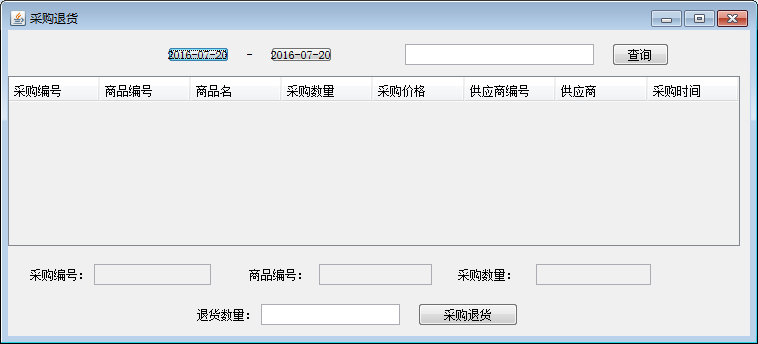
**进销存整体业务**

**商品采购业务**

2.采购退货



确定采购单 - 填写退货数量 —— 添加到采购退货表

—— 修改商品库存

2.1 查询采购单

2.1.1 创建视图：

|  |
| --- |
| Create view view\_purchase  as  SELECT  purchase.purid,  purchase.purno,  purchase.pid,  purchase.purchaseamount,  purchase.purchaseprice,  purchase.purtotals,  purchase.supplierid,  purchase.purchasedate,  product2.pname,  supplier.supname  FROM  purchase ,  product2 ,  supplier  WHERE  purchase.pid = product2.pid AND  purchase.supplierid = supplier.supplierid |

2.1.2 创建实体类ViewPurchase ,对应视图

|  |
| --- |
| public class ViewPurchase {  private Integer purid;  private String purno;  private Integer pid;  private Integer purchaseamount;  private BigDecimal purchaseprice;  private Integer supplierid;  private Date purchasedate;  private Integer state;  private String pname;  private String supname;  //set get方法  } |

2.1.3 PurchaseBiz增加业务接口

// 查询采购单

public List<ViewPurchase> findByCondition(String start,String end,String conditions);

2.1.4 PurchaseBizImpl实现方法

|  |
| --- |
| public List<ViewPurchase> findByCondition(String start, String end, String conditions) {  String sql = "select \* from view\_purchase where purchasedate between '"+start+"' and '"+end+"'";  if(conditions != null && conditions.length()>0){  sql +=" and concat(purno,pname,supname) like '%"+conditions+"%'";  }  return purdao.query(sql, ViewPurchase.class);  } |

2.1.5 PurchaseOutFrame 中给“查询”按钮增加事件

|  |
| --- |
| //按条件查询采购单  //1获取条件  String start = this.dateStart.getText();  String end = this.dateEnd.getText();  String conditions = this.txtConditions.getText().trim();  //调用业务  List<ViewPurchase> list = pbiz.findByCondition(start, end, conditions);  //显示在表格中  showOnTable(list); |

2.1.6 showOnTable(list)方法：

|  |
| --- |
| private void showOnTable(List<ViewPurchase> list) {  //1 表格模型  DefaultTableModel dtm = (DefaultTableModel) this.tblPurchase.getModel();  //2清空表格  while(dtm.getRowCount()>0){  dtm.removeRow(0);  }  //3 添加数据  for(ViewPurchase pur:list){  Vector vt = new Vector();  vt.add(pur.getPurid());  vt.add(pur.getPid());  vt.add(pur.getPname());  vt.add(pur.getPurchaseamount());  vt.add(pur.getPurchaseprice());  vt.add(pur.getSupplierid());  vt.add(pur.getSupname());  vt.add(pur.getPurchasedate());  dtm.addRow(vt);    }  } |

2.1.7 JTable隐藏列方法

|  |
| --- |
| public void HiddenCell(JTable table, int column) {  TableColumn tc = table.getTableHeader().getColumnModel().getColumn(  column);  tc.setMaxWidth(0);  tc.setPreferredWidth(0);  tc.setWidth(0);  tc.setMinWidth(0);  table.getTableHeader().getColumnModel().getColumn(column)  .setMaxWidth(0);  table.getTableHeader().getColumnModel().getColumn(column)  .setMinWidth(0);  } |

2.1.8 构造方法中调用

HiddenCell(this.tblPurchase,1); //隐藏第1列

HiddenCell(this.tblPurchase,5); //隐藏第5列

2.2 获取要退货信息（table的鼠标单击事件）

|  |
| --- |
| selectrow = this.tblPurchase.getSelectedRow();//selectrow为类属性  Integer purid = (Integer) this.tblPurchase.getValueAt(selectrow, 0);  Integer pid = (Integer) this.tblPurchase.getValueAt(selectrow, 1);  Integer purchaseamount = (Integer) this.tblPurchase.getValueAt(selectrow, 3);  //显示在文本框  this.txtPurid.setText(purid.toString());  this.txtPid.setText(pid.toString());  this.txtPurchaseamount.setText(purchaseamount.toString()); |

2.3 “退货”按钮数据获取和验证

|  |
| --- |
| if(selectrow == -1){  JOptionPane.showMessageDialog(this, "请选择要退货的采购单信息");  return;  }  //获取退货信息  Integer purid = Integer.parseInt(this.txtPurid.getText());  Integer pid = Integer.parseInt(this.txtPid.getText());  Integer purchaseamount = Integer.parseInt(this.txtPurchaseamount.getText());  //获取退货数量  String sreturnAmount = this.txtReturnAmount.getText();  //数据验证  String regex = "[1-9][0-9]\*";  if(sreturnAmount.matches(regex)== false){  JOptionPane.showMessageDialog(this, "请填写整数");  return;  }  //转换退货数量  Integer returnAmount = Integer.parseInt(sreturnAmount);  //判断是否超过采购数量  if(returnAmount>purchaseamount){  JOptionPane.showMessageDialog(this, "退货数量大于采购数量");  return;  }    //调用业务  java.sql.Date returntime = new java.sql.Date( new java.util.Date().getTime());  boolean result = pbiz.purchasereturn(purid,returnAmount,returntime, pid);  if(result == true){  JOptionPane.showMessageDialog(this, "退货成功");  btnQueryActionPerformed(evt);  }else{  JOptionPane.showMessageDialog(this, "退货失败");  }  //清空文本框  this.txtPid.setText("");  this.txtPurid.setText("");  this.txtPurchaseamount.setText("");  this.txtReturnAmount.setText("");  selectrow = -1; |

BaseDao增加更新方法重载

|  |
| --- |
| /\*\*     \* 更新数据库重载方法     \* @param conn 连接对象     \* @param sql sql语句     \* @param params 参数     \* @return      \*/     public boolean update(Connection conn,String sql,Object[] params) throws SQLException {             boolean flag = false;             QueryRunner qRunner = new QueryRunner();             int i = qRunner.update(conn,sql,params);             if (i > 0) {                 flag = true;             }             return flag;     } |

purchasereturn 是PurchaseBizImpl业务类的退货方法：

PurchaseBiz增加业务接口

public boolean purchasereturn(int purid,int returnamont,Date returntime,int pid);

PurchaseBizImpl 业务实现

|  |
| --- |
| public boolean purchasereturn(int purid, int returnamount, Date returntime, int pid) {  boolean result = true;  //修改采购表的退货数量和退货时间  String sql1 = "update purchase set returnamount=?,returntime=?,purchaseamount=purchaseamount-? where purid=?";  Object[] params1 = {returnamount,returntime,returnamount,purid};  String sql2 = "update product2 set currentamount= currentamount-? where pid=?";  Object[] params2 = {returnamount,pid};  Connection conn = new BaseDao().getConnection();  try {    conn.setAutoCommit(false);//启动事务  purdao.update(conn, sql1, params1);  prodao.update(conn, sql2, params2);  conn.commit();  } catch (Exception e) {  try {  result = false;  conn.rollback();  } catch (Exception ex) {  ex.printStackTrace();  }  }  return result;  }    } |

注：