package total.store.dao.impl;  
  
import total.store.dao.OrderDao;  
import total.store.model.Order;  
import total.store.util.DbUtil;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
  
public class OrderDaoImpl implements OrderDao {  
 @Override  
 public Boolean orderid\_single(String orderid) {  
 DbUtil dbUtil = new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null){  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("select \* from 订单信息表 where 编号=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,orderid);  
 rs = ps.executeQuery();  
 String num="";  
 while (rs.next()){  
 num = rs.getString("编号");  
 }  
 if(num==""){  
 return true;  
 }else {  
 return false;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public Boolean order\_add(Order order) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con=null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
 String dateNow = sdf.format(new Date());  
 StringBuilder sb = new StringBuilder();  
 sb.append("insert into 订单信息表 values(?,?,?,?,?,?,'待支付')");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,order.getOrderid());  
 ps.setInt(2,order.getBookinformation());  
 ps.setInt(3,order.getNumber());  
 ps.setInt(4,order.getAdressinformation());  
 ps.setDouble(5,order.getOrderprice());  
 ps.setString(6,dateNow);  
 int num=ps.executeUpdate();  
 if(num==1){  
 return true;  
 }else {  
 return false;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public List<Order> order\_list(String shopid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("select \* from 订单信息表,书籍信息表,地址表 where 订单信息表.订单状态 = '待发货' and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 书籍信息表.商铺信息=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,shopid);  
 rs=ps.executeQuery();  
 List<Order> orderList=new ArrayList<>();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setBookinformation(rs.getInt("书籍信息"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrderconsignee(rs.getString("收货人"));  
 order.setOrderuserphone(rs.getString("电话"));  
 order.setOrderaddressdetail(rs.getString("详细地址"));  
 orderList.add(order);  
 }  
 return orderList;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public List<Order> order\_user\_list(int userid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("select \* from 订单信息表,书籍信息表,地址表,商铺信息表 where 订单信息表.订单状态 = '待支付' and 书籍信息表.商铺信息=商铺信息表.编号 and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 地址表.用户信息=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setInt(1,userid);  
 rs=ps.executeQuery();  
 List<Order> orderuserList=new ArrayList<>();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrdershopname(rs.getString("商铺名"));  
 order.setOrderstatus(rs.getString("订单状态"));  
 orderuserList.add(order);  
 }  
 StringBuilder sb1 = new StringBuilder();  
 sb1.append("select \* from 订单信息表,书籍信息表,地址表,商铺信息表 where 订单信息表.订单状态 = '待收货' and 书籍信息表.商铺信息=商铺信息表.编号 and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 地址表.用户信息=?");  
 ps = con.prepareStatement(sb1.toString());  
 ps.setInt(1,userid);  
 rs=ps.executeQuery();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrdershopname(rs.getString("商铺名"));  
 order.setOrderstatus(rs.getString("订单状态"));  
 orderuserList.add(order);  
 }  
 StringBuilder sb2 = new StringBuilder();  
 sb2.append("select \* from 订单信息表,书籍信息表,地址表,商铺信息表 where 订单信息表.订单状态 = '待发货' and 书籍信息表.商铺信息=商铺信息表.编号 and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 地址表.用户信息=?");  
 ps = con.prepareStatement(sb2.toString());  
 ps.setInt(1,userid);  
 rs=ps.executeQuery();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrdershopname(rs.getString("商铺名"));  
 order.setOrderstatus(rs.getString("订单状态"));  
 orderuserList.add(order);  
 }  
 StringBuilder sb3 = new StringBuilder();  
 sb3.append("select \* from 订单信息表,书籍信息表,地址表,商铺信息表 where 订单信息表.订单状态 = '已发货' and 书籍信息表.商铺信息=商铺信息表.编号 and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 地址表.用户信息=?");  
 ps = con.prepareStatement(sb3.toString());  
 ps.setInt(1,userid);  
 rs=ps.executeQuery();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrdershopname(rs.getString("商铺名"));  
 order.setOrderstatus(rs.getString("订单状态"));  
 orderuserList.add(order);  
 }  
 return orderuserList;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public Boolean order\_buy(String orderid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("update 订单信息表 set 订单状态='待发货' where 编号=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,orderid);  
 int num = ps.executeUpdate();  
 if (num==1){  
 return true;  
 }else {  
 return false;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public int order\_num(String orderid) {  
 DbUtil dbUtil = new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null){  
 return 0;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("select 数量 from 订单信息表 where 编号=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,orderid);  
 rs = ps.executeQuery();  
 int num = 0;  
 while (rs.next()){  
 num = rs.getInt("数量");  
 }  
 return num;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return 0;  
 }  
  
 @Override  
 public int order\_book(String orderid) {  
 DbUtil dbUtil = new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null){  
 return 0;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("select 书籍信息 from 订单信息表 where 编号=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,orderid);  
 rs = ps.executeQuery();  
 int num = 0;  
 while (rs.next()){  
 num = rs.getInt("书籍信息");  
 }  
 return num;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return 0;  
 }  
  
 @Override  
 public Boolean order\_delete(String orderid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con=null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("delete from 订单信息表 where 编号=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,orderid);  
 int num=ps.executeUpdate();  
 if(num==1){  
 return true;  
 }else {  
 return false;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public Boolean order\_shop\_modity(String orderid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("update 订单信息表 set 订单状态='已发货' where 编号=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,orderid);  
 int num = ps.executeUpdate();  
 if (num==1){  
 return true;  
 }else {  
 return false;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public List<Order> order\_history\_list(String shopid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("select \* from 订单信息表,书籍信息表,地址表 where 订单信息表.订单状态 = '已发货' and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 书籍信息表.商铺信息=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,shopid);  
 rs=ps.executeQuery();  
 List<Order> orderList=new ArrayList<>();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setBookinformation(rs.getInt("书籍信息"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrderconsignee(rs.getString("收货人"));  
 order.setOrderuserphone(rs.getString("电话"));  
 order.setOrderaddressdetail(rs.getString("详细地址"));  
 orderList.add(order);  
 }  
 StringBuilder sb1 = new StringBuilder();  
 sb1.append("select \* from 订单信息表,书籍信息表,地址表 where 订单信息表.订单状态 = '已签收' and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 书籍信息表.商铺信息=?");  
 ps = con.prepareStatement(sb1.toString());  
 ps.setString(1,shopid);  
 rs=ps.executeQuery();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setBookinformation(rs.getInt("书籍信息"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrderconsignee(rs.getString("收货人"));  
 order.setOrderuserphone(rs.getString("电话"));  
 order.setOrderaddressdetail(rs.getString("详细地址"));  
 orderList.add(order);  
 }  
 return orderList;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public Boolean order\_shmodity(String orderid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("update 订单信息表 set 订单状态='已收货' where 编号=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setString(1,orderid);  
 int num = ps.executeUpdate();  
 if (num==1){  
 return true;  
 }else {  
 return false;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
  
 @Override  
 public List<Order> order\_user\_history\_list(int userid) {  
 DbUtil dbUtil =new DbUtil();  
 ResultSet rs = null;  
 PreparedStatement ps = null;  
 Connection con = null;  
 try {  
 con = dbUtil.getCon();  
 if (con == null) {  
 return null;  
 }  
 StringBuilder sb = new StringBuilder();  
 sb.append("select \* from 订单信息表,书籍信息表,地址表,商铺信息表 where 订单信息表.订单状态 = '已收货' and 书籍信息表.商铺信息=商铺信息表.编号 and 订单信息表.地址信息=地址表.编号 and 订单信息表.书籍信息=书籍信息表.编号 and 地址表.用户信息=?");  
 ps = con.prepareStatement(sb.toString());  
 ps.setInt(1,userid);  
 rs=ps.executeQuery();  
 List<Order> orderuserList=new ArrayList<>();  
 while(rs.next()) {  
 Order order = new Order();  
 order.setOrderid(rs.getString("编号"));  
 order.setOrderbookname(rs.getString("书籍名"));  
 order.setNumber(Integer.parseInt(rs.getString("数量")));  
 Date date=rs.getDate("订单时间");  
 order.setOrderdate(date);  
 order.setOrdershopname(rs.getString("商铺名"));  
 order.setOrderstatus(rs.getString("订单状态"));  
 orderuserList.add(order);  
 }  
 return orderuserList;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 dbUtil.closeResultSet(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closeCon(con);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 try {  
 dbUtil.closePs(ps);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return null;  
 }  
}