



長安大學

# 数据库原理 课程设计报告

设计题目: 鸭鸭便利店管理系统

---

院 系: 信息工程学院

---

组 长: 刘文越

---

组 员: 王子卓、李欣蓉、许睿

---

指导教师: 李晓莉

---

日 期: 2023.06

---

## 1. 设计内容与要求

本课程设计旨在设计一个便利店管理系统，方便不同类型的用户通过该系统管理便利店的日常业务数据，如商品信息，供应商信息，出售信息等。

1、熟悉数据库产品的设计思路，按照规范的数据库设计路线进行设计开发，构造优化的数据库逻辑模式和物理结构，使之有效地存储和管理数据，满足各种用户的应用需求。

2、进一步掌握 sql 语句，通过 sql 对数据库相关数据的进行访问及查询，并进行 C 语言的嵌入式编程，最终结合 Qt 设计应用前端对数据库进行操作。

## 2. 实验环境

1) 硬件环境：

笔记本电脑

- 处理器：AMD Ryzen 9 7945HX with Radeon Graphics(2.50 GHz)
- 主 板：微星 MS-17K2
- 内 存：32 GB
- 显 卡：Nvidia GeForce RTX 4080 Laptop GPU

2) 软件环境：

- Windows 11 系统
- Visual Studio 2022 (with Qt Visual Studio Tools 插件)
- Qt Creator (with Qt designer)
- Wampserver64(sqlserver8.0.31)

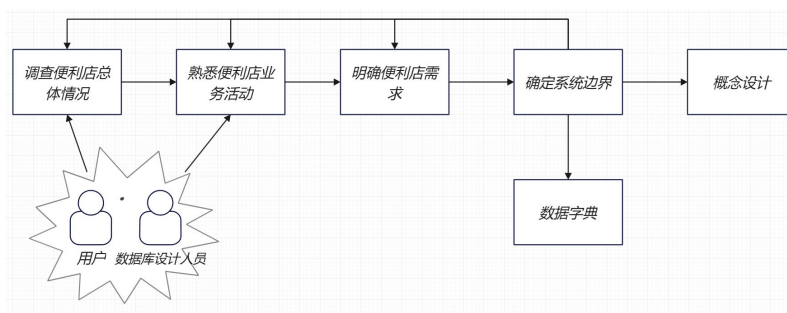
## 3. 小组分工

- 刘文越（组长）：数据库概念结构、逻辑结构设计、ODBC 类的封装设计（应用程序连接数据库）、Qt UI 总界面设计、撰写报告
- 王子卓（组员）：物理结构设计、后台数据载入、Qt UI 异常数据弹窗设计
- 李欣蓉（组员）：数据备份和恢复、Qt UI 登录弹窗界面设计
- 许 睿（组员）：Qt UI 收支界面设计、撰写报告

## 4. 需求分析

课程设计主体为一个便利店管理系统，通过准确了解与分析用户需求，考虑实际情况，调查便利店的业务活动情况，确定管理系统的边界，即确定哪些功能由计算机完成，哪些活动由人工完成。

图 4.1 需求分析流程图



对于店主：

- 管理员工信息，包含姓名、性别、薪资等信息。
- 管理登录账号，包含不同管理层的级别信息、账号和密码。

对于员工：

- 管理商品信息，包含商品的名字、类型、定价、折扣等信息。
- 管理出售记录，包含出售时间、出售商品、营业额等信息。
- 管理进货记录，包含进货时间、进货商品、进货成本等信息。

对于商品：

- 需要包含库存信息，折扣信息等。
- 需要查询供应商信息，包括供应商姓名、地址、联系方式等信息。

对于业务活动：

- 需要月份或年份的收支信息
- 需要对商品进行折扣定价的修改

通过需求分析建立起数据字典，对数据库进行数据描述，并在数据库设计过程中不断修改、充实、完善。

图 4.2 duck\_store 数据库数据字典导出图

表名	列名	类型	默认值	允许为空	数据类型	字符最大长度	数字精度	小数位数
inventory	id	int	NULL	NO	int	NULL	10	0
inventory	name	char(20)	NULL	NO	char	20	NULL	NULL
inventory	number	int	NULL	NO	int	NULL	10	0
inventory	type	char(10)	NULL	NO	char	10	NULL	NULL
merchandise	Discount	double	NULL	NO	double	NULL	22	NULL
merchandise	ID	int	NULL	NO	int	NULL	10	0
merchandise	Name	char(20)	NULL	NO	char	20	NULL	NULL
merchandise	Number	int	NULL	NO	int	NULL	10	0
merchandise	Price	double	NULL	NO	double	NULL	22	NULL
merchandise	Type	char(10)	NULL	NO	char	10	NULL	NULL
purchase	Date	date	NULL	NO	date	NULL	NULL	NULL
purchase	ID	int	NULL	NO	int	NULL	10	0
purchase	Merchandise_ID	int	NULL	NO	int	NULL	10	0
purchase	Number	int	NULL	NO	int	NULL	10	0
purchase	Price	double	NULL	NO	double	NULL	22	NULL
purchase	Supplier_ID	int	NULL	NO	int	NULL	10	0
sell	Date	date	NULL	NO	date	NULL	NULL	NULL
sell	ID	int	NULL	NO	int	NULL	10	0
sell	Merchandise_ID	int	NULL	NO	int	NULL	10	0
sell	Number	int	NULL	NO	int	NULL	10	0
sell	Price	double	NULL	NO	double	NULL	22	NULL
sell	Staff_ID	int	NULL	NO	int	NULL	10	0
staff	Age	int	NULL	NO	int	NULL	10	0
staff	ID	int	NULL	NO	int	NULL	10	0
staff	Information	char(20)	NULL	NO	char	20	NULL	NULL
staff	Name	char(10)	NULL	NO	char	10	NULL	NULL
staff	Salary	double	NULL	NO	double	NULL	22	NULL
staff	Sex	char(2)	NULL	NO	char	2	NULL	NULL
supplier	Address	char(20)	NULL	NO	char	20	NULL	NULL
supplier	ID	int	NULL	NO	int	NULL	10	0
supplier	Information	char(20)	NULL	NO	char	20	NULL	NULL
supplier	Name	char(20)	NULL	NO	char	20	NULL	NULL
user	ID	int	NULL	NO	int	NULL	10	0
user	Name	char(10)	NULL	NO	char	10	NULL	NULL
user	PWD	char(16)	NULL	NO	char	16	NULL	NULL
user	Role	int	NULL	NO	int	NULL	10	0

## 5. 概念结构设计

概念结构设计需要能够真实、充分地反映现实世界，易于理解，易于更改，

易于向数据模型进行转化。本管理系统的概念结构设计采用 E-R 模型来进行描述，见下图。

图 5.1 鸭鸭便利店商品系统 E-R 图

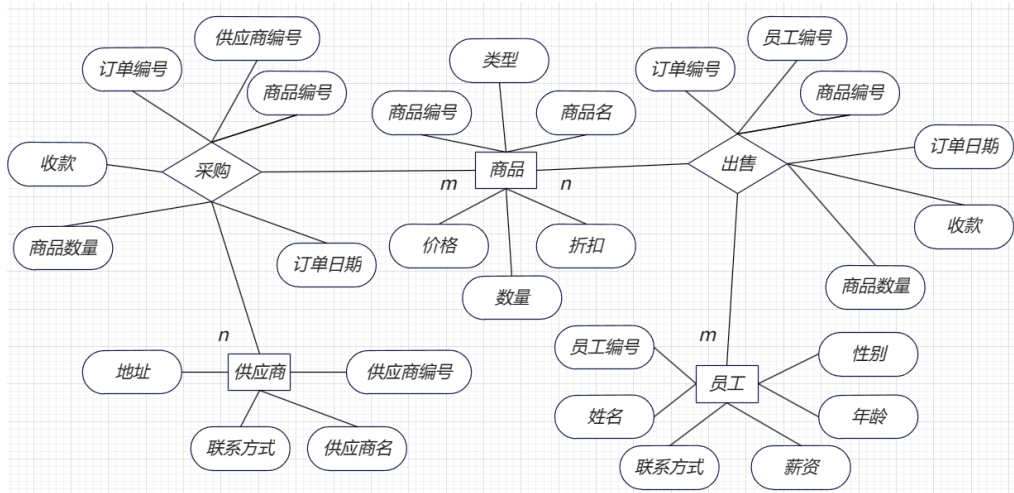


图 5.2 鸭鸭便利店员工 E-R 图

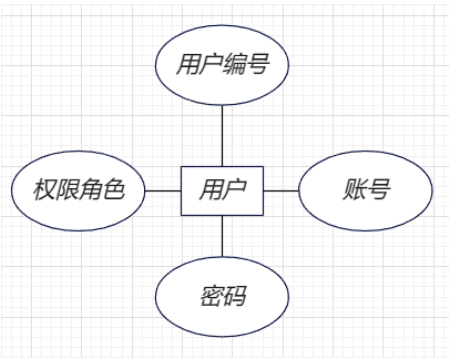
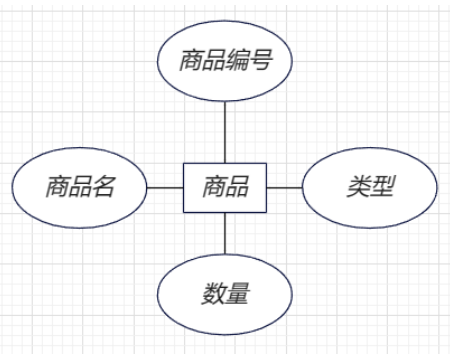


图 5.3 鸭鸭便利店库存 E-R 图



## 6. 逻辑结构设计

将概念结构设计阶段的 E-R 图转换为关系数据库中的关系模式，以实现数据库设计中的逻辑结构设计。

本管理系统采用 mysql 关系数据库，关系模式如下：

员工（员工编号，姓名，性别，年龄，联系方式，薪资）

主码：员工编号

商品（商品编号，商品名，类型，价格，数量，折扣）

主码：商品编号

供应商（供应商编号，供应商名，联系方式，地址）

主码：供应商编号

出售（订单编号，员工编号，商品编号，订单日期，收款，商品数量）

主码：订单编号 外码：员工编号，商品编号

采购（订单编号，商品编号，供应商编号，订单日期，收款，商品数量）

主码：订单编号 外码：员工编号，商品编号

在确定关系模式的基础上，我们定义了库存视图，建立在商品关系模式上，选取商品编号、商品名、类型、数量四个特征。

图 6.1 鸭鸭便利店库存视图（左侧为商品表，右侧为库存视图）

ID	Type	Name	Price	Number	Discount	id	type	name	number
1	食品类	四个圈	3.5	150	1	1	食品类	四个圈	150
2	饮品类	农夫山泉	2	182	1	2	饮品类	农夫山泉	182
3	饮品类	魔爪	6.5	105	1	3	饮品类	魔爪	105
4	饮品类	脉动	4.8	110	0.9	4	饮品类	脉动	110
5	饮品类	可乐	3	105	1	5	饮品类	可乐	105
6	饮品类	雪碧	3	105	1	6	饮品类	雪碧	105
7	饮品类	卡布奇诺	8	55	0.9	7	饮品类	卡布奇诺	55
8	饮品类	牛奶	3.5	120	1	8	饮品类	牛奶	120
9	饮品类	拿铁	8	65	0.9	9	饮品类	拿铁	65
10	饮品类	咖啡	6	65	0.9	10	饮品类	咖啡	65

依据现实中便利店的业务，店员进行进货和销售时，商品的数量应该随之变化，管理系统中采用触发器来维护商品的数量和商品折扣时的营业额。

图 6.2 鸭鸭便利店触发器列表

名字	表	时间	事件
tri_purchase_delete	purchase	BEFORE	DELETE
tri_purchase_insert	purchase	AFTER	INSERT
tri_sell_delete	sell	BEFORE	DELETE
tri_sell_discount	sell	BEFORE	INSERT
tri_sell_insert	sell	AFTER	INSERT

编辑

触发器名称

tri\_sell\_insert

表

sell

时间

AFTER

事件

INSERT

定义

```
1 BEGIN
2   UPDATE `merchandise`
3   SET `merchandise`.`Number` = `merchandise`.`Number` - NEW.`Number`
4   WHERE `merchandise`.`ID` = NEW.`Merchandise_ID`;
5 END
```

用户

root@localhost

这里以 tri\_sell\_insert 触发器内部为例：

```
BEGIN
    UPDATE `merchandise`
    SET `merchandise`.`Number` = `merchandise`.`Number` -
NEW.`Number`
    WHERE `merchandise`.`ID` = NEW.`Merchandise_ID`;
END
```

在对 sell 表进行插入操作时，在操作执行之后，会使 merchandise 表中对应的商品数量减去卖出的商品数量，以此来维护商品的数量符合现实情况。

## 7. 物理结构设计

### 7.1 mysql 数据库物理存储位置

通过查询本地 mysql 服务器的 my.ini 文件，查询到数据库的存储地址为 "D:/develop/wamp64/bin/mysql/mysql8.0.31/data"

### 7.2 数据库大小

通过分析方式估算数据库大小为：

#### (1) 员工表大小

目前表内有 5 名员工，根据店内规模预计未来会发展为 7 名员工，员工表内记录了员工的员工编号，姓名，性别，年龄，联系方式，薪资信息。员工编号为 int 类型，每个占 4 字节，姓名为 char(10)类型，每个占 10 字节，性别为 char(2)类型，每个占 2 字节，年龄为 int 类型，每个占 4 字节，薪资为 double 类型，每个占 8 字节，联系方式为 char(20)类型，每个占 20 字节。故每个员工的信息占  $4+10+2+4+8+20=48$  字节。目前有五位员工，故占  $48*5=240$  字节。未来发展为 7 名员工时，占  $48*7=336$  字节。

#### (2) 商品表大小

目前商品表内有 46 种商品，根据店内规模未来可能会发展为 100 种商品，商品表内记录了商品的商品编号，商品名，类型，价格，折扣，数量信息。商品编号为 int 类型，每个占 4 字节，商品名为 char(20)类型，每个占 20 字节，类型为 char(10)类型，每个占 10 字节，价格为 double 类型，每个占 8 字节，折扣为 double 类型，每个占 8 字节，数量为 int 类型，每个占 4 字节。故每个商品的信息占  $4+20+10+8+8+4=54$  字节。目前有 46 种商品，占  $46*54=2484$  字节。未来发展为 100 种商品，占  $46*100=4600$  字节。

#### (3) 供应商表大小

目前供应商表内有 5 个供应商，根据店内规模预计未来会发展为 10 个供应商，供应商表内记录了供应商的供应商编号，供应商名，联系方式，地址。供应商编号为 int 类型，每个占 4 字节，供应商名为 char(20)类型，每个占 20 字节，联系方式为 char(20)类型，每个占 20 字节，地址为 char(20)类型，每个占 20 字节。故每个供应商信息占  $4+20+20+20=64$  字节，目前有 5 个供应商，占  $5*64=320$  字节。未来发展为 10 个供应商，占  $10*64=640$  字节。

#### (4) 出售表大小

目前出售表内有 80 条出售数据，根据店内规模预计未来每月会发展为 200 条出售数据，出售表内记录了出售信息的订单编号，员工编号，商品编号，订单日期，收款，商品数量。订单编号为 int 类型，每个占 4 字节，员工编号为 int 类型，每个占 4 字节，商品编号为 int 类型，每个占 4 字节，订单日期为 date 类型，每个占 3 字节，收款为 double 类型，每个占 8 字节，商品数量为 int 类型，每个占 4 字节。故每条出售信息占  $4+4+4+3+4+4=23$  字节，目前有 80 条出售数据，占  $23*80=1840$  字节。未来发展为 200 条出售数据，占  $200*23=4600$  字节。

#### (5) 采购表大小

目前采购表内 6 月有 28 条采购数据，根据店内规模预计未来每月会发展为 60 条采购数据，采购表内记录了采购信息的订单编号，商品编号，供应商编号，

订单日期，收款，商品数量。订单编号为 int 类型，每个占 4 字节，供应商编号为 int 类型，每个占 4 字节，商品编号为 int 类型，每个占 4 字节，订单日期为 date 类型，每个占 3 字节，收款为 double 类型，每个占 8 字节，商品数量为 int 类型，每个占 4 字节。故每条采购信息占  $4+4+4+3+4+4=23$  字节，目前有 28 条采购数据，占  $23*28=644$  字节。未来发展为 50 条采购数据，占  $50*23=1150$  字节。

当前情况数据库大小为 5528 字节，估算未来发展为 11326 字节  
通过代码方式查询数据库大小：

图 7.1 代码查询数据库大小

```
mysql> select concat(round(sum(data_length/1024/1024),2),'MB') as data from tables where table_schema='duck_store';
+-----+
| data |
+-----+
| 0.08MB |
+-----+
1 row in set (0.00 sec)
```

7.4 日志文件大小

利用 mysql 语句查询日志文件大小：

图 7.2 代码查询日志文件大小

```
mysql> show binary logs;
+-----+-----+-----+
| Log_name          | File_size | Encrypted |
+-----+-----+-----+
| mysql_bin.000001 | 157       | No       |
+-----+-----+-----+
1 row in set (0.00 sec)
```

7.5 数据库索引

本数据库每个表均采用其主码作为索引，无其他索引。

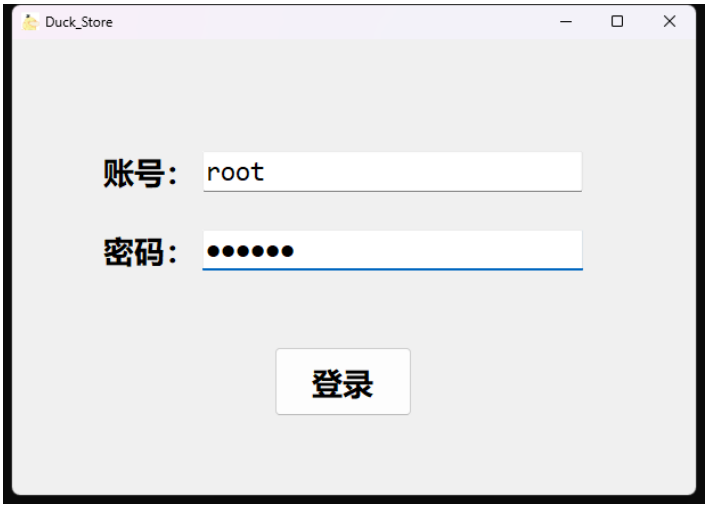
8. 数据库的实施与实现

8.1 前台应用程序的设计与实现

本管理系统采用 Qt 编写前台应用程序，并进行 UI 设计，将前台应用分为登录界面、首页、商品管理、库存管理、采购管理、出售管理、供应管理、人员管理、收支管理等若干界面，每个界面对应其特定操作，不同权限的用户可操作的界面不同。

1. 登录界面，对于登录界面，用户输入自己的账户和密码，以不同的权限等级进入数据库的管理界面。

图 8.1 登录界面



2. 首页界面，有软件 Logo、设计成员和指导老师信息。

图 8.2 首页界面



3. 商品管理界面，可对商品进行查询操作，按类型查询，添加、修改商品信息。

图 8.3 商品管理界面



4. 库存管理界面，可查看商品的库存信息，并按照特定顺序进行排序，以



excel 格式导出所有的库存信息。

图 8.4 库存管理界面

ID	类型	名称	数目
1	食品类	四个圈	150
2	饮品类	农夫山泉	140
3	饮品类	魔爪	70
4	饮品类	脉动	41
5	饮品类	可乐	105
6	饮品类	喜糖	105
7	饮品类	卡布奇诺	28
8	饮品类	牛奶	120
9	饮品类	拿铁	56
10	饮品类	咖啡	29
11	食品类	山楂丸	94
12	食品类	橙汁	110
13	食品类	大山楂丸	125
14	食品类	可达鸭	60
15	食品类	辣条	234
16	食品类	鸭头	49
17	食品类	瓜子	52
18	食品类	小红薯	70
19	食品类	沙琪玛	80
20	食品类	巧克力面包	60
21	食品类	巧克力	130
22	食品类	压缩饼干	60
23	食品类	火腿肠	121
24	食品类	汤圆	180
25	食品类	饼干	75
26	食品类	牙刷	60
27	日用品	牙膏	60
28	日用品	纸杯	90
29	日用品	纸巾	30

5. 采购管理界面，当便利店进行进货业务时，可插入进货的订单信息，也可删除采购记录和以 excel 格式导出所有的采购信息。

图 8.5 采购管理界面

ID	进价	数量	日期	供货商ID	商品ID
1	1	90	2023-06-01	1	2
2	2	50	2023-06-01	4	1
3	3	50	2023-06-01	4	3
4	2	50	2023-06-01	4	4
5	1.5	50	2023-06-01	4	5
6	1.5	50	2023-06-01	4	6
7	3	20	2023-06-01	4	7
8	1.5	60	2023-06-08	3	8
9	3	30	2023-06-08	3	9
10	2	30	2023-06-08	3	10
11	3	50	2023-06-08	2	11
12	2	50	2023-06-08	3	12
13	4	60	2023-06-08	2	13
14	6	30	2023-06-08	3	14
15	0.3	100	2023-06-15	5	15
16	2	30	2023-06-15	5	16
17	3	30	2023-06-15	5	17
18	1	30	2023-06-15	5	18
19	0.5	40	2023-06-15	5	19
20	2	30	2023-06-15	5	20
21	1.5	60	2023-06-15	5	21
22	1	30	2023-06-22	3	22
23	0.6	60	2023-06-22	3	23
24	2	80	2023-06-22	3	24
25	2	40	2023-06-22	3	25
26	1	30	2023-06-22	3	26
27	3	30	2023-06-22	3	27
28	3	40	2023-06-22	3	28

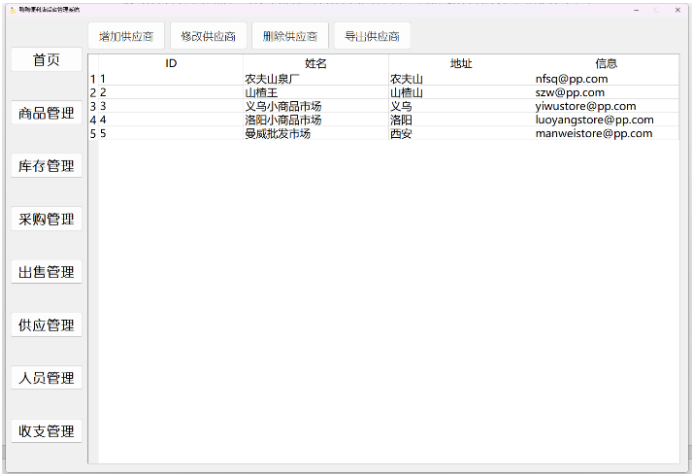
6. 出售管理界面，当便利店进行出售业务时，可插入出售的订单信息，也可删除出售记录和以 excel 格式导出所有的出售信息。

图 8.6 出售管理界面

ID	价格	数量	日期	员工ID	商品ID
1	2	3	2023-01-01	1	2
2	4.32	5	2023-01-01	1	4
3	6.5	6	2023-01-03	1	3
4	5	3	2023-01-05	2	35
5	6.5	2	2023-01-10	2	3
6	4.32	3	2023-01-13	3	4
7	7.2	5	2023-01-21	4	7
8	5.4	3	2023-01-23	5	10
9	6	3	2023-02-02	5	11
10	1	1	2023-02-03	1	15
11	2	3	2023-02-05	1	2
12	4.32	5	2023-02-15	1	4
13	6.5	6	2023-02-23	1	3
14	2	3	2023-02-27	2	23
15	1	2	2023-03-03	2	33
16	4.32	3	2023-03-07	3	4
17	7.2	5	2023-03-11	4	7
18	5.4	3	2023-03-13	5	10
19	6	3	2023-03-15	5	11
20	1	1	2023-03-15	1	15
21	2	3	2023-03-15	1	2
22	4.32	5	2023-03-16	1	4
23	5.4	6	2023-04-07	1	10
24	2	3	2023-04-12	2	2
25	6.5	2	2023-04-17	2	3
26	4.32	3	2023-04-19	3	4
27	7.2	5	2023-04-23	4	7
28	5.4	3	2023-04-26	5	10
29	6	3	2023-04-27	5	11

7. 供应管理界面，保存便利店的供应商相关信息，可增加供应商、修改供应商、删除供应商、excel 格式导出所有的供应商信息。

图 8.7 供应管理界面



The screenshot shows the 'Supply Management' interface. At the top, there are buttons for 'Add Supplier', 'Edit Supplier', 'Delete Supplier', and 'Export Supplier'. Below these is a table with columns: ID, Name, Address, and Information. The table contains 5 rows of data. On the left sidebar, there are buttons for 'Home', 'Goods Management', 'Inventory Management', 'Purchase Management', 'Sales Management', 'Supply Management' (which is highlighted), 'Personnel Management', and 'Financial Management'.

ID	姓名	地址	信息
1 1	农夫山泉厂	农夫山	nfsq@pp.com
2 2	山推王	山推山	szw@pp.com
3 3	义乌小商品市场	义乌	yiwustore@pp.com
4 4	洛阳小商品市场	洛阳	luoyangstore@pp.com
5 5	曼威批发市场	西安	manweistore@pp.com

8. 人员管理界面，保存便利店的店员相关信息，可增加员工、修改员工、删除员工、excel 格式导出所有的员工信息。

图 8.8 人员管理界面

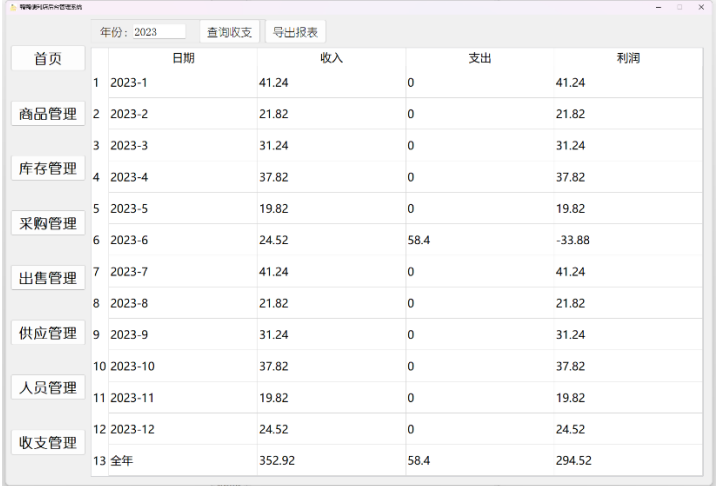


The screenshot shows the 'Personnel Management' interface. At the top, there are buttons for 'New Employee', 'Edit Employee', and 'Delete Employee'. Below these is a table with columns: ID, Name, Gender, Age, Salary, and Information. The table contains 5 rows of data. On the left sidebar, there are buttons for 'Home', 'Goods Management', 'Inventory Management', 'Purchase Management', 'Sales Management', 'Supply Management', 'Personnel Management' (which is highlighted), and 'Financial Management'.

ID	姓名	性别	年龄	薪资	信息
1 1	刘鹏鹏	男	20	7000	liuyaya@pp.com
2 2	张紫青	女	20	5000	zzzzq@pp.com
3 3	路遥	男	21	5555	luyao@pp.com
4 4	玛丽	女	19	5432	mali@pp.com
5 5	桌子	男	20	5555	zhuozi@pp.com

9. 收支管理界面，按年度查询便利店的收支情况，支持 excel 格式导出报表。

图 8.9 收支管理界面



The screenshot shows the 'Financial Management' interface. At the top, there are buttons for 'Year: 2023', 'Query Income and Expense', and 'Export Report'. Below these is a table with columns: Date, Income, Expense, and Profit. The table contains 13 rows of data. On the left sidebar, there are buttons for 'Home', 'Goods Management', 'Inventory Management', 'Purchase Management', 'Sales Management', 'Supply Management', 'Personnel Management', and 'Financial Management' (which is highlighted).

日期	收入	支出	利润
1 2023-1	41.24	0	41.24
2 2023-2	21.82	0	21.82
3 2023-3	31.24	0	31.24
4 2023-4	37.82	0	37.82
5 2023-5	19.82	0	19.82
6 2023-6	24.52	58.4	-33.88
7 2023-7	41.24	0	41.24
8 2023-8	21.82	0	21.82
9 2023-9	31.24	0	31.24
10 2023-10	37.82	0	37.82
11 2023-11	19.82	0	19.82
12 2023-12	24.52	0	24.52
13 全年	352.92	58.4	294.52

10. 对于员工账号，没有人员管理的权限，前端应用会拒绝店员账号的访问。

图 8.10 拒绝访问界面

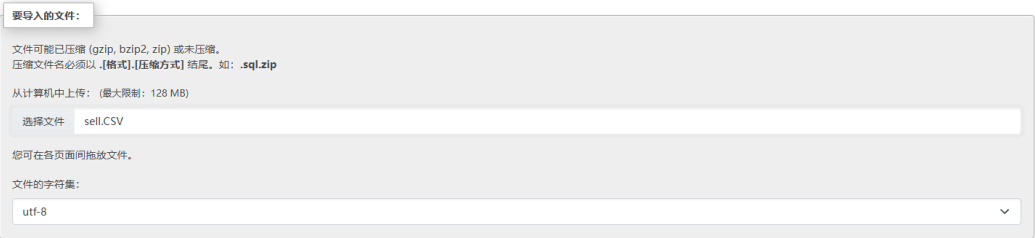


## 8.2 后台数据的载入

Mysql 数据库可以利用 sql 语句进行数据的插入，此外也可以通过 PhpMyAdmin 将数据按照 csv 格式批量导入至数据库中,实现大批量数据的导入。

图 8.11 数据导入界面

正在导入到数据表 "sell"



## 8.3 应用程序连接数据库的方法

本管理系统利用 ODBC 编程进行数据库的连接，ODBC 是微软公司开放服务体系中有关数据库的一个组成部分，其提供了一组访问数据库的应用程序编程接口，在避免了应用程序直接操作数据库系统的同时，极大的增强了应用程序的可移植性、扩展性和可维护性。

富有特点的是，本管理系统封装了 DataBase 类，将 C 语言的数据库连接、数据库初始化、数据库的操作语句封装成函数的形式，并且将 sql 类封装在 DataBase 类的私有成员中，隔绝了外部和内部的联系，一定程度上提高了数据库的安全性。

图 8.12 DataBase 类定义图

```

class DataBase {
public:
    DataBase();
    ~DataBase();
    void setRole(int user_role);
    int getRole();
    bool DB_init(std::string DataBase_UserName, std::string DataBase_Password);
    bool DB_query(std::string str, MYSQL_RES* res, std::pair<int, int>& size);
    //对于商品关系模式的操作:
    bool DB_insert_merchandise(int ID, std::string Type, std::string Name, double Price, double Discount = 1.0);
    bool DB_delete_merchandise(int ID);
    bool DB_modify_merchandise(int ID, std::string Type, std::string Name, int Number, double Price, double Discount);
    bool DB_query_merchandise(MYSQL_RES* res, std::pair<int, int>& size);
    //对于供应商关系模式的操作:
    bool DB_insert_supplier(int ID, std::string Name, std::string Address, std::string Information);
    bool DB_delete_supplier(int ID);
    bool DB_modify_supplier(int ID, std::string Name, std::string Address, std::string Information);
    bool DB_query_supplier(MYSQL_RES* res, std::pair<int, int>& size);
    //对于购入关系模式的操作:
    bool DB_insert_purchase(int ID, double Price, int Number, std::string Date, int Supplier_ID, int Merchandise_ID);
    bool DB_modify_purchase(int ID, double Price, int Number, std::string Date, int Supplier_ID, int Merchandise_ID);
    bool DB_delete_purchase(int ID);
    bool DB_query_purchase(MYSQL_RES* res, std::pair<int, int>& size);
    //对于售货员关系模式的操作:
    bool DB_insert_staff(int ID, std::string Name, std::string Sex, int Age, double Salary, std::string Information);
    bool DB_delete_staff(int ID);
    bool DB_modify_staff(int ID, std::string Name, std::string Sex, int Age, double Salary, std::string Information);
    bool DB_query_staff(MYSQL_RES* res, std::pair<int, int>& size);
    //对于售出关系模式的操作:
    bool DB_insert_sell(int ID, double Price, int Number, std::string Date, int Staff_ID, int Merchandise_ID);
    bool DB_modify_sell(int ID, double Price, int Number, std::string Date, int Staff_ID, int Merchandise_ID);
    bool DB_delete_sell(int ID);
    bool DB_query_sell(MYSQL_RES* res, std::pair<int, int>& size);
    //对于用户关系模式的操作:
    bool DB_query_user(MYSQL_RES* res, std::pair<int, int>& size);
private:
    MYSQL mysql;
    const std::string DataBase_Host = "localhost";
    const std::string DataBase_Name = "duck_store";
    unsigned int DataBase_Port = 3306;
    int role = 0;
};

```

## 8.4 数据库的备份

本管理系统采用 mysql 自带的 mysqldump 工具进行数据库备份，在 bash 界面中输入命令：

```

mysqldump -u root -p 123456 duck_store >
D:\develop\code\course_design\Database_course_design\backup\bak.sql;

```

则将数据库备份至指定文件夹”backup”下。

图 8.13 备份图

« develop > code > course_design > Database_course_design > backup			
名称	修改日期	类型	大小
 bak.sql	2023/6/24 22:54	SQL 源文件	15 KB

## 8.5 数据库的恢复

进入数据库，删除数据库，同时创建新的为空的数据库。

图 8.14 清空数据库

```

mysql> drop database duck_store;
Query OK, 7 rows affected (0.10 sec)

mysql> create database duck_store;
Query OK, 1 row affected (0.01 sec)

```

将备份好的数据库导入到刚刚新建好的数据库中：

```
mysql -u root -p 123456 duck_store <
D:\develop\code\course_design\Database_course_design\backup\bak.sql;
```

图 8.15 恢复数据库

表	操作	行数	类型	排序规则	大小	多余
<input type="checkbox"/> inventory	浏览  结构  搜索  插入  编辑  删除  视图	0		---	-	-
<input type="checkbox"/> merchandise	浏览  结构  搜索  插入  清空  删除	46	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
<input type="checkbox"/> purchase	浏览  结构  搜索  插入  清空  删除	28	InnoDB	utf8mb4_0900_ai_ci	32.0 KB	-
<input type="checkbox"/> sell	浏览  结构  搜索  插入  清空  删除	80	InnoDB	utf8mb4_0900_ai_ci	48.0 KB	-
<input type="checkbox"/> staff	浏览  结构  搜索  插入  清空  删除	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
<input type="checkbox"/> supplier	浏览  结构  搜索  插入  清空  删除	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
<input type="checkbox"/> user	浏览  结构  搜索  插入  清空  删除	2	MyISAM	utf8mb4_0900_ai_ci	2.2 KB	-
7 张表	总计	166	MyISAM	utf8mb4_0900_ai_ci	130.2 KB	0 字节