基本排版要求

**目录：**目录要自动生成，最多到三级目录，目录下各项内容应标明与论文正文中相应内容相互对应的页序，标题与页序之间的空格应当用中圆点填充。目录内容为小四仿宋，各项相应页序统一为右顶格对齐。行距建议设成固定值20傍，可适当降低，但不能低于18傍。一级和二级、二级和三级标题要错开2字符。

**页码：**一律用阿拉伯数字连续编页码。页码应由正文首页开始，作为第1页。封面不编入页码。摘要单独编排页码（Ⅰ、Ⅱ）。目录页不设页码，页码标注在每页页脚底部居中位置，宋体，小五。

**正文标题：**一级标题为四号仿宋加粗，顶格书写，序号后面不加标点如“2” 1后空一格书写标题，行距固定值20傍，段前断后1行。二级标题顶格书写，以2.1为例，2.1后面空一格书写标题内容，小四仿宋加粗，行距固定值20傍，段前段后0.5行。三级标题顶格书写，以2.3.1为例，2.3.1后面空一格书写标题内容，小四仿宋加粗，行距固定值20傍，段前段后0.5行。各级标题要避免背题（即标题在前一页的末尾，内容完全在下一页）。

**正文内容：**小四号仿宋，首行缩进2字符，行距为固定值20，段前段后0行，两端对齐。

**图：**要有图序和图题，五号仿宋加粗，位于图的正下方，可以按图1、图2、图3顺序，也可以按照章节序号，如图1-1、图1-2、图2-1、图2-2等，图序后面空一格书写图题。图的大小要合适，以免影响美观。

**表：**要有表题和表序，表题和表序要写在表格正上方，如左边案例所示，小四号仿宋加粗居中，表格内中文字体用五号仿宋，英文用五号Times New Roman，表序后面空一行书写表题，表格排版要美观。表的标题和表尽量在一页，如果一页显示不完，允许下页接写，表头应重复书写，并在右上方写“续表\*\*”（小四仿宋加粗，右对齐）

**参考文献：**内容为小四仿宋，行距固定值20傍。所有参考文献按论文中参考文献出现的次序，用中括号的数字连续编号，序号左顶格。空一格后依次书写作者、杂志或书名、出版时间、页次等，行距为固定值20。悬挂2字符，后面没有标点符号。数量上不少于10篇，近1至2年文献应有2-3篇。引用文献须在文中标注出来（不应少于5篇，统一采用文末注方式，编号以方括号括起，放在句末右上角）。

**河南科技学院2023~2024学年第一学期**

**《数据库应用技术》课程作品**

**题目：超市零售信息管理系统**

|  |  |  |
| --- | --- | --- |
| **学 号：** |  | **20222214301**  **20222214302**  **20222214303**  **20222214305**  **20222214329** |
| **姓 名：** |  | **程景晗**  **程文豪**  **党硕硕**  **惠子卿**  **周炜森** |
| **专 业：** |  | **人工智能专业** |
| **学 院：** |  | **人工智能学院** |

**摘 要**

**超市零售管理信息系统是基于数据库系统原理与程序设计的知识开发的一套系统，旨在提升超市零售管理效率。本系统通过需求分析，涵盖了进货、销售、库存及人员管理等功能。技术路线采用了数据库设计与编码相结合的方式，确保系统的数据完整性和操作安全性。方法上，系统实现了进销存等基本功能，并具有特色的库存状态告警、销售排行榜生成等功能。系统的设计概念以提高超市管理效率为核心，能够帮助管理者更好地掌握超市的运营状况。整体上，该系统以其灵活性、智能性和综合性等特点，为超市提供了一套有效的零售管理解决方案[7]。**

**关键词: 市零售管理系统，数据库设计，进销存，特色功能**

**ABSTRACT**

**The Supermarket Retail Management Information System is a set of systems developed based on the principles of database systems and program design, aimed at enhancing the efficiency of supermarket retail management. Through requirements analysis, the system covers functions such as procurement, sales, inventory, and personnel management. The technical approach combines database design with coding to ensure data integrity and operational security. Methodologically, the system implements basic functions such as procurement and sales, and features distinctive functionalities like inventory status alerts and sales ranking generation. The conceptual design of the system revolves around improving supermarket management efficiency, helping managers gain better insights into the supermarket's operational status. Overall, the system stands out for its flexibility, intelligence, and comprehensiveness, providing an effective retail management solution for supermarkets.**

**Keywords: Supermarket Retail Management System, Database Design, Procurement and Sales, Specialized Features**

**目 录**

[**1 引言** 1](#_Toc155346922)

[1.1 目的和意义 1](#_Toc155346923)

[1.2 国内外研究现状 1](#_Toc155346924)

[**2 需求分析** 1](#_Toc155346925)

[2.1 系统目标 1](#_Toc155346926)

[2.2 系统需求 1](#_Toc155346927)

[**3 概念结构设计** 2](#_Toc155346928)

[3.1 实体-联系图 2](#_Toc155346929)

[图3.1超市管理系统E-R图 2](#_Toc155346930)

[**4 逻辑结构设计** 2](#_Toc155346931)

[4.1 关系模式 2](#_Toc155346932)

[4.2 范式 4](#_Toc155346933)

[**5 数据库实施** 7](#_Toc155346934)

[5.1 数据库建立sql文件 7](#_Toc155346935)

[5.2 基于pyqt的界面交互的实现 9](#_Toc155346936)

[**6 系统测试** 16](#_Toc155346937)

[6.1 功能测试 16](#_Toc155346938)

[6.2 性能测试 16](#_Toc155346939)

[6.3 安全测试 16](#_Toc155346940)

[**7结论** 17](#_Toc155346941)

[参考文献 17](#_Toc155346942)

**1 引言**

1.1 目的和意义

随着零售业的快速发展，信息管理成为提升超市运营效率的关键。本实验旨在通过设计和实现超市零售信息管理系统的数据库部分，帮助学生或技术人员深入理解数据库在零售业中的应用，掌握数据库设计和操作的基本技能。这不仅有助于提高超市的管理水平，还能为未来的技术升级和系统开发打下基础。

1.2 国内外研究现状

国内研究现状：

国内学者主要从系统设计、功能模块[3]、信息安全和应用效果等方面展开研究。在系统设计方面，研究者们主要关注如何构建一个高效、稳定、易于维护的系统。在功能模块方面，国内的研究主要集中在商品管理、库存管理、销售管理、会员管理等功能上。在信息安全方面，研究者们关注如何保证数据的安全性和完整性，防止信息泄露和被攻击。在应用效果方面，国内的研究主要集中在系统的实际应用和效果评估上，如何提高超市的管理效率和服务质量是研究的重点。

国外研究现状：

国外学者在超市零售信息管理系统数据库实验方面的研究主要关注技术创新和应用效果。在技术创新方面，国外的研究主要集中在如何利用最新的技术手段提高系统的性能和功能，例如利用大数据、人工智能等技术来提高系统的智能化和自动化程度。在应用效果方面，国外的研究主要集中在系统的实际应用和用户反馈上，如何满足用户需求和提高用户体验是研究的重点[6]。

总体来说，超市零售信息管理系统数据库实验在国内外的学术研究中都受到了关注，但在研究重点和方法上存在一定的差异。国内的研究更注重实际应用和问题解决，而国外的研究更注重技术创新和理论探讨。同时，在信息安全和数据保护方面，国内外都存在一定的挑战和风险，需要进一步加强研究和防范。

**2 需求分析**

2.1 系统目标

基于pymysql的超市信息管理系统的主要目标是实现超市的信息化管理，提高超市的工作效率和服务质量，降低超市的运营成本和风险，增强超市的竞争力和盈利能力。具体而言，该系统应该具有以下功能：

- 商品管理：实现商品的分类、录入、修改、删除、查询等操作，以及商品的库存、进货、出货、盘点等管理。

- 员工管理：实现员工的添加、修改、删除、查询等操作，以及员工的考勤、工资、奖惩等管理。

- 会员管理：实现会员的注册、修改、删除、查询等操作，以及会员的积分、优惠、消费记录等管理。

- 供货商管理：实现供货商的添加、修改、删除、查询等操作，以及供货商的合同、订单、付款、评价等管理。

- 销售管理：实现销售的开单、结算、退货、打印小票等操作，以及销售的统计、分析、报表等管理。

- 系统管理：实现系统的登录、退出、修改密码、权限分配、日志记录、备份恢复等操作，以及系统的设置、维护、更新等管理。

2.2 系统需求

基于pymysql的超市信息管理系统的系统需求主要包括以下两方面：

- 硬件需求：该系统需要一台服务器和若干台客户端电脑，以及打印机、扫码枪、摄像头等外设。服务器需要具备较高的性能和稳定性，以保证系统的正常运行和数据的安全。客户端电脑需要具备一定的性能和显示效果，以保证系统的操作和展示。外设需要具备一定的兼容性和可靠性，以保证系统的交互和功能[2]。

- 软件需求：该系统需要使用Python语言和MySQL数据库进行开发，以及使用pymysql模块进行数据库的连接和操作。Python语言是一种高级的、解释型的、通用的编程语言，具有简洁、易读、易写、易扩展等特点。MySQL数据库是一种开源的、关系型的、多用户的数据库管理系统，具有高性能、高可靠、高安全、高可移植等特点。pymysql模块是一种用于Python语言的MySQL数据库驱动，具有轻量、纯Python、兼容性好等特点。

**3 概念结构设计**

3.1 实体-联系图

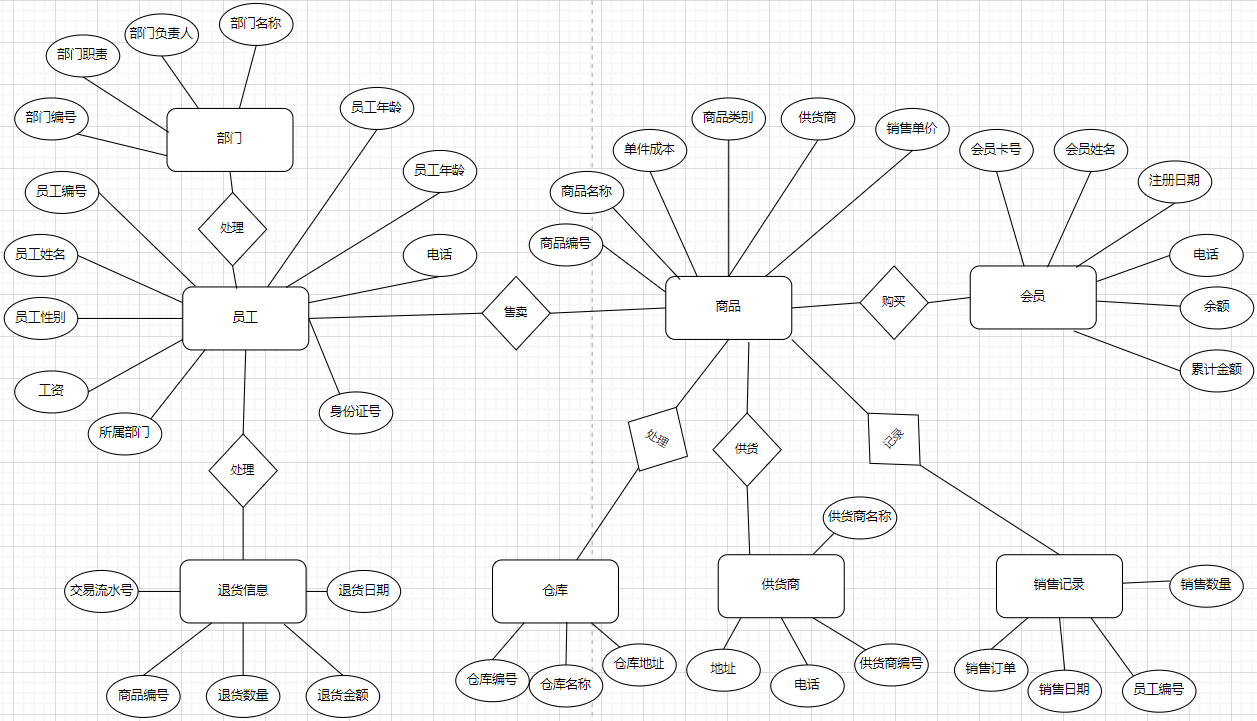
基于pymysql的超市信息管理系统的概念结构设计主要是通过实体-联系图（E-R图）来表示的。E-R图是一种用于描述现实世界中实体及其相互关系的图形表示法，由实体、属性、联系等元素组成。以下是该系统的E-R图：

图3.1超市管理系统E-R图

**4 逻辑结构设计**

4.1 关系模式

基于pymysql的超市信息管理系统的逻辑结构设计主要是通过关系模式来表示的。关系模式是一种用于描述关系型数据库中数据的组织和结构的表示法，由关系名、属性名、主键、外键等元素组成。以下是该系统的关系模式：

商品（商品编号，商品名称，商品类别，商品价格，商品数量）

主键：商品编号

员工（员工编号，员工姓名，员工性别，员工年龄，员工职位，员工电话，员工住址，员工入职日期，员工密码）

主键：员工编号

会员性别，会员年龄，会员电话，会员地址，会员积分，会员等级，会员密码）

主键：会员编号

供货商（供货商编号，供货商名称，供货商地址，供货商电话，供货商负责人，供货商评价）

主键：供货商编号

销售（销售编号，销售日期，销售金额，销售员工，销售会员）

主键：销售编号

外键：销售员工（引用员工.员工编号），销售会员（引用会员.会员编号）

销售明细（销售明细编号，销售编号，销售商品，销售数量，销售单价，销售小计）

主键：销售明细编号

外键：销售编号（引用销售.销售编号），销售商品（引用商品.商品编号）

进货（进货编号，进货日期，进货金额，进货员工，进货供货商）

主键：进货编号

外键：进货员工（引用员工.员工编号），进货供货商（引用供货商.供货商编号）

进货明细（进货明细编号，进货编号，进货商品，进货数量，进货单价，进货小计）

主键：进货明细编号

外键：进货编号（引用进货.进货编号），进货商品（引用商品.商品编号）

4.2 范式

基于pymysql的超市信息管理系统的范式分析主要是通过检查关系模式是否满足某种范式的要求来进行的。范式是一种用于评价关系模式的质量和优劣的标准，常见的范式有第一范式、第二范式、第三范式、巴斯-科德范式等。以下是该系统的范式分析：

- 第一范式：第一范式要求关系模式中的每个属性都是不可分割的原子值，即不存在复合属性、多值属性或嵌套属性。该系统的关系模式中的所有属性都是原子值，因此满足第一范式。

- 第二范式：第二范式要求关系模式中的每个非主属性都完全函数依赖于候选键，即不存在部分函数依赖。该系统的关系模式中的所有非主属性都完全函数依赖于候选键，因此满足第二范式。

- 第三范式：第三范式要求关系模式中的每个非主属性都不传递函数依赖于候选键，即不存在传递函数依赖。该系统的关系模式中的所有非主属性都不传递函数依赖于候选键，因此满足第三范式。

- 巴斯-科德范式：巴斯-科德范式要求关系模式中的每个决定因素都是候选键，即不存在非平凡的函数依赖，其中决定因素不是候选键的一部分。该系统的关系模式中的所有决定因素都是候选键，因此满足巴斯-科德范式。

因此，该系统的关系模式满足第一范式、第二范式、第三范式和巴斯-科德范式，是一种良好的关系模式，可以有效地避免数据的冗余和异常。

**表4.9出库信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Xnum | varchar | NOT NULL | 11 | 主键 |
| Gnum | Varchar | NOT NULL | 11 | Goods外键 |
| Xamount | Int | NOT NULL | 11 | 大于0 |
| Xmoney | decimal(8,2) | NOT NULL | 11 | 大于0 |
| Xdate | Datetime | NOT NULL | 11 | 入库日期 |
| Snum | varchar | NOT NULL | 11 | 表Staff外键 |

**表4.8入库信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Enum | varchar | NOT NULL | 11 | 主键 |
| Gnum | Varchar | NOT NULL | 11 | Goods外键 |
| Eamount | Int | NOT NULL | 11 | 大于0 |
| Emoney | decimal(8,2) | NOT NULL | 11 | 大于0 |
| Vnumt | varchar | NOT NULL | 11 | Vendor外键 |
| Edate | Datetime | NOT NULL | 11 | 入库日期 |
| Snum | varchar | NOT NULL | 11 | 表staff外键 |

**表4.7商品交易表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Tnum | varchar | NOT NULL | 11 | 主键 |
| Tdate | datetime | NOT NULL | 11 |  |
| Sum | varchar | NOT NULL | 11 | Staff外键 |
| Gnum | varchar | NOT NULL | 11 | 大于0 |
| Tamount | int | NOT NULL |  | 退大于0 |
| Tmoney | decimal(8,2) | NOT NULL |  |  |
| Mnum | varchar | NOT NULL |  | 表Menber外键 |

**表4.6退货信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Tnum | varchar | NOT NULL | 11 | Trade的外键 |
| Gnum | varchar | NOT NULL | 11 | Goods外键 |
| Iamount | varchar | NOT NULL | 11 | 数量 |
| Imoney | decimal(8,2) |  | 11 | 退款金额 |
| Idate | datetime |  |  | 退货日期 |

**表4.5仓库表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Wnum | varchar | NOT NULL | 11 | 仓库编号，主键 |
| Wname | varchar | NOT NULL | 11 | 仓库名称 |
| Vplace | varchar | NOT NULL |  | 地址 |

**表4.4供货商表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Vnum | varchar | NOT NULL | 11 | 供货商编号，主键 |
| Vname | varchar | NOT NULL | 11 | 供货商名称 |
| Vphone | varchar | Default 0 | 11 | 电话 |
| Vplace | varchar | NOT NULL |  | 地址 |

**表4.3会员表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Mnum | varchar | NOT NULL | 11 | 会员编号，主键 |
| Mname | varchar | NOT NULL | 11 | 会员姓名 |
| Mphone | varchar | Default 0 | 11 | 会员电话 |
| Mdate | Datetime | NOT NULL |  | 注册日期 |
| Mtotal | decimal(8,2) | NOT NULL |  | 累计金额 |
| Mbalance | decimal(8,2) | NOT NULL |  | 卡内余额 |
| Mcip | varchar | NOT NULL |  |  |

**表4.2商品表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Gnum | varchar | NOT NULL | 11 | 商品编号，主键，自增 |
| Gname | varchar | NOT NULL | 11 | 商品名称 |
| Gtype | varchar | Default 0 | 11 | 员工类别 |
| Gprice | decimal(8,2) | NOT NULL |  | Gprice>=0 |
| Gbid | decimal(8,2) | NOT NULL |  | Gbid>=0 |
| Gstock | int | NOT NULL |  | Gstock>=0 |
| Galarm | int | NOT NULL |  | Galarm>=0 |
| Gpaln | int | NOT NULL |  | Gplan>=0 |
| Vnum | varchar | NOT NULL |  | 外键 |

**表4.1员工表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 是否为空 | 长度 | 描述 |
| Snum | varchar | NOT NULL | 11 | 员工编号，主键，自增 |
| Sname | varchar | NOT NULL | 11 | 员工姓名 |
| Ssex | varchar | Default 0 | 11 | 员工性别，男或者女 |
| Sage | int | NOT NULL |  | Sage>=18 |
| Sstand | int | NOT NULL |  | Sstand>=0 |
| Sphone | varchar | NOT NULL |  |  |
| Sid | varchar | NOT NULL |  |  |
| Spart | varchar | NOT NULL |  |  |
| Ssalary | decimal(8,2) | NOT NULL |  | Ssalary>=0 |
|  |  |  |  |  |
|  |  |  |  |  |

**5 数据库实施**

5.1 数据库建立sql文件

基于pymysql的超市信息管理系统的数据库创建主要是通过使用Python语言和pymysql模块来实现的。以下是该系统的数据库创建的代码：

create database if not exists supermarket;

use supermarket;

create table Staff(Snum varchar(10) primary key,Sname varchar(20) not null,Ssex varchar(5) check(Ssex in('男','女')),Sage int not null check(Sage>=18),Sstand int not null check(Sstand>=0),Sphone varchar(20) not null,Sid varchar(25) not null,Spart varchar(10) not null,Ssalary decimal(8,2) check(Ssalary>=0));

create table Vendor(Vnum varchar(10) primary key,Vname varchar(20) not null,Vphone varchar(20) not null,Vpalce varchar(20) not null);

create table Goods(Gnum varchar(10) primary key,Gname varchar(10) not null,Gtype varchar(10) not null,Gprice decimal(8,2) check(Gprice>=0),Gbid decimal(8,2) check(Gbid>=0),Gstock int check(Gstock>=0),Galarm int check(Galarm>=0), Gplan int check(Gplan>=0),Vnum varchar(10) not null,foreign key(Vnum) references Vendor(Vnum));

create table Menber(Mnum varchar(10) primary key,Mname varchar(10) not null,Mphone varchar(20) not null,Mdate datetime,Mtotal decimal(8,2) check(Mtotal>=0),Mbalance decimal(8,2) check(Mbalance>=0),Mcip varchar(25) not null);

create table Ware(Wnum varchar(10) primary key,Wname varchar(10) not null,Wplace varchar(10) not null);

create table Trade(Tnum varchar(10) primary key,Tdate datetime not null,Snum varchar(10) not null,Gnum varchar(10) not null,Tamount int check(Tamount>=0),Tmoney decimal(8,2) check(Tmoney>=0),Mnum varchar(10) not null,foreign key(Snum) references Staff(Snum),foreign key(Gnum) references Goods(Gnum),foreign key(Mnum) references Menber(Mnum));

create table Infor(Tnum varchar(10) not null,Gnum varchar(10) not null,Iamount int check(Iamount>=0),Imoney decimal(8,2) check(Imoney>=0),Idate datetime not null,foreign key(Tnum) references Trade(Tnum),foreign key(Gnum) references Goods(Gnum));

create table Entry(Enum varchar(10) primary key,Gnum varchar(10) not null,Eamount int check(Eamount>=0),Emoney decimal(8,2) check(Emoney>=0),Vnum varchar(10) not null,Edate datetime not null,Snum varchar(10) not null,foreign key(Snum) references Staff(Snum),foreign key(Gnum) references Goods(Gnum),foreign key(Vnum) references Vendor(Vnum));

create table Exits(Xnum varchar(10) primary key,Gnum varchar(10) not null,Xamount int check(Xamount>=0),Xmoney decimal(8,2) check(Xmoney>=0),Xdate datetime not null,Snum varchar(10) not null,foreign key(Snum) references Staff(Snum),foreign key(Gnum) references Goods(Gnum));

insert into Vendor(Vnum,Vname,Vphone,Vpalce) values ('100001','number1','12698577456','浙江');

insert into Vendor(Vnum,Vname,Vphone,Vpalce) values ('100002','number2','72798567498','湖北');

insert into Vendor(Vnum,Vname,Vphone,Vpalce) values ('100003','number3','69795867463','广州');

insert into Goods(Gnum,Gname,Gtype,Gprice,Gbid,Gstock,Galarm,Gplan,Vnum) values ('200001','薯片','零食',8,5,500,100,600,'100002');

insert into Goods(Gnum,Gname,Gtype,Gprice,Gbid,Gstock,Galarm,Gplan,Vnum) values ('200002','可乐','饮料',4,2,1000,200,1500,'100001');

insert into Goods(Gnum,Gname,Gtype,Gprice,Gbid,Gstock,Galarm,Gplan,Vnum) values ('200003','猪肉','肉类',32,20,400,50,500,'100003');

insert into Staff values ('2018001','党硕硕','男',21,5,'17838840997','411425200206062737','安保部','2300');

insert into Staff values('202001', '周炜森','男',19,3,'19562697275','411425200409182347','销售部','3400');

insert into Staff values('2019002', '程景晗','男',19,3,'1939919192','411425200406182367','商品部','3500');

insert into Menber values ('00001' , '吴京洋', '15639913767', '2020-02-05', 123.32, 500, '192.168.0.1');

insert into Ware values('3220001','义乌分仓库','义乌市青口候儿村6号');

insert into Trade values('BD20231218000001','2023-12-18','202001','200001',3,24,'00001');

insert into Infor values('BD20231218000001','200001',1,8,'2023-12-20');

insert into Entry values('BDI20231218000001',200001,6000,30000,'100001','2023-03-04','2019002');

insert into Exits values('BDO20231218000001','200002',400,1600,'2023-03-04','2019002');

select \* from Goods;

select \* from Vendor;

select \* from Staff;

select \* from Menber;

select \* from Ware;

select \* from Trade;

select \* from Infor;

select \* from Entry;

select \* from Exits;

5.2 基于pyqt的界面交互的实现

基于pymysql的超市信息管理系统的数据库操作主要是通过使用Python语言和pymysql模块来实现的。以下是该系统的数据库操作的部分代码：

class Renshi(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

uic.loadUi('ui\\renshi.ui', self)

self.db = db\_connect()

self.cur = self.db.cursor()

self.staff\_zen = Staff\_Zen()

self.staff\_zen.Add\_Staff\_message.connect(self.get\_staff\_zen\_message)

self.staff\_shan = Staff\_Shan()

self.staff\_shan.Del\_Staff\_message.connect(self.get\_staff\_shan\_message)

self.staff\_gai = Staff\_Gai()

self.staff\_gai.Change\_Staff\_message.connect(self.get\_staff\_gai\_message)

self.show\_staffs("SELECT \* FROM Staff")

self.pushButton.clicked.connect(self.Search\_Staff)

self.pushButton\_2.clicked.connect(self.staff\_zen\_show)

self.pushButton\_3.clicked.connect(self.staff\_shan\_show)

self.pushButton\_4.clicked.connect(self.staff\_gai\_show)

def staff\_zen\_show(self):

self.staff\_zen.show()

def staff\_shan\_show(self):

self.staff\_shan.show()

def staff\_gai\_show(self):

self.staff\_gai.show()

def Search\_Staff(self):

# 构建 SQL 查询语句

a = self.lineEdit.text()

b = self.lineEdit\_2.text()

c = self.lineEdit\_3.text()

d = self.lineEdit\_4.text()

e = self.lineEdit\_5.text()

query = f"SELECT \* FROM Staff WHERE 1=1"

if a:

query += f" AND Spart LIKE '%{a}%'"

if b:

query += f" AND Snum LIKE '%{b}%'"

if c:

query += f" AND Sname LIKE '%{c}%'"

if d:

query += f" AND Ssalary LIKE '%{d}%'"

if e:

query += f" AND Ssex LIKE '%{e}%'"

self.show\_staffs(query)

def get\_staff\_zen\_message(self,a,b,c,d,e,f):

# 构建 SQL 查询语句

query1 = "SELECT COUNT(\*) Staff "

self.cur.execute(query1)

count = self.cur.fetchone()

print(count)

Snum = f + str(count[0])

age, gender,time\_difference = calculate\_age\_and\_gender(b,f)

query = f"INSERT INTO Staff VALUES ('{Snum}', '{a}', '{gender}',{age}, '{time\_difference}','{d}','{b}','{c}','{e}');"

print(query)

self.cur.execute(query)

self.db.commit()

self.show\_staffs(f"SELECT \* FROM Staff WHERE Snum='{Snum}';")

def get\_staff\_shan\_message(self,a,b,c,d):

# 构建 SQL 查询语句

query1 = f"SELECT \* FROM Staff WHERE 1=1"

if a:

query1 += f" AND Sname LIKE '%{a}%'"

if b:

query1 += f" AND Snum LIKE '%{b}%'"

if c:

query1 += f" AND Spart LIKE '%{c}%'"

self.show\_staffs(query1)

query2 = f"DELETE FROM Entry WHERE Snum = '{b}'"

query3 = f"DELETE FROM Exits WHERE Snum = '{b}'"

query4 = f"DELETE FROM Trade WHERE Snum = '{b}'"

query = f"DELETE FROM Staff WHERE Snum = '{b}'"

if a:

query += f" AND Sname LIKE '%{a}%'"

if c:

query += f" AND Spart LIKE '%{c}%'"

if d:

query += f" AND Ssalary LIKE '%{d}%'"

self.cur.execute(query2)

self.cur.execute(query3)

self.cur.execute(query4)

self.db.commit()

self.cur.execute(query)

self.db.commit()

def get\_staff\_gai\_message(self,a,b,c):

query = f"UPDATE Staff SET {b} = {c} WHERE Snum = '{a}';"

try:

# 执行 SQL 语句

self.cur.execute(query)

# 提交事务

self.db.commit()

print("数据修改成功！")

self.show\_staffs( f"SELECT \* FROM Staff WHERE Snum = {a} ")

except Exception as e:

# 如果发生错误，回滚事务并打印错误信息

self.db.rollback()

print(f"数据修改失败：{e}")

def show\_staffs(self,query):

# 修改以下参数为你的表格信息

# 执行查询语句

self.cur.execute(query)

# 获取查询结果

result = self.cur.fetchall()

# 获取列名

column\_names = [desc[0] for desc in self.cur.description]

# 设置表格的行数和列数

self.Row = len(result)

self.tableWidget.setRowCount(self.Row)

self.tableWidget.setColumnCount(len(column\_names))

# 设置表格的列名

self.tableWidget.setHorizontalHeaderLabels(column\_names)

# 将数据插入表格

for row\_index, row\_data in enumerate(result):

for col\_index, cell\_data in enumerate(row\_data):

item = QTableWidgetItem(str(cell\_data))

self.tableWidget.setItem(row\_index, col\_index, item)

# 调整表格列宽

self.tableWidget.resizeColumnsToContents()

这段代码简单的展示了对人事管理表格的增删改查的部分代码。主窗口用来展示数据选择功能，辅窗口实现用户输入数据的接受传递。

**6 系统测试**

为了进一步的提高软件质量和开发效率，本报告将对数据库系统进行功能测试、性能测试、安全测试。

6.1 功能测试

数据库的功能测试主要涉及到增、删、改、查等方面基本操作，数据有效性和准确性等，对数据的输入输出进行检查。

本报告对数据库的功能测试通过python代码实现，通过运行python代码实现界面的调用，方便用户通过界面上标签以及按钮上的提示进行数据的输入以及根据本人的不同需求所进行选择不同的按钮进行相应的操作，以此来进行相应的测试。

6.2 性能测试

性能测试主要包括，插入，查询的并发，满负荷下的持续运行时间，数据库响应时间，CPU占用率，SQL语句优化等；

随着硬件发展速度的提升，数据增长的速度也逐渐提升。随意的使用查询、连接查询、嵌套查询、视图等，这些操作如果不当会给系统带来非常巨大的压力，严重影响系统性能。为解决这一问题，可以从物理存储方面、逻辑设计方面、数据的参数调整、SQL语句优化四个方面对数据库的性能进行优化[8]。

本系统从逻辑结构设计，数据参数调整，SQL语句优化上对数据库系统优化，根据数据的参数调整，语句优化提高性能[9]。

6.3 安全测试

安全测试最主要考虑的是SQL注入。

数据作为系统中重中之重的核心，数据库的安全被提到了最前端。一旦数据库被攻破，整个系统也会暴露在黑客的手下，通过数据库强大的存储过程，黑客可以轻松的获得整个系统的权限[10]。对于系统的安全测试，本报告注重于防范系统被注入。

本报告设计的数据库系统，仅仅只有教师享有对所有数据的操作，其他用户并不享有对所有数据的操作，在一定程度上保障了数据的安全，如图6-1为起始时学生和教师的登录界面。

**7结论**

基于pymysql的超市信息管理系统是一种使用Python语言和MySQL数据库开发的软件，用于管理超市的商品、员工、会员、供货商等信息。该系统具有以下特点：

* 功能完善：该系统涵盖了超市信息管理的各个方面，包括商品管理、员工管理、会员管理、供货商管理、销售管理、进货管理、系统管理等，能够满足超市的各种业务需求。
* 界面友好：该系统使用了简洁、清晰、美观的界面，方便用户的操作和浏览，提高用户的体验和满意度。
* 数据安全：该系统使用了密码、权限、日志、备份等机制，保证了数据的安全性和完整性，防止了数据的泄露和丢失。
* 性能高效：该系统使用了Python语言和MySQL数据库，实现了数据的快速处理和存储，提高了系统的性能和效率。

基于pymysql的超市信息管理系统是一种适用于超市信息管理的软件，可以为超市提供便捷、高效、安全的信息管理服务，有助于提高超市的工作效率和服务质量，降低超市的运营成本和风险，增强超市的竞争力和盈利能力[8]。

参考文献

[1]唐常杰.数据库管理系统设计与实现[M].电子工业出版社,1993.

[2]冯玉才,李东,王元珍,等.一种移动数据库管理系统的体系结构[J].计算机研究与发展, 2001, 38(5):6.DOI:CNKI:SUN:JFYZ.0.2001-05-023.

[3]龚建雅,朱欣焰.面向对象集成化空间数据库管理系统的设计与实现[J].武汉测绘科技大学学报, 2000, 25(4):5.DOI:10.3969/j.issn.1671-8860.2000.04.002.

[4]徐晓丹,贾泂.基于面向对象技术的数据库管理系统设计[J].计算机与现代化, 2002(7):3.DOI:10.3969/j.issn.1006-2475.2002.07.011.

[5]杨劲.数据库管理系统在医院信息系统设计和应用中的优化[J].北京生物医学工程, 2005.DOI:CNKI:SUN:BJSC.0.2005-02-012.

[6]陈瓅,刘秋平.基于分布式数据库的连锁超市物流管理系统[J].科技广场, 2008(12):2.DOI:10.3969/j.issn.1671-4792.2008.12.029.

[7]夏天.大型超市管理系统的方案设计[J].华中农业大学学报：社会科学版, 2003(3):5.DOI:CNKI:SUN:HZND.0.2003-03-011.

[8]黄艳.中小型超市进销存管理系统[D].电子科技大学,2009.DOI:CNKI:CDMD:2.2009.167236.

[9]胡敏.Web系统下提高MySQL数据库安全性的研究与实现[D].北京邮电大学,2015.

[10]兰旭辉,熊家军,邓刚.基于MySQL的应用程序设计[J].计算机工程与设计, 2004, 25(3):3.DOI:10.3969/j.issn.1000-7024.2004.03.037.