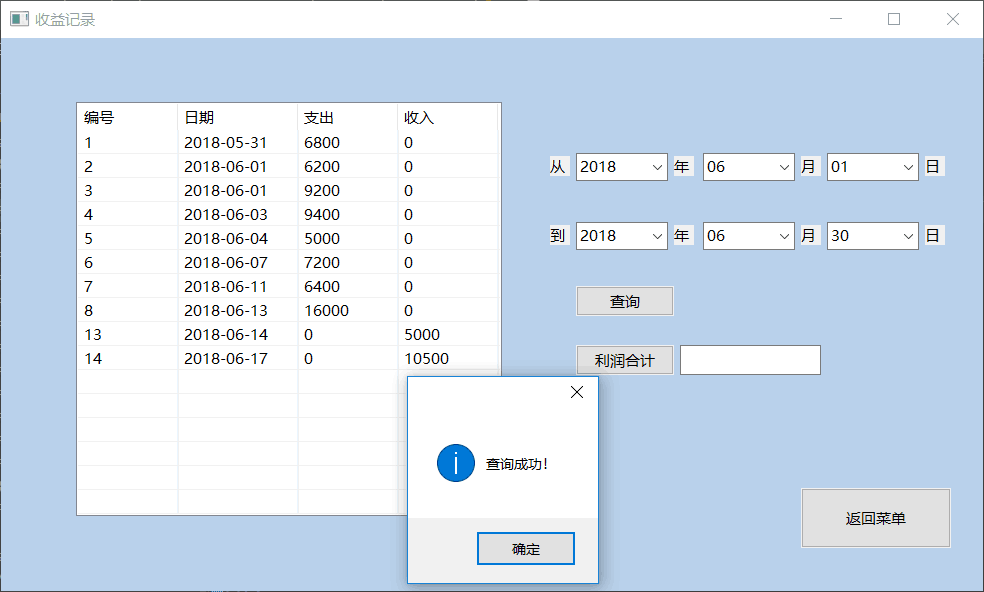


**8.1.6 收益信息查询**

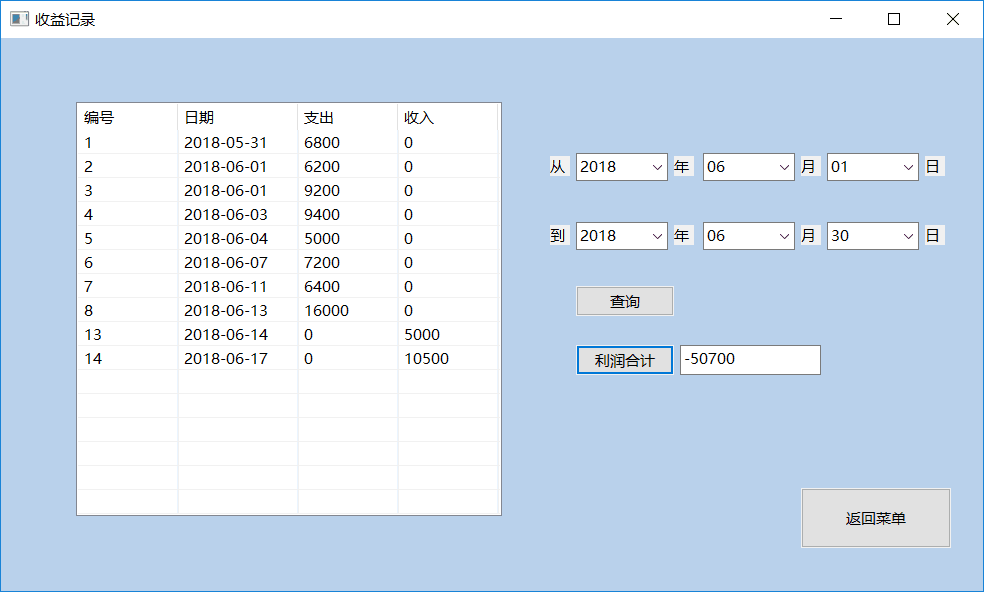
（1）收益信息查询界面



（2）收益信息查询结果：



（3）收益信息金额求和



# 九、总结

## 9.1 系统优点

1、系统设计全面、功能齐全，在有限时间内将需求的功能完全实现，设计十分清晰明快能够很好的满足实际操作需求。

2、界面美观，软件易用性高，操作简单，能够很好的满足需求

3、数据库设计内容具体详细，条理清晰，关系明确，能够遵循数据库设计的准则来描述信息关系，可以稳定地为系统提供服务。

4、信息提示系统细致完善，对于用户可能发生的错误操作，给予错误信息提示。

5、对异常进行了处理，在用户操作不规范时给出错误信息

## 9.2 系统不足

1、系统的稳定性不高，长时间的运行会导致程序的运行缓慢且影响软件的效率，甚至数据丢失。

2、数据库的设计存在一些问题，可以进行进一步的优化

3、项目涉及到的是商业方面的使用，系统安全性不高，可能会遭到攻击

4、可以增加对苗木实景图的处理展示，将图片放在一个文件路径下，将各自路径名称导入到数据库中

## 9.3 系统改进

可以增加对苗木实景图的处理展示，将图片放在一个文件路径下，将各自路径名称导入到数据库中。可以增加一些删除操作对旧的过时数据进行清理，增加修改操作提高容错率。

## 9.4 经验与收获

经过这为期八周的课程设计，让我认识到从理论跨越到实践的难度，更人性化的设计背后都需要更多的操作代码去支持。

在这次系统的设计和实现过程中，我更加深刻的了解了一个成熟的系统的开发流程，也意识到了前期准备的重要性。并且，通过本次的课程设计，我亲身体会到了MVC Model模型的开发优势，这集中体现在MVC将业务和表现分离，程序的扩展性和维护性很强，程序非常易于修改。并且由于c/s架构，与用户的交互性提高了。

当然，本次课程设计我制作的系统还是有一些缺陷的，由于时间较短，所以我的数据库设计上一些细节还需要优化。并且苗木销售管理系统是一个较为复杂的系统，而我做的系统并未投入使用，所以在一些功能上、安全性上、稳定性上可能存在漏洞，但是由于采用MVC模型，所以可以在后期需要的时候进行扩展。