

南京信息工程大学

《数据库系统原理》课程设计



题 目 小型超市商品管理系统设计与实现

学生姓名 _____ 徐杨 朱宸扬 _____

学 号 202113880067 202183760012

学 院 _____ 计算机学院 _____

专 业 _____ 信息安全 _____

指导教师 _____ 马瑞 _____

二〇二三年六月二十日

目 录

1	引言	3
1.1	课题背景和意义	3
1.2	课题内容	3
2	系统需求分析及技术介绍	3
2.1	功能需求分析	3
2.2	可行性分析	3
2.3	系统运行环境	4
2.4	相关技术介绍	4
3	系统总体设计	4
3.1	系统功能结构设计	4
3.2	系统功能流程设计	5
3.3	数据库设计	7
4	系统详细设计	11
4.1	登录认证模块详细设计	11
4.2	常规查询模块详细设计	11
4.3	精确查询模块详细设计	12
4.4	增、删、改模块详细设计	12
4.5	数据处理模块详细设计	12
4.6	UI 界面详细设计	12
5	系统实现	13
5.1	登录功能实现	13
5.2	查询模块实现	15
5.3	修改模块实现	21
6	总结	37

小型超市商品管理系统设计与实现

徐杨、朱宸扬

南京信息工程大学计算机与网络安全学院，江苏 南京 210044

1、引言

1.1 课题背景和意义

在当今科技飞速发展条件下，超市管理逐步从传统经营转向自动化、科技化、智能化，超市管理系统在计算机功能发展下产生。超市管理系统是一种基于计算机技术的对超市集合管理的多功能系统，它能够帮助超市提高运营效率、提高运营准确率、提升工作人员和客户的满意度、降低成本等。随着互联网普及，超市管理系统的应用愈增愈多。

1.2 课题内容

本课题旨在开发一小型超市管理系统。通过对实际客户应用进行需求分析、客户环境分析、界面设计分析等多方面分析，对产品进行多功能开发。

2、系统需求分析及技术介绍

2.1 功能需求分析

1. 存储数据：用户可以使用该系统存储进货信息、售出货物信息、货物库存信息和每日盈利信息。

2. 刷新数据：在客户使用系统完成对数据的更改（如增、删、改）后，系统会自动刷新货物的库存和盈利额。

3. 使用数据：用户通过该系统可以实现对超市数据库的增加、删除、更改、查询操作。

4. 数据安全：必须在登陆界面登录才可进入后台。只有特定权限用户登录后台才可以更改数据。

2.2 可行性分析

2.2.1. 市场前景

中国经济快速发展,人民对生活质量提高存在迫切需求,小型超市以其便利、快捷的优势受到广泛推广。同时由于移动支付的推进，小型超市的发展更进一步。

由于小型超市市场广阔且转向数字化，小型超市商品管理系统的前景非常广阔，可以提升超市的效率，进而提升用户体验。

2.2.2 目标群体

产品目标群体主要面向中小型超市，通过该系统可以实现商品自动化管理、监控库存、销量、盈利等方面，大大提升工作效率，提升企业竞争力。

2.3 系统运行环境

1. 操作系统:Windows 与 Linux 均可运行。
2. 中间件:NGINX1.15.11。
3. 服务端:PHP5.5.9。
4. 数据库:MySQL5.7.26。

开发系统环境为 Windows22621.1702，中间件环境为 NGINX1.15.11，数据库环境为 MySQL5.7.26，PHP 版本为 5.5.9。由于系统支持集成化部署，本文配合软件 Phpstudypro 环境演示。

2.4 相关技术介绍

前端实现:HTML+CSS+Javascript。HTML 实现整体网站元素的布置，CSS 实现整体页面结构及美感优化，Javascript 实现动效设计。

后端实现:PHP，借助 PHP 语言实现数据库的连接与操作。

数据库:MYSQL。MYSQL 是一种关系型数据库管理系统，它可以帮助我们存储和管理数据。与其他的数据库管理系统相比，MYSQL 具有开源、免费、可定制化等优秀特性，并且支持多线程处理、事务控制、安全性高等特点。

3、系统总体设计

3.1 系统功能结构设计

系统功能模块划分如图 3-1 所示。

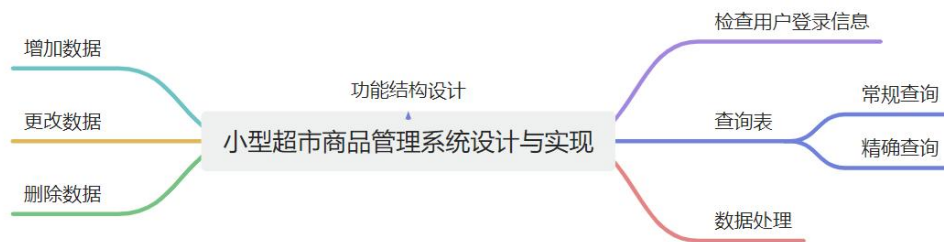


图 3-1 系统功能结构图

1. 检查用户登录信息:登录后查询用户身份,不同身份用户享有不同权限。
2. 查询表:
 - 2.1 常规查询:对指定表存在单独的页面可以做一键查询以及简单的单列查询。
 - 2.2 精确查询:对存在多个指定查询列的情况进行查询。
3. 数据处理:借助后端对数据库中的关联数据进行运算处理,并且对处理后数据进行数据表更新或插入。具体体现为在进货后会自动更新货物库存,在卖出货物后也会自动更新货物库存,并且自动刷新盈利额。在访问主页时也会自动刷新盈利额和库存。
4. 增加数据:管理员从数据库中选择表录入数据。
5. 更改数据:管理员对数据库中表的数据进行更改。
6. 删除数据:管理员删除指定的数据库表的数据。

3.2 系统功能流程设计

3.2.1 检查用户登录功能流程设计



用户从登录界面登录,如果登录成功会给以此用户生成的 cookie。后台的页面都会验证是否存在 cookie 中的 username 参数,存在才会继续访问,不存在则跳转至登录页面。若要进行增删改,这四个页面会检查 cookie 中的 isadmin 参数,如果合法则进行操作,不合法跳转至网站主页。

3.2.2 查询流程设计



图 3-2-2 查询功能流程图

选择进行查询可以选择基础查询和精确查询。基础查询是进入页面自动会查询表内所有信息，也可以输入一个列名和对应的值进行简单的单列查询。精确查询可以选择查询表后，选择指定两列和对应值做查询或指定三列和对应值做查询。

3.2.3 数据处理流程设计



图 3-2-3 数据处理流程图

数据处理进行计算盈利额和计算库存余数。

计算盈利额通过从 goodsinfo 中提取进货价与卖价计算出单品盈利额, 从 sale 表中可以提取出售货量, 因而计算出总盈利额, 再将结果更新至 profitdata 表中。

计算库存余数通过从 stock 中查询指定货物进货所有量, 从 sale 中查询指定货物卖出所有量, 作差得到库存数, 再将结果更新至 goodsinfo 表中。

在进行增、删、改操作时, 操作完成后都会进行数据处理操作。

3.2.4 增、删、改数据流程设计



图 3-2-4 增删改功能流程图

增：从前端获取想要增加数据的表，更具对应的列名写想要插入的值进行插入。

删：从前端获取想要删除数据的表，通过输入想要删除的列名和对应的值完成对该行数据的删除。

改：从前端获取想要更改数据的表，输入想要更改表的列和更改的值以及用于定位到该行数据的列和对应的值。

3.3 数据库设计

3.3.1 概念结构设计

(1) 用户登录信息。

管理系统后台登录需要授权用户才可以访问，所以需要数据表存储授权用户。

如图 3-1-1 所示。用户登录信息主要包括账户 id、用户名、密码。

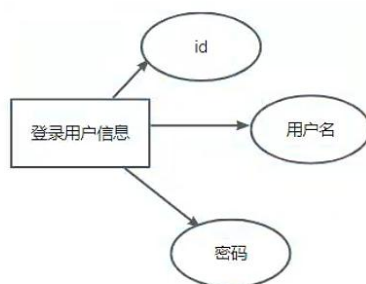


图 3-1-1 登录用户信息属性图

(2) 进货信息

进货信息为商家进货后的更新进货信息的表，用于记录进货数据。记录进货信息可在特定情况对每一条信息进行查询，可以良好保存成本记录。

如图 3-1-2 所示。进货信息主要包括进货事件 id、货物名、进货时间、货物种类、进货量、进货单价。

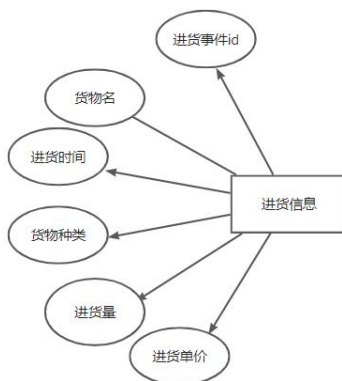


图 3-1-2 进货信息属性图

(3) 售出信息

在商品售出后对商品售出信息进行记录的表。记录售出信息便于核对账单与收银信息。

如图 3-1-3 所示。进货信息主要包括售出事件 id、货物名、售出时间、货物种类、售出量、售出单价。

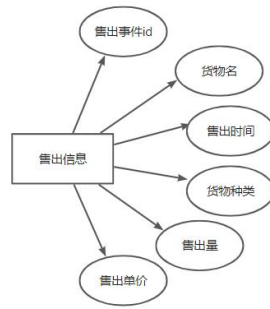


图 3-1-3 售出信息属性图

(4) 货物信息

货物信息主要针对每种超市中售卖的货物。对货物信息进行记录便于分辨每种货物的存
货与其特性。

如图 3-1-4 所示。进货信息主要包括货物 id、货物名、货物种类、货物存量、售出单价、
进货单价。

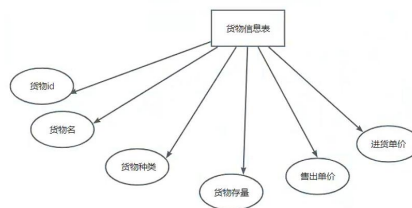


图 3-1-4 货物信息属性图

(5) 盈利信息

盈利信息主要针对超市的每日超市的盈利量。盈利信息是考量超市运营状况的重要内容，
需要建表保存。如图 3-1-5 所示。进货信息主要包括记录日期、当日盈利额。

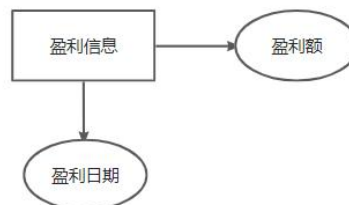


图 3-1-5 盈利信息属性图

(6) 货物信息实体联系

在进货时, 进货 m 条数据则要在货物信息表中对应的 n 条数据进行增加。在售出货物时, 售出

m 件货物后需要对货物信息表的 n 条数据的存货量进行减少操作。货物信息实体联系如图 3-1-5 所示。



图 3-1-5 货物信息 er 图

(7) 盈利信息实体联系

当售出商品后,售出记录内容会增多,因而导致盈利量提高。盈利实体联系如图 3-1-6 所示。



图 3-1-6 盈利信息 er 图

3.3.2 逻辑结构设计

(1) Login 表

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/> 1	id	int(11)			否	无		AUTO_INCREMENT	修改 删除 更多
<input type="checkbox"/> 2	name	text	utf8_unicode_ci		否	无			修改 删除 更多
<input type="checkbox"/> 3	password	text	utf8_unicode_ci		否	无			修改 删除 更多

图 3-3-2-1 login 表结构说明

login 表存放用户登录信息。

id:账号的唯一 id，为 login 表主键。

Name:账号名称。

Password:账号密码。

(2) Stock 表

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/> 1	stockid	int(11)			否	无		AUTO_INCREMENT	修改 删除 更多
<input type="checkbox"/> 2	goodsname	text	utf8_unicode_ci		否	无			修改 删除 更多
<input type="checkbox"/> 3	stocktime	date			否	无			修改 删除 更多
<input type="checkbox"/> 4	type	text	utf8_unicode_ci		否	无			修改 删除 更多
<input type="checkbox"/> 5	stocknumber	int(11)			否	无			修改 删除 更多
<input type="checkbox"/> 6	stockunitprice	double			否	无			修改 删除 更多

图 3-3-2-2 Stock 表结构说明

Stock 表存放进货信息。

Stockid:进货事件 id,为 stock 表主键。

Goodsname:货物名称。

Stocktime:进货日期。

Type:货物种类。

Stocknumber:进货数量。

Stockunitprice:进货价格。

(3) Sale 表

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/> 1	saleid	int(11)			否	无		AUTO_INCREMENT	修改 删除 更多
<input type="checkbox"/> 2	goodsname	text	utf8_unicode_ci		否	无			修改 删除 更多
<input type="checkbox"/> 3	saletime	date			否	无			修改 删除 更多
<input type="checkbox"/> 4	type	text	utf8_unicode_ci		否	无			修改 删除 更多
<input type="checkbox"/> 5	salenumber	int(11)			否	无			修改 删除 更多
<input type="checkbox"/> 6	saleunitprice	double			否	无			修改 删除 更多

图 3-3-2-3 sale 表结构说明

Sale 表存放货物销售信息。

Saleid:销售事件的 id。为 sale 表主键。

Goodsname:货物名称。

Saletime:销售事件。

Type:货物种类。

Salenumber:销售数量。

Saleunitprice:销售价格。

(4) Goodsinfo 表

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/> 1	goodsid	int(11)			否	无		AUTO_INCREMENT	修改 删除 更多
<input type="checkbox"/> 2	goodsname	text	utf8_unicode_ci		否	无			修改 删除 更多
<input type="checkbox"/> 3	type	text	utf8_unicode_ci		否	无			修改 删除 更多
<input type="checkbox"/> 4	inventory	int(11)			否	无			修改 删除 更多
<input type="checkbox"/> 5	stockprice	double			否	无			修改 删除 更多
<input type="checkbox"/> 6	saleprice	double			否	无			修改 删除 更多

图 3-3-2-4 goodsinfo 表结构说明

Goodsinfo 表存放货物信息。

Goodsid:货物 id，为 goodsinfo 表主键。

Goodsname:货物名称。

Type:货物种类。
Inventory:存货量。
Stockprice:进货价格。
Saleprice:销售价格。

(5) Profitdata 表

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/> 1	checkdate	date			否	无			修改 删除 更多
<input type="checkbox"/> 2	profitnumber	double			否	无			修改 删除 更多

图 3-3-2-5 profitdata 表结构说明

Profitdata 表存放盈利信息。
Checkdate:计算盈利日期。
Profitnumber:每日盈利额。

4. 系统详细设计

4.1 登录认证模块详细设计

登录页面由前端向后端借助 POST 方法传输表单, 后端连接数据库后做判断, 如果用户的账户和密码都正确, 则成功登录借助 js 的 `window.location.href` 实现跳转至 `index.php`。同时使用 `setcookie` 给用户保存 cookie。Cookie 有 `username` 和 `isadmin` 两部分组成。`Username` 保存用户名 base64 后的字符串, 用于判定用户是否成功登录。`Isadmin` 保存 0 或者 1, 如果是管理员用户则为 1, 其他为 0, 只有在 `isadmin` 为 1 时才可以进行对数据库的修改操作。

在进行主页登录会进行判定是否存在 `username`, 存在的前提下证明用户成功登录, 存在才可以访问, 否则跳转至登录界面。再检查 `isadmin` 参数, 如果是管理员登录会直接刷新数据 (盈利数据和货物库存量)。

访问查询页面时, 会检查用户 cookie 的 `username` 参数, 如果存在 `username` 参数, 即可查询。

访问增、删、改页面时, 会对 cookie 中的 `username` 先做判定, 存在的前提下证明用户成功登录。再对用户的 `isadmin` 参数进行判断是否为 1, 只有为 1 才可以访问, 如果是 0 则跳转回 `index.html`。

4.2 常规查询模块详细设计

常规查询模块包含两个功能, 第一个是全表查询, 第二个是指定单列查询。全表查询直接从数据库中取出所有内容。单列查询从前端获取两个参数, 一个是 `$select_column` 用于指定列查询数据, 另一个是 `$select_value` 用于指定列的值查询数据。

4.3 精确查询模块详细设计

精确查询可以选择 2 列查询和 3 列查询。先通过 POST 方法获取 `select_table` 参数去指定查询哪个表,再从 `information_schema.columns` 中提取出该表的所有列名。用户选择 2 列查询或 3 列查询,选择 2 列查询提交两列名和对应的值,选择 3 列查询提供 4 列名和对应的值。

4.4 增、删、改模块详细设计

增模块:用`$select_table`传入需要增加数据的表,传回该表的所有列名,使用`$up1,$up2,$up3,$up4,$up5,$up6`六个变量传入需要增加数据的列对应值,`$up1-6`变量按照列顺序填写,如果表列数不足六,多填的列会被自动忽略,不影响增加数据。

删模块:用`$select_table`传入需要增加数据的表,传回该表的所有列名,使用 POST 方法从前端传回两个变量:`$delete1`和`$delete2`。删除语句如下`$sql="delete from table1 where $a = '$b'"`,其中`$a`为`$delete1`,即指定的列,`$b`为`$delete2`,即列对应的值。前端提示用户删除操作需要使用指定主键(id)。

改模块:用`$select_table`传入需要增加数据的表,传回该表的所有列名,使用`$up1,$up2,$up3,$up4`四个变量分别传入要更改数据的列、要更改数据列的值、指定到对应行的列、指定到对应行的列对应的值。语句如下`$sql="update table1 set $up1 = '$up2' where $up3 = '$up4'"`。

在完成增、删、改操作后会自动调用数据处理,刷新数据库数据。

4.5 数据处理模块详细设计

该模块包含了计算盈利额和计算货物库存量的功能。

计算盈利额函数的实现首先通过`$count_date`传入需要计算盈利额的日期。根据该日期去从 `sale` 表中取出当日所有的货物名和对应的货物销售量,对每个货物名去 `goodsinfo` 中查出对应的进货价格和卖出价格,算出单品的盈利额,通过销售量乘单品盈利额算出总盈利额。再通过 `update` 去对 `profitdata` 表进行更改,如果更改失败(不存在数据)选择 `insert` 进行插入。

计算货物库存量的实现是通过从 `goodsinfo` 中取出每一条货物数据,根据其名称在 `stock` 中取出其总进货量,在 `sale` 中取出其总销售量,将两者作差得到最终的库存量。得到库存量后对 `goodsinfo` 表进行更新。

在管理员用户登录到后台主页时,或者完成对数据的增、删、改操作时,会自动调用计算盈利额函数,并且传入近 31 天的日期作为参数。

4.6 UI 界面详细设计

使用了 bootstrap5 的样式模板



图 4-6 UI

整个界面分为三个部分，导航栏，操作选择，数据库操作展示界面

每一种操作只有展示界面不同，核心代码都在展示界面中，包括表单的设计和数据库逻辑设计。

5 系统实现

5.1 登录功能实现

登录界面如图 4-1 所示，管理员用户可以输入账号密码登录。

图 5-1 登录功能实现界面

登录功能核心代码如下：

```
<?php

// 连接数据库
$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "supermarket";

$con = new mysqli($servername,$username,$password,$dbname);

//前端拿到登录表单
$name = $_POST['username'];
$pass = $_POST['password'];

//账户验证

//取出对应用户的密码
$sql = "SELECT * FROM login WHERE name='$name'";
$result = $con->query($sql);
if($result->num_rows>0){
    while($row=$result->fetch_assoc()){
        #判断密码和用户输入的密码是否一致
        if ($row['password'] == $pass){
            echo "成功登录";

            $b="0";
            if($name == "root"){
                $b = "1";
            }

            //创建 cookie 并保存
            //只有已 root 登录 isadmin 才为 1
            $expire = time()+60*60;
            $a = base64_encode($name);
            echo $a;

            setcookie("username",$a,$expire);
            setcookie("isadmin",$b,$expire);

            echo "<script language=\"javascript\"
type=\"text/javascript\"> window.location.href=\"index.php\"; </script>";
        }
    }
    else {
        echo "密码错误!请重新输入";
    }
}
}
```

```
//关闭连接
$con->close();
?>
```

5.2 查询模块实现

5.2.1 基本表查询

如图，可以看到四个选项，分别对四个表的显示与查询。



图 5-2-1-1 基本表选择按钮

以进货记录表为例，如图 5-2-2

hello连接成功

#	stockid	goodsname	stocktime	type	stocknumber	stockunitprice
1	1	bread	2023-05-01	food	50	4.5
2	2	banana	2023-05-01	food	20	5
3	3	footable	2023-05-01	ball	10	50
4	4	basketball	2023-05-01	ball	4	5
5	5	cake	2023-05-01	food	10	5
6	6	bread	2023-06-02	food	20	4.5
7	7	bread	2023-06-05	food	20	4.5

选择要查询的列:

列

选择要查询的值:

值

Submit

#	stockid	goodsname	stocktime	type	stocknumber	stockunitprice
1	1	bread	2023-05-01	food	50	4.5
2	6	bread	2023-06-02	food	20	4.5
3	7	bread	2023-06-05	food	20	4.5

图 5-2-1-2 基本表选择按钮

选择进货记录表并进行查看，上方为全表显示。

可以进行列筛选，查询的列为 goodsname，值为 bread，下方把 goodsname 列名为 bread 的数据显示了出来。

核心代码如下：

全表展示：

```
$sql = "SELECT * FROM stock";
$result = $con->query($sql);
if($result->num_rows>0){

    echo '<table class="table">
    <thead>
        <tr>
            <th scope="col">#</th>
            <th scope="col">stockid</th>
            <th scope="col">goodsname</th>
            <th scope="col">stocktime</th>
            <th scope="col">type</th>
            <th scope="col">stocknumber</th>
            <th scope="col">stockunitprice</th>
        </tr>
    </thead>
    <tbody>';

    function show_all($result){
        $x=1;
        while($row=$result->fetch_assoc()){
            echo '<tr>
            <th scope="row">';
            echo $x.'</th>
            <td>';
            echo $row['stockid'];
            echo '</td>
            <td>';
            echo $row['goodsname'];
            echo '</td>
            <td>';
            echo $row['stocktime'];
            echo '</td>
            <td>';
            echo $row['type'];
            echo '</td>
            <td>';
```



```

        echo $row['stocknumber'];

        echo '</td>';

        echo $row['stockunitprice'];

        echo '</td>';

        </tr>';

        $x+=1;

    }

}

show_all($result);

echo '</tbody></table>';
}

else{
    echo '没有数据';
}

```

筛选展示:

```

<form action="selectstock.php" method="post">

    <div class="mb-3 mt-3">

        <label for="email" class="form-label">选择要查询的列:</label>

        <input type="text" class="form-control" id="email" placeholder="列" name="selectcolumn">

    </div>

    <div class="mb-3 mt-3">

        <label for="email" class="form-label">选择要查询的值:</label>

        <input type="text" class="form-control" id="email" placeholder="值" name="selectvalue">

    </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

<?php

//指定查看内容
//指定查询列
$selectcolumn = $_POST['selectcolumn'];
//指定查询列的值
$selectvalue = $_POST['selectvalue'];
$sql = "SELECT * FROM stock WHERE $selectcolumn='$selectvalue'";
$result1 = $con->query($sql);
if($result1->num_rows>0){
    echo '<table class="table">

    <thead>

    <tr>

```

```
        <th scope="col">#</th>

        <th scope="col">stockid</th>

        <th scope="col">goodsname</th>

        <th scope="col">stocktime</th>

        <th scope="col">type</th>

        <th scope="col">stocknumber</th>

        <th scope="col">stockunitprice</th>

    </tr>
</thead>
<tbody>';

    // <tr>
    //   <th scope="row">2</th>
    //   <td>Jacob</td>
    //   <td>Thornton</td>
    //   <td>@fat</td>
    // </tr>
    // <tr>
    //   <th scope="row">3</th>
    //   <td colspan="2">Larry the Bird</td>
    //   <td>@twitter</td>
    // </tr>

    show_all($result1);

    echo '</tbody></table>';

}

}
```

5.2.2 精确查询

选择要查询的表:

sale

两列查询

条件1的列:

goodsname

条件1的值:

banana

条件2的列:

type

条件2的值:

food

三列查询

条件1的列:

id

条件1的值:

id

条件2的列:

id

条件2的值:

id

条件3的列:

id

条件3的值:

id

Submit

已有数据

saleid	goodsname	saletime	type	salenumber	saleunitprice
2	banana	2023-05-02	Food	20	8

图 5-2-2-1 精确查询

可以通过两重条件筛选或者三重条件筛选来筛查目标，如图 5-2-2-1，查询出符合条件的数据。

核心代码：

```
<?php

$a = base64_decode($_COOKIE['username']);

$url1 = "login.php";

if ($a != NULL){

    //选择表名打印列名
    //选择需要查询数据的表
    $select_table = $_POST['select_table'];
    // 连接数据库
    $servername = "localhost";
    $username = "root";
    $passwd = "123456";
    $dbname = "supermarket";
    $con = new mysqli($servername,$username,$passwd,$dbname);
    if($con->connect_error){
        die("i am die".connect_error);
    }
    else{

    }

    //查询出要查询数据表的列名
    $sql = "SELECT * FROM information_schema.columns WHERE table_schema='supermarket' and
table_name='$select_table'";

    $result = $con->query($sql);
    if($result->num_rows>0){
        echo '<table class="table">
        <thead>
        <tr>';
        while($row=$result->fetch_assoc()){
            echo '<th scope="col">';
            echo $row['COLUMN_NAME'];
            echo '</th>';
        }
        echo'
        </tr>
        </thead>
        <tbody>';
```

```

    }

    //指定两列查询

    $select21 = $_POST['select21'];
    $select22 = $_POST['select22'];
    $select23 = $_POST['select23'];
    $select24 = $_POST['select24'];

    //查询

    $sql = "SELECT * FROM $select_table WHERE $select21='$select22' and $select23='$select24'";
    $result = $con->query($sql);
    if($result->num_rows>0){
        while($row=$result->fetch_array()){
            echo '<tr>
            <th scope="row">';
            echo $row[0];
            echo '</td>
            <td>';
            echo $row[1];
            echo '</td>
            <td>';
            echo $row[2];
            echo '</td>
            <td>';
            echo $row[3];
            echo '</td>
            <td>';
            echo $row[4];
            echo '</td>
            <td>';
            echo $row[5];
            echo '</td>
            <td>';
            echo $row[6];
            echo '</td>
            </tr>';
        }
    }

    //指定三列查询
    //最多指定三列查询，此信息需表示于前端

    $select31 = $_POST['select31'];
    $select32 = $_POST['select32'];
    $select33 = $_POST['select33'];
    $select34 = $_POST['select34'];
    $select35 = $_POST['select35'];
    $select36 = $_POST['select36'];

    //查询

```

```

        $sql = "SELECT * FROM $select_table WHERE $select31='$select32' and $select33='$select34' and
$select35='$select36'";

        $result = $con->query($sql);

        if($result->num_rows>0){

            while($row=$result->fetch_array()){

                echo '<tr>

                <th scope="row">';

                echo $row[0];

                echo '</td>

                <td>';

                echo $row[1];

                echo '</td>

                <td>';

                echo $row[2];

                echo '</td>

                <td>';

                echo $row[3];

                echo '</td>

                <td>';

                echo $row[4];

                echo '</td>

                <td>';

                echo $row[5];

                echo '</td>

                <td>';

                echo $row[6];

                echo '</td>

                </tr>';

            }

        }

        else{

        }

        $con->close();

    }

    else{

        header("Location:$url1");

    }

?>

```

5.3 修改模块实现

5.3.1 增加数据

选择要插入的表:

sale

列1:

列2:

列3:

列4:

列5:

列6:

Submit

saleid	goodsname	saletime	type	salenumber	saleunitprice
2	banana	2023-05-02	food	20	8
1	bread	2023-05-02	food	40	6
3	footable	2023-05-02	ball	4	95
4	basketball	2023-05-02	ball	4	10
5	cake	2023-05-02	food	10	10
7	bread	2023-06-02	food	20	6
6	bread	2023-06-09	food	10	6

图 5-3-1-1 增加数据 1

如图 5-3-1-1，管理员输入表名后可以点击 submit，可以直接看到表格显示在下方。

选择要插入的表:

sale

列1:

8

列2:

bread

列3:

2023-06-02

列4:

food

列5:

10

列6:

6

Submit

saleid	goodsname	saletime	type	salenumber	saleunitprice
2	banana	2023-05-02	food	20	8
1	bread	2023-05-02	food	40	6
3	footable	2023-05-02	ball	4	95
4	basketball	2023-05-02	ball	4	10
5	cake	2023-05-02	food	10	10
7	bread	2023-06-02	food	20	6
6	bread	2023-06-09	food	10	6
8	bread	2023-06-02	food	10	6

图 5-3-1-1 增加数据 2

如图 5-3-1-1，成功增加了一条序列为 8 的记录。

核心代码：

```
<?php
//判断是否是 root,不是跳转到 index.php
$a = base64_decode($_COOKIE['username']);
$url1 = "login.php";
if($a != NULL){
    $b = $_COOKIE['isadmin'];
    //php header 重定向
```

```

$url = "index.php";

if ($b!='1'){
    header("Location:$url");
}else{

    //选择需要插入数据的数据库
    $select_table = $_POST['select_table'];
    // 连接数据库
    $servername = "localhost";
    $username = "root";
    $passwd = "123456";
    $dbname = "supermarket";
    $con = new mysqli($servername,$username,$passwd,$dbname);
    if($con->connect_error){
        die("i am die".connect_error);
    }
    else{

```

```

    }
    //查询出要插入表的列名
    $sql = "SELECT * FROM information_schema.columns WHERE table_schema='supermarket' and
table_name='$select_table'";
    $result = $con->query($sql);
    if($result->num_rows>0){
        echo '<table class="table">

        <thead>

        <tr>';
        while($row=$result->fetch_assoc()){
            echo '<th scope="col">';
            echo $row['COLUMN_NAME'];
            echo '</th>';
        }
        echo '

        </tr>

        </thead>

        <tbody>';
    }
    $sql = "SELECT * FROM $select_table ";
    $result = $con->query($sql);
    if($result->num_rows>0){
        while($row=$result->fetch_array()){
            echo '<tr>

            <th scope="row">';
            echo $row[0];
            echo '</td>

```

```

        <td>';
        echo $row[1];
        echo '</td>';
        <td>';
        echo $row[2];
        echo '</td>';
        <td>';
        echo $row[3];
        echo '</td>';
        <td>';
        echo $row[4];
        echo '</td>';
        <td>';
        echo $row[5];
        echo '</td>';
        <td>';
        echo $row[6];
        echo '</td>';
    </tr>';
}
}

//前端设置6个格子
$up1 = $_POST['up1'];
$up2 = $_POST['up2'];
$up3 = $_POST['up3'];
$up4 = $_POST['up4'];
$up5 = $_POST['up5'];
$up6 = $_POST['up6'];

//插入语句
if ($select_table=='sale'){
    $sql = "INSERT INTO sale(saleid,goodsname,saletime,type,salenummer,saleunitprice)
VALUES('$up1','$up2','$up3','$up4','$up5','$up6')";
}elseif($select_table=='stock'){
    $sql = "INSERT INTO stock(stockid,goodsname,stocktime,type,stocknumber,stockunitprice)
VALUES('$up1','$up2','$up3','$up4','$up5','$up6')";
}elseif($select_table=='goodsinfo'){
    $sql = "INSERT INTO goodsinfo(goodsid,goodsname,type,inventory,stockprice,saleprice)
VALUES('$up1','$up2','$up3','$up4','$up5','$up6')";
}elseif($select_table=='profitdata'){
    $sql = "INSERT INTO profitdata(checkdate,profitnumber) VALUES('$up1','$up2')";
}

if($con->query($sql)===True){

    include "freshdata.php";

}

```



```

else{

}

}

}
?>

```

5.3.2 删除数据

如图 5-3-2-1，成功删除 saleid 为 8 的记录。

选择要操作的表:

sale

列:

saleid

值:

8

Submit

saleid	goodsname	saletime	type	salenumber	saleunitprice
2	banana	2023-05-02	Food	20	8
1	bread	2023-05-02	food	40	6
3	footable	2023-05-02	ball	4	95
4	basketball	2023-05-02	bell	4	10
5	cake	2023-05-02	food	10	10
7	bread	2023-06-02	food	20	6
6	bread	2023-06-09	food	10	6

图 5-3-2-1 删除数据

删除数据功能核心代码如下：

```

<?php
//判断是否是 root,不是跳转到 index.php
$a = base64_decode($_COOKIE['username']);
$url1 = "login.php";
if($a != NULL){
    $b = $_COOKIE['isadmin'];
    //php header 重定向
    $url = "index.php";
    if ($b!='1'){
        header("Location:$url");
    }else{
        //选择需要删除数据的表

```

```

$select_table = $_POST['select_table'];

// 连接数据库

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "supermarket";

$con = new mysqli($servername,$username,$password,$dbname);
if($con->connect_error){
    die("i am die".connect_error);
}
else{

}

//查询出要删除数据表的列名

$sql = "SELECT * FROM information_schema.columns WHERE table_schema='supermarket' and
table_name='$select_table'";

$result = $con->query($sql);
if($result->num_rows>0){
    echo '<table class="table">

    <thead>

    <tr>';
    while($row=$result->fetch_assoc()){
        echo '<th scope="col">';
        echo $row['COLUMN_NAME'];
        echo '</th>';
    }
    echo '
    </tr>
</thead>
<tbody>';
}

$sql = "SELECT * FROM $select_table ";
$result = $con->query($sql);
if($result->num_rows>0){
    while($row=$result->fetch_array()){
        echo '<tr>

        <th scope="row">';
        echo $row[0];
        echo '</td>

        <td>';
        echo $row[1];
        echo '</td>

        <td>';
        echo $row[2];
    }
}

```

```

        echo '</td>';
        echo $row[3];
        echo '</td>';
        echo $row[4];
        echo '</td>';
        echo $row[5];
        echo '</td>';
        echo $row[6];
        echo '</td>';
    </tr>';
}

}

//前端设置2个格子
//指定删除列名
$delete1 = $_POST['delete1'];
//指定列对应值
$delete2 = $_POST['delete2'];
//删除语句
if(isset($delete1)){
    if ($select_table=='sale'){
        $sql1 = "DELETE FROM sale WHERE $delete1 = '$delete2'";
    }elseif($select_table=='stock'){
        $sql1 = "DELETE FROM stock WHERE $delete1 = '$delete2'";
    }elseif($select_table=='goodsinfo'){
        $sql1 = "DELETE FROM goodsinfo WHERE $delete1 = '$delete2'";
    }elseif($select_table=='profitdata'){
        $sql1 = "DELETE FROM profitdata WHERE $delete1 = '$delete2'";
    }elseif($select_table=='workerinfo'){
        $sql1 = "DELETE FROM workerinfo WHERE $delete1 = '$delete2'";
    }
}

if(isset($sql1)){
    if($con->query($sql1)===True){
        include "freshdata.php";
    }
    else{
    }
}

}
}
}

```

```
? >
```

5.3.3 更改数据

如图 5-3-3-1-1，5-3-3-1-2，
将 saleid 为 7 这列的 salenumber 修改为 10。

选择要修改的表:

sale

列:

值:

列名:

值:

Submit

saleid	goodsname	saletime	type	salenumber	saleunitprice
2	banana	2023-05-02	food	20	8
1	bread	2023-05-02	food	40	6
3	footable	2023-05-02	ball	4	95
4	basketball	2023-05-02	ball	4	10
5	cake	2023-05-02	food	10	10
7	bread	2023-06-02	food	20	6
6	bread	2023-06-09	food	10	6

图 5-3-3-1-1 更改数据 1

选择要修改的表:

sale

列:

salenumber

值:

10

列名:

saleid

值:

7

Submit

saleid	goodsname	saletime	type	salenumber	saleunitprice
2	banana	2023-05-02	food	20	8
1	bread	2023-05-02	food	40	6
3	footable	2023-05-02	ball	4	95
4	basketball	2023-05-02	ball	4	10
5	cake	2023-05-02	food	10	10
7	bread	2023-06-02	food	10	6
6	bread	2023-06-09	food	10	6

图 5-3-3-1-2 更改数据 2

更改数据功能核心代码如下：

```
<?php
//判断是否是 root,不是跳转到 index.php
$a = base64_decode($_COOKIE['username']);
$url1 = "login.php";
if ($a != NULL){
    $b = $_COOKIE['isadmin'];
    //php header 重定向
    $url = "index.php";
    if ($b!='1'){
        header("Location:$url");
    }else{

        //选择需要更改数据的数据库
        $select_table = $_POST['select_table'];
        // 连接数据库
        $servername = "localhost";
        $username = "root";
        $passwd = "123456";
        $dbname = "supermarket";
        $con = new mysqli($servername,$username,$passwd,$dbname);
        if($con->connect_error){
            die("i am die".connect_error);
        }
        else{

        }

        //查询出要更改表的列名
        $sql = "SELECT * FROM information_schema.columns WHERE table_schema='supermarket' and
table_name='$select_table'";
        $result = $con->query($sql);
        if($result->num_rows>0){
            echo '<table class="table">
            <thead>
            <tr>';
            while($row=$result->fetch_assoc()){
                echo '<th scope="col">';
                echo $row['COLUMN_NAME'];
```

```

        echo '</th>';
    }

    echo'

    </tr>

</thead>
<tbody>';

    }

    $sql = "SELECT * FROM $select_table ";
    $result = $con->query($sql);
    if($result->num_rows>0){
        while($row=$result->fetch_array()){
            echo '<tr>

            <th scope="row">';
            echo $row[0];
            echo '</td>

            <td>';
            echo $row[1];
            echo '</td>

            <td>';
            echo $row[2];
            echo '</td>

            <td>';
            echo $row[3];
            echo '</td>

            <td>';
            echo $row[4];
            echo '</td>

            <td>';
            echo $row[5];
            echo '</td>

            <td>';
            echo $row[6];
            echo '</td>

            </tr>';

        }
    }

    else{
        echo '没有数据';
    }

    //前端设置 4 个格子
    //要更改的列名
    $up1 = $_POST['up1'];
    //更改的值
    $up2 = $_POST['up2'];
    //索引列名

```

```

$up3 = $_POST['up3'];

//索引值

$up4 = $_POST['up4'];

//更改语句

if ($select_table=='sale'){

    $sql = "UPDATE sale SET $up1 = '$up2' WHERE $up3 = '$up4'";

}elseif($select_table=='stock'){

    $sql = "UPDATE stock SET $up1 = '$up2' WHERE $up3 = '$up4'";

}elseif($select_table=='goodsinfo'){

    $sql = "UPDATE goodsinfo SET $up1 = '$up2' WHERE $up3 = '$up4'";

}elseif($select_table=='profitdata'){

    $sql = "UPDATE profitdata SET $up1 = '$up2' WHERE $up3 = '$up4'";

}elseif($select_table=='workerinfo'){

    $sql = "UPDATE workerinfo SET $up1 = '$up2' WHERE $up3 = '$up4'";

}

if($con->query($sql)===True){

}

else{

}

$con->close();

}

?>

```

5.3.4 数据处理功能

数据刷新功能没有页面，在进行增删改后，会自动执行，下面演示两大功能，一是存量自动刷新，二是利润自动刷新。

5.3.4.1 存量自动刷新

#	goodsid	goodsname	type	inventory	stockprice	saleprice
1	1	bread	food	30	4.5	6
2	2	banana	food	0	5	8
3	3	footable	ball	6	50	95
4	4	basketball	ball	0	5	10
5	5	cake	food	0	5	10
6	6	apple	food	0	3	5

图 5-3-4-1-1 库存刷新 1

选择要插入的表:

stock

列1:

8

列2:

bread

列3:

2023-06-02

列4:

food

列5:

10

列6:

4.5

图 5-3-4-1-2 库存刷新 2

#	goodsid	goodsname	type	inventory	stockprice	saleprice
1	1	bread	food	40	4.5	6
2	2	banana	food	0	5	8
3	3	footable	ball	6	50	95
4	4	basketball	ball	0	5	10
5	5	cake	food	0	5	10
6	6	apple	food	0	3	5

图 5-3-4-1-3 库存刷新 3

如图 5-3-4-1-1，bread 库存为 30，在图 5-3-4-1-2 中增加进货记录后，如图 5-3-4-1-3，库存中的 bread 变为 40。

存量自动刷新功能核心代码如下：

```
// 判断货物的余量
// 判断当前时间之前所有的货物的存货，计算后存放于 goodsinfo 中
// 根据 goodsname 可以取出指定货物的进货量以及售出量
// 需要提前对货物的种类进行录入
//提取所有 goodsinfo 中的信息
$sql = "SELECT * FROM goodsinfo";
$result = $con->query($sql);
if($result->num_rows>0){
    while($row=$result->fetch_assoc()){
        //此处是遍历 goodsinfo 中的每一行元素，对每一个 goodsname 到 sale 和 stock 表中查询数量即可
        //定义变量存储进货货物的数量值
        $t = 0;

        //进货
        $m = $row['goodsname'];
```



```

$sql2 = "SELECT * FROM stock WHERE goodsname='$m'";
$result2 = $con->query($sql2);
if($result2->num_rows>0){
    while($row2=$result2->fetch_assoc()){
        $t = $t + $row2['stocknumber'];
    }
}

//售出
$n = $row['goodsname'];
$sql1 = "SELECT * FROM sale WHERE goodsname='$n'";
$result1 = $con->query($sql1);
if($result1->num_rows>0){
    while($row1=$result1->fetch_assoc()){
        $t = $t - $row1['salenumber'];
    }
}
}

```

```

//更新 goodsinfo 表
$sql = "UPDATE goodsinfo SET inventory = '$t' WHERE goodsname = '$m'";
if($con->query($sql)===True){
    if($orderid!=0){
        }
    else{
        }
    }
    else{
    }
}
}
$con->close();
}
}else{
    echo "<script language=\"javascript\"
type=\"text/javascript\"> window.location.href=\"login.html\"; </script>";
}

```

5.3.4.2 利润自动刷新

#	checkdate	profitnumber
1	2023-05-02	370
2	2023-06-09	15
3	2023-06-08	0
4	2023-06-07	0
5	2023-06-06	0
6	2023-06-05	0
7	2023-06-04	0
8	2023-06-03	0
9	2023-06-02	15

图 5-3-4-2-1 利润刷新

选择要插入的表:

sales

列1:

8

列2:

bread

列3:

2023-05-02

列4:

food

列5:

20

列6:

6

图 5-3-4-2-2 利润刷新 2

#	checkdate	profitnumber
1	2023-05-02	400
2	2023-06-09	15
3	2023-06-08	0
4	2023-06-07	0
5	2023-06-06	0
6	2023-06-05	0
7	2023-06-04	0
8	2023-06-03	0

图 5-3-4-2-3 利润刷新 3

如图 5-3-4-2-1，2023-05-02 利润为 370，在图 5-3-4-2-2 中增加销售记录后，如图 5-3-4-2-3，销售利润表中的利润额变为 400。

利润自动刷新功能核心代码如下：

```
//连接数据库

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "supermarket";

$con = new mysqli($servername,$username,$password,$dbname);

//计算盈利
//计算盈利并且插入到
//输入需要刷新盈利数据的日期
function function_profit($orderdate){
    //函数中单独连接数据库
    $servername = "localhost";
    $username = "root";
    $password = "123456";
    $dbname = "supermarket";
    $con = new mysqli($servername,$username,$password,$dbname);
    $count_date = $orderdate;
    if(isset($count_date)){
        //卖出求和
        $sum = 0;

        $sql = "SELECT * FROM sale WHERE saletime='$count_date'";
        $result = $con->query($sql);
        if($result->num_rows>0){
            while($row=$result->fetch_assoc()){
                $a = $row['goodsname'];
                //在 goodsinfo 表中取成本值
                $sql1 = "SELECT * FROM goodsinfo WHERE goodsname='$a'";
                $result1 = $con->query($sql1);
                if($result1->num_rows>0){
                    while($row1=$result1->fetch_assoc()){
                        //卖出单品的盈利额
                        $y1 = $row1['saleprice']-$row1['stockprice'];
                    }
                }
                //当日总盈利额
                $sum = $sum + $row['salenumber']*$y1;
            }
        }
    }
}
```

```

    }

    //如果是主页调用,刷新成功后无输出结果
    if($orderdate!=0){

    }

    else{

        //如果是 fresh 页面则进行输出

        echo $count_date.'盈利额为'.$sum;

    }

    //从 profitnumber 根据指定日期查询
    $sql = "SELECT * FROM profitdata WHERE checkdate='$count_date'";
    $result = $con->query($sql);
    if($result->num_rows>0){

        //有数据就更改数据

        $sql = "UPDATE profitdata SET profitnumber = '$sum' WHERE checkdate ='$count_date'";
        if($con->query($sql)===True){

            if($orderdate!=0){

            }

            else{

                //如果是 fresh 页面则进行输出

                echo '更改成功'.'<br />';

            }

        }

        else{

            echo '执行失败';

        }

    }

    //没有数据就插入数据
    else{

        $sql = "INSERT INTO profitdata(checkdate,profitnumber) VALUES('$count_date','$sum')";
        if($con->query($sql)===True){

            if($orderdate!=0){

            }

            else{

                //如果是 fresh 页面则进行输出

                echo '插入成功'.'<br />';

            }

        }

        else{

            echo '执行失败';

        }

    }

}

```

```
    }  
}  
  
echo "已自动刷新数据".'\<br />';  
$a = date("Y-m-d",time());  
$i = 0;  
  
//循环获取前 31 天的日期  
while ($i <= 30){  
    $t = '-'.$i.' day';  
    $b = date('Y-m-d', strtotime($t));  
    $i += 1;  
    function_profit($b);  
}
```

6 总结

课程设计包括需求分析，数据库框架设计，逻辑实现等阶段。

目前系统功能有自动刷新库存，计算利润，未来可以添置更多功能，例如模糊搜索，多表查询等。