package total.store.dao.impl;

import total.store.dao.CartDao;

import total.store.model.Auditor;

import total.store.model.Cart;

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 CareDaoImpl implements CartDao {

@Override

public Boolean cart\_one(Cart cart) {

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 购物车 where 用户编号 = ? and 书籍编号 = ?");

ps = con.prepareStatement(sb.toString());

ps.setInt(1,cart.getUserCart());

ps.setInt(2,cart.getBookinfor());

rs = ps.executeQuery();

int bookid = -1;

while(rs.next()) {

bookid = rs.getInt("书籍编号");

}

if (bookid>0){

return false;

}else {

return true;

}

} 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 cart\_add(Cart cart) {

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("insert into 购物车 values(?,?)");

ps = con.prepareStatement(sb.toString());

ps.setInt(1,cart.getUserCart());

ps.setInt(2,cart.getBookinfor());

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<Cart> cart\_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 购物车,书籍信息表 where 用户编号 = ? and 书籍信息表.编号=购物车.书籍编号");

ps = con.prepareStatement(sb.toString());

ps.setInt(1,userid);

rs=ps.executeQuery();

List<Cart> cartList = new ArrayList<>();

while(rs.next()) {

Cart cart = new Cart();

cart.setBookinfor(rs.getInt("书籍编号"));

cart.setBookname(rs.getString("书籍名"));

cart.setBook\_img(rs.getString("书籍封面"));

cart.setBookprice(rs.getDouble("价格"));

cartList.add(cart);

}

return cartList;

} 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 cart\_del(Cart cart) {

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 购物车 where 用户编号=? and 书籍编号=?");

ps = con.prepareStatement(sb.toString());

ps.setInt(1,cart.getUserCart());

ps.setInt(2,cart.getBookinfor());

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;

}

}