|  |
| --- |
|  |
|  |

**数据库课程设计**

|  |
| --- |
|  |

院系 计算机科学与技术学院

专业 15级计算机科学与技术

项目名称 仓储管理系统

小组成员 E11514108 许哲龙 E11514082 张宝根 E11514130 邹梁琦

E11514040 张银磊

指导教师

日期 2017.11.3

目录

[第一章 需求分析 2](#_Toc496878029)

[1.1系统需求 2](#_Toc496878030)

[1.2数据需求 2](#_Toc496878031)

[1.3处理需求 2](#_Toc496878032)

[1.4安全性需求 2](#_Toc496878033)

[1.5完整型需求 3](#_Toc496878034)

[1.6 数据流图 3](#_Toc496878035)

[1.6.1 .系统顶层数据流图 3](#_Toc496878036)

[1.6.2 .数据流图分解 4](#_Toc496878037)

[1.6.3 数据流图整体结构 5](#_Toc496878038)

[1.7数据字典 6](#_Toc496878039)

[第二章 概念设计 8](#_Toc496878040)

[2.1 实体之间的联系 8](#_Toc496878041)

[2.2 E-R图 8](#_Toc496878042)

[第三章 逻辑设计 10](#_Toc496878043)

[3.1 概念模型向关系模型转换 10](#_Toc496878044)

[3.2概念模型的优化 11](#_Toc496878045)

[3.3数据库的表设计 12](#_Toc496878046)

[3.4数据库视图的创建 14](#_Toc496878047)

[第四章 物理设计 15](#_Toc496878048)

[4.1存储结构 15](#_Toc496878049)

[4.2关系模式的存储方法 15](#_Toc496878050)

[第五章 数据库的实施 15](#_Toc496878051)

[5.1 实现 15](#_Toc496878052)

[5.1.1 数据库和表的建立 15](#_Toc496878053)

[5.1.2 视图的实现 20](#_Toc496878054)

[5.1.3 数据的录入 25](#_Toc496878055)

[5.2建立触发器、存储过程、索引 29](#_Toc496878056)

# 第一章 需求分析

需求分析阶段的任务是通过分析系统的数据需求和处理需求，建立起系统的分层数据流图和数据字典。数据流图反映数据在系统中的流向及处理转换过程，数据字典是数据流图中各类数据的定义和描述的集合。

## 1.1系统需求

软件的用户为中小型企业仓库管理人员。

●系统用户进入系统需要经过身份验证。

●系统用户在登录系统以后，在不同页面跳转浏览时的身份验证。

仓库管理功能

●库存信息的管理，包括添加仓库、查看仓库、删除仓库。

●提货单信息的添加。

●货物类别信息的添加。

●物品信息的添加。

●提货单的管理。

●软件能进行各种物资的入库、出库操作，库存物资的数量、金额同步更新。

## 1.2数据需求

仓储管理系统涉及到的数据主要有：

1. 管理员：包括管理员编号、账号、密码、姓名、等级
2. 供应商：包括供应商编号、供应商名、电话、地址
3. 采购员：包括账号、密码、姓名
4. 订单：包括订单号、物品信息、供应商信息、订货时间、数量
5. 物品：包括物品编号、仓库编号、物品名称、价格、类型
6. 仓库：包括仓库编号、仓库名、管理员编号

## 1.3处理需求

管理员处理需求：管理员在以用户名和密码登录系统后，可以查看、添加、删除信息，这些信息包括月报表、订单、仓库、物品等。

## 1.4安全性需求

根据以上需求分析，系统将用户定义为中小型企业仓库管理人员。不同等级的仓库管理人员或是采购员，在系统中的作用和权限有所不同。采购员只能浏览物品信息，发起采购请求，以及更新个人信息。管理员拥有所有权限，但是在查看用户信息时，不能查看用户的密码，但是高等级的管理员拥有管理所有低等级的管理员的权限。在本系统中，将权限分为俩种，一种为超级管理员权限，该权限可以管理系统中定义的所有仓库；而另一种为普通管理员权限，该权限只能管理分配给他的某个仓库。

## 1.5完整型需求

(1)商品库存要求大于0

(2)订单里面的商品数量要求大于0

(3)用户名要求是由5位任意字符组成

(4)密码必须在20位范围内

修改、删除物品或仓库信息时，要求订单的信息要和物品表、仓库表以及用户表的信息一致

## 1.6 数据流图

### 1.6.1 .系统顶层数据流图

信息流端点为管理员、供应商和采购员，存储分为用户信息，库存信息，订货信息，出库信息，入库信息五个数据库表进行存储。

管理员 供应商

用户信息 入库信息

订货信息

D1 用户信息

信息处理

D2 库存信息

出库信息 D3 订货信息

用户信息

D4 入库信息

采购员 D5 出库信息

### 1.6.2 .数据流图分解

商品入库数据流图：

入库信息

产生入库表

更新数据库

处理事务

入库单

仓库管理员

事务 事务

供应商

库存信息

商品出货数据流图：

出库信息

更新数据库

处理事务

出库单

采购员

产生出库表

事务 事务

管理员

库存信息

商品订货数据流图：

订货信息

产生订货表

处理事务

更新数据库

事务 事务 订货单

采购员

管理员

库存信息

### 1.6.3 数据流图整体结构

用户信息

入库单 产生入库表

用户信息 管理员

操作数据库 入库信息

事务

更新数据库 库存信息 处理事务

事务

处理事务 出库信息 更新数据库

事务

产生订货表 订货信息 产生出库表 供应商

订货单 出库单

用户信息 采购员

## 1.7数据字典

1：

|  |  |
| --- | --- |
| 名字： | 用户表 |
| 描述： | 记录所有可以登录该系统的用户名、密码和其他相关数据； |
| 定义： | 用户名+密码+备注+IP地址； |
| 位置： | 输出到打印机 |

2：

|  |  |
| --- | --- |
| 名字： | 供应商信息表 |
| 描述： | 此表记录所有供应商的信息； |
| 定义： | 供应商名称+供应商编号+备注； |
| 位置： | 输出到打印机； |

3：

|  |  |
| --- | --- |
| 名字： | 提货单位信息表 |
| 描述： | 该表是指某物在出库时提货者的信息；、 |
| 定义： | 使用编号+名称+备注； |
| 位置： | 输出到打印机； |

4:

|  |  |
| --- | --- |
| 名字： | 仓库信息表 |
| 描述： | 一个企业可能会有许多仓库，就存放在这个表中； |
| 定义: | 仓库编号+仓库名称+备注； |
| ： | 输出到打印机； |

5：

|  |  |
| --- | --- |
| 名字： | 物品类别信息表 |
| 描述： | 该表记录仓库中具体的物件名称和其他相关的信息； |
| 定义： | 类别名称+类型编号+所属仓库名+所属仓库编号 |
| 位置： | 输出到打印机； |

6.

|  |  |
| --- | --- |
| 名字： | 物品信息表 |
| 描述： | 此表记录仓库中具体的物件名称和其他相关的信息； |
| 定义: | 仓库编号+类型编号+品名编号+品名物品的名称+计量单位+当前库存数+当前库存总金额+最大储存量+最小储存量+储存地点+规格+货号+货名； |
| 位置： | 输出到打印机； |

7.

|  |  |
| --- | --- |
| 名字： | 入库/出库明细表 |
| 描述： | 该表记录所有各个仓库的入库、出库的记录，每当进行出库或者入库操作是，就会有一条记录插入该表； |
| 定义： | 编号+记录日期+种类+号数+收入数量+收入单价+收入金额+发出数量+发出单价+发出金额+结存数量+结存单价+结存金额+品名编号+仓库编号+供应商编号+接受这编号+剩余数量 |
| 位置： | 输出到打印机； |

# 第二章 概念设计

## 2.1 实体之间的联系

根据需求分析，可以归纳出合适的联系：

1. 一个管理员可以管理多个仓库，一个仓库可以由多个管理员管理；
2. 一个管理员可以被多个供应商供应，一个供应商可以向多个管理员供应；
3. 一个分类可以包括多个物品，一个物品只能属于一个分类；
4. 一个仓库可以拥有多个分类，而一个分类只能属于一个仓库；
5. 一个采购员可以查询多个订单，一个订单可以被多个管理员查询；
6. 一个采购员可以查询多个订单，而一个订单只能被一个采购员查询
7. 一个采购员可以查看大量商品，一个商品可以被多个采购员查询

## 2.2 E-R图

【注意】

1. 管理员拥有采购员的所有权限；

管理员号

供应商名

供应商号

IP地址

备注

密码

管理员号

用户名

状态

仓库编号

类名

类型编号

备注

仓库名

仓库编号

名称

类型编号

价格

备注

当前库存

物品编号

库存上限

仓库编号

备注

金额

数量

日期

采购员号

物品编号

类型编号

订单编号

采购员名

备注

采购员号

管理员

仓库

从属理

分类

从属理

供应理

物品

采购员

供应商

订单

报表

仓库管理理

m n 1 n

m m 1

等级

仓库查询理

n n n

物品查询理

m n

1

创建查询订单理

n

# 第三章 逻辑设计

## 3.1 概念模型向关系模型转换

将E-R图转换成关系模型实际上就是要将实体型、实体的属性和实体之间的联系转换成关系模式，这种转换关系遵循一定的规则：

一个实体型转换成一个关系模式。实体的属性就是关系的属性，实体的码就是关系的码。

对于实体型的关系有不同的情况：

1. 一个1:1联系可以转化成一个单独的关系模式，也可以与任意一端对应的关系模式合并。本次E-R图中没有1:1关系。
2. 一个1:n联系可以转化成一个独立的关系模式，也可以与n端对应的关系模式合并。
3. 一个m:n联系可以转换成一个独立的关系模式。与该关系相连的各实体的码以及联系本身的属性均转换成关系的属性。各个实体的码组成关系的码或者码的一部分。
4. 3个或3个以上的实体间的一个多元关系可以转换成一个关系模式。
5. 具有相同码的关系模式可合并
6. 本次E-R图转换的具体关系模式如下：

|  |  |
| --- | --- |
| 关系模式： | 管理员（管理员号，用户名，密码，IP地址，状态，备注，等级） |
| 主键： | 管理员号 |
| 外键： | 无 |

|  |  |
| --- | --- |
| 关系模式： | 仓库（仓库编号，管理员号，仓库名，备注） |
| 主键： | 仓库编号 |
| 外键： | 管理员号 |

|  |  |
| --- | --- |
| 关系模式： | 分类（类型编号，类名，仓库编号） |
| 主键： | 类型编号 |
| 外键： | 仓库编号 |

|  |  |
| --- | --- |
| 关系模式： | 供应商（供应商号，供应商名，备注） |
| 主键： | 供应商名 |
| 外键： | 无 |

|  |  |
| --- | --- |
| 关系模式： | 物品（物品编号，仓库编号，类型编号，名称，库存上限，当前库存，备注，计量单位） |
| 主键： | （物品编号，类型编号） |
| 外键： | 仓库编号，类型编号 |

|  |  |
| --- | --- |
| 关系模式： | 订单（订单编号，类型编号，物品编号，采购员号，日期，数量，金额） |
| 主键： | 订单编号 |
| 外键： | 类型编号，物品编号，采购员号 |

|  |  |
| --- | --- |
| 关系模式： | 采购员（采购员号，采购员名，备注） |
| 主键： | 采购员号 |
| 外键： | 无 |

|  |  |
| --- | --- |
| 关系模式： | 仓库管理（管理员号，仓库编号） |
| 主键： | （管理员号，仓库编号） |
| 外键： | 管理员号，仓库编号 |

|  |  |
| --- | --- |
| 关系模式： | 供应（供应商号，管理员号） |
| 主键： | （供应商号，管理员号） |
| 外键： | 供应商号，管理员号 |

|  |  |
| --- | --- |
| 关系模式： | 仓库查询（仓库编号，采购员号） |
| 主键： | （仓库编号，采购员号） |
| 外键： | 仓库编号，采购员号 |

|  |  |
| --- | --- |
| 关系模式： | 物品查询（采购员号，物品编号） |
| 主键： | （采购员号，物品编号） |
| 外键： | 采购员号，物品编号 |

## 3.2概念模型的优化

（1）管理员（管理员号，用户名，密码，IP地址，状态，备注，等级）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（2）仓库（仓库编号，管理员号，仓库名，备注）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（3）分类（类型编号，类名，仓库编号）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（4）供应商（供应商号，供应商名，备注）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（5）物品（物品编号，仓库编号，类型编号，名称，库存上限，当前库存，备注，计量单位）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（6）采购员（采购员号，采购员名，备注）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（7）订单（订单编号，类型编号，物品编号，采购员号，日期，数量，金额）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（8）管理（管理员号，仓库编号）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（9）供应（供应商号，管理员号）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（10）仓库查询（仓库编号，采购员号）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

（11）物品查询（采购员号，物品编号）

该关系模式不存在非主属性对码的部分依赖和函数依赖，也不存在主属性对码的部分依赖和函数依赖，故属于**BCNF**

## 3.3数据库的表设计

进一步确定上一章逻辑设计中设计好的关系模式中各个数据项的类型和长度，将每个关系转换为数据库中的二维表格，并确定了各个表的主键和外键，得到以下表结构：

（1）Administrator表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型及长度 | 约束 | 默认值 |
| Ano | 管理员号 | char(6) | 主键 |  |
| Aname | 用户名 | Varchar(50) | Not null |  |
| Apassword | 密码 | Varchar(20) | Not null |  |
| Aip | IP地址 | Varchar(20) | Not null |  |
| Astatus | 状态 | Varchar(10) | Not null |  |
| Aremark | 备注 | Varchar(100) |  |  |
| Alevel | 等级 | int | >0 |  |

（2）Storehouse表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型及长度 | 约束 | 默认值 |
| Sno | 仓库编号 | Char(3) | 主键 |  |
| Ano | 管理员号 | Char(6) | 外键 |  |
| Sname | 仓库名 | Varchar(20) | Not null |  |
| Sremark | 备注 | Varchar(100) |  |  |

（3）Classification表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型及长度 | 约束 | 默认值 |
| Cno | 类型编号 | Char(5) | 主键 |  |
| Cname | 类名 | Varchar(20) | Not null |  |
| Sno | 仓库编号 | Char(3) | 外键 |  |

（4）Supplier表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型及长度 | 约束 | 默认值 |
| SUno | 供应商号 | Char(6) | 主键 |  |
| SUname | 供应商名 | Varchar(20) | Not null |  |
| SUremark | 备注 | Varchar(100) |  |  |

（5）Goods表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型及长度 | 约束 | 默认值 |
| Gno | 物品编号 | Char(7) | 主键 |  |
| Sno | 仓库编号 | Char(3) | 外键 |  |
| Cno | 类型编号 | Char(5) | 外键 |  |
| Gname | 名称 | Varchar(20) | Not null |  |
| GStockLimit | 库存上限 | Int | >=0 |  |
| GNowStock | 当前库存 | Int | >=0 |  |
| Gremark | 备注 | Varchar(100) |  |  |
| Gprice | 价格 | Float | >0 |  |

（6）Buyer表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型及长度 | 约束 | 默认值 |
| Bno | 采购员号 | Char(6) | 主键 |  |
| Bname | 采购员名 | Varchar(20) | Not null |  |
| Bremark | 备注 | Varchar(100) |  |  |

（7）Orders表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型及长度 | 约束 | 默认值 |
| Ono | 订单编号 | Char(20) | 主键 |  |
| Cno | 类型编号 | Char(5) | 外键 |  |
| Gno | 物品编号 | Char(7) | 外键 |  |
| Bno | 采购员号 | Char(6) | 外键 |  |
| Odate | 日期 | Date | Not null |  |
| Onumber | 数量 | Int | >0 |  |
| Omoney | 金额 | Float | >0 |  |

## 3.4数据库视图的创建

（1）创建仓库信息查询的视图；

（2）创建销量前10名商品的视图；

（3）创建月报表视图；

（4）创建订单视图；

（5）创建用户个人信息视图。

# 第四章 物理设计

为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程就是数据库的物理设计。物理设计时要确定数据库的存储路径、数据规模和增长速度等，在数据库管理系统中创建数据库，建立数据库的所有数据模式，并根据访问要求给数据库的基本表设计适当的索引作为存取路径。

## 4.1存储结构

根据本系统的数据库的使用情况，主数据文件信息量大且使用频繁将其存储在高速存储器上。将日志文件和数据库对象分别放在不同的磁盘上可以改进的性能。所以系统将日志文件和数据文件存放在不同的磁盘上以便提高查询效率，同时这样可以提高物理I/O读写效率。数据库备份文件和日志等文件因为使用频率小而且数据量非常大，存放在低速存储设备上。

确定数据库的存储结构主要指确定数据的存放位置和存储结构，包括确定关系、索引、日志、备份等的存储安排及存储结构，以及确定系统存储参数的配置。

## 4.2关系模式的存储方法

关系模式采用索引存取方法与聚簇索引共用。具体如下：

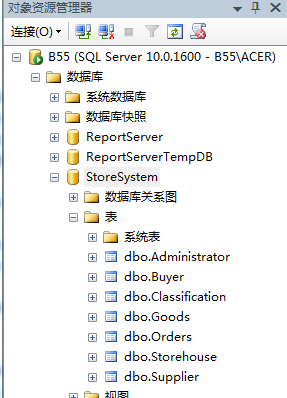
1. 为每个关系建立主键
2. 在经常用于查询条件的物品编号建立普通索引
3. 在经常查询的物品名称、价格建立聚簇索引

# 第五章 数据库的实施

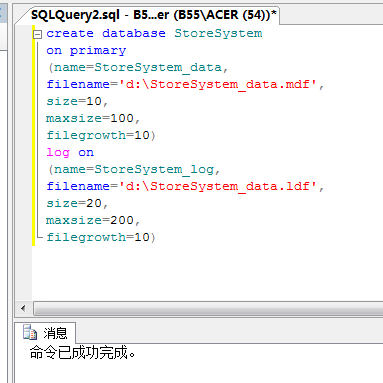
## 5.1 实现

### 5.1.1 数据库和表的建立

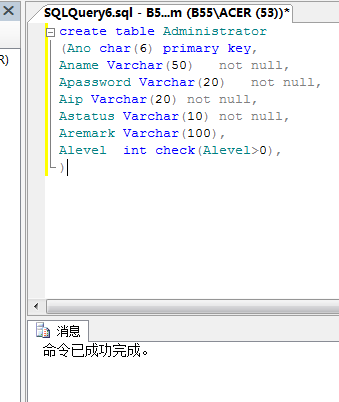
数据库的整体结构：



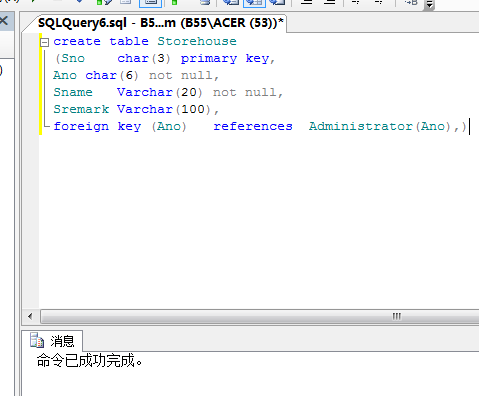
1. 创建数据库StoreSystem；



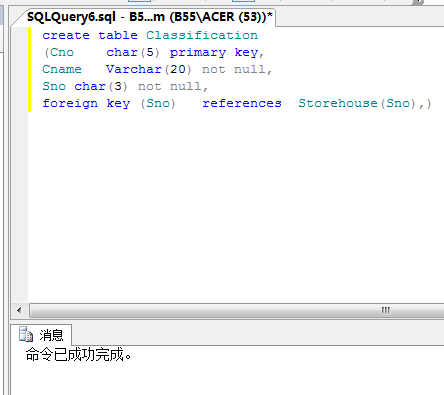
1. 创建Administrator表；



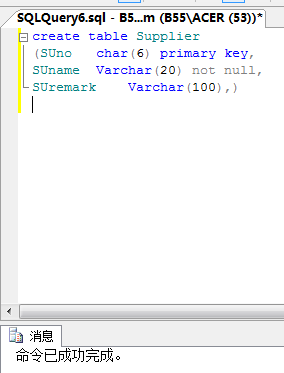
1. 创建Storehouse表；



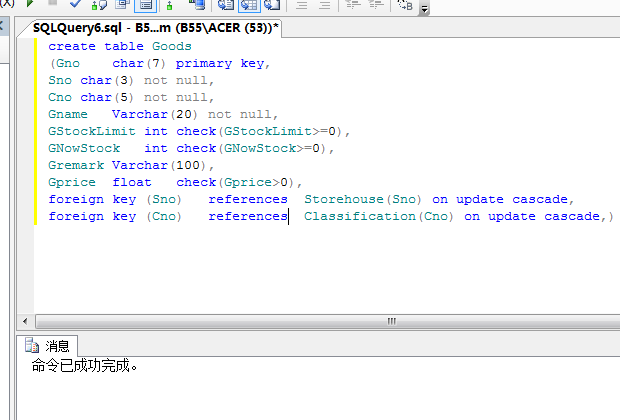
1. 创建Classification表；



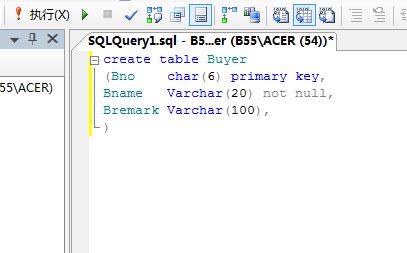
1. 创建Supplier表；



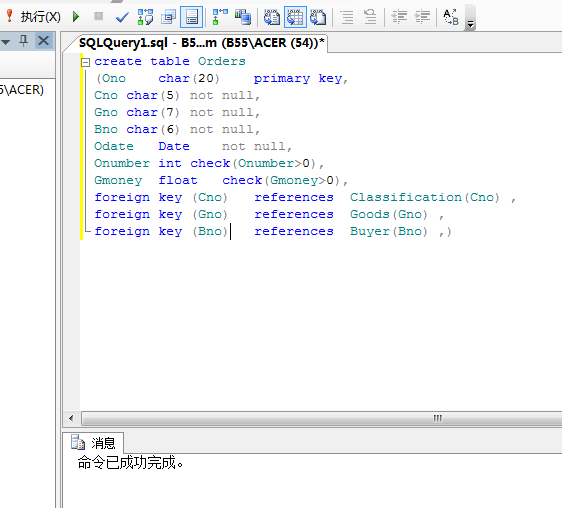
1. 创建Goods表；



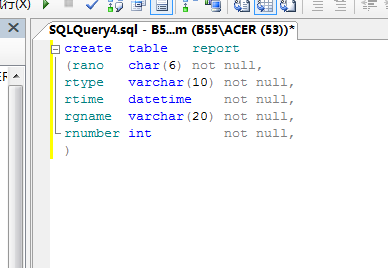
1. 创建Buyer表；



1. 创建Orders表；



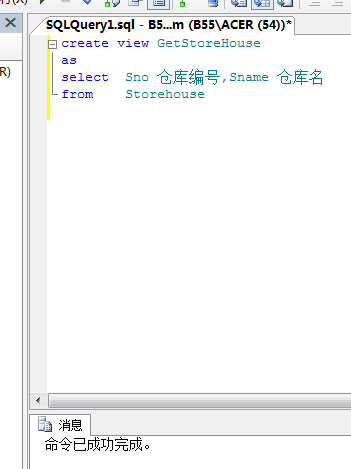
1. 创建报表；



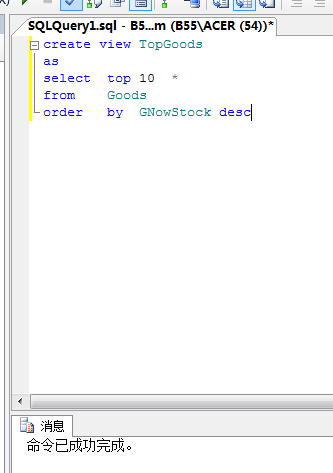
### 5.1.2 视图的实现

通过视图修改数据，可以在一定程度上保证数据库的安全性，使没有合法权限的用户无法访问核心数据。

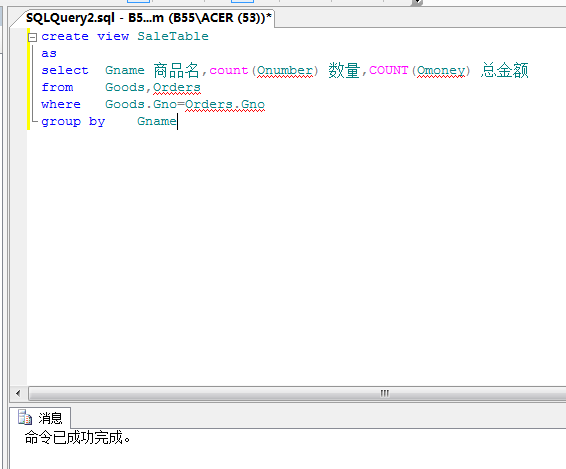
1. 创建GetStoreHouse视图；



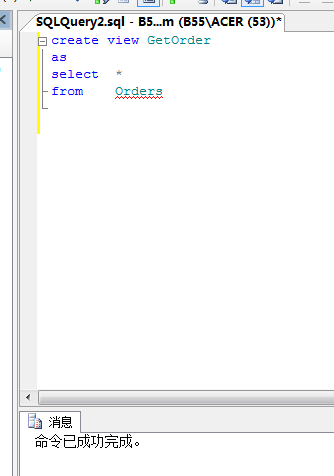
1. 创建TopGoods视图



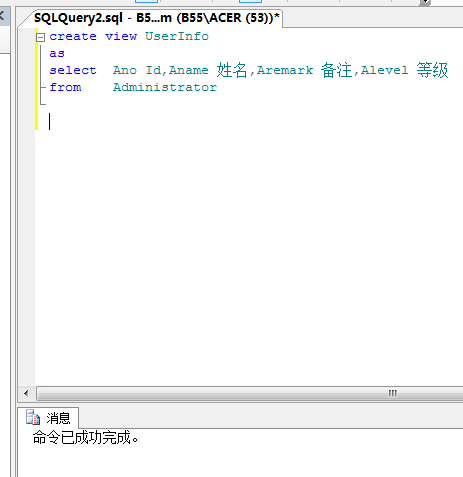
1. 创建SaleTable视图



1. 创建GetOrder视图



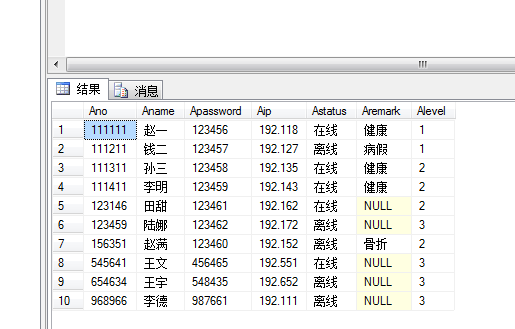
1. 创建UserInfo视图



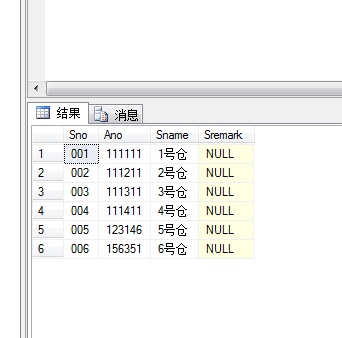
### 5.1.3 数据的录入

根据关系模式在数据库里创建好各个表之后，开始录入数据，每个表不少于10条记录（仓库表除外）。

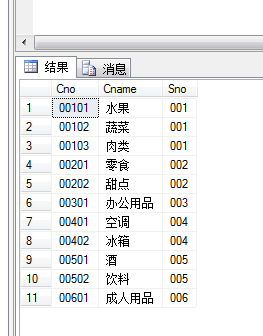
1. Administrator表数据的录入；



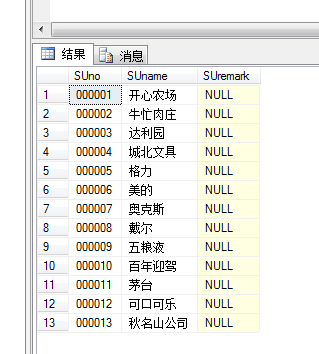
1. Storehouse表数据的录入；



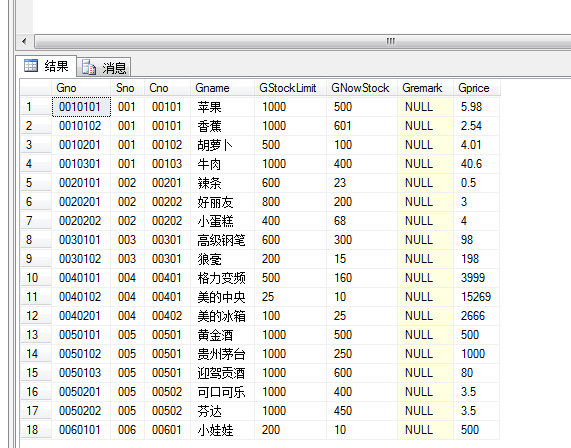
1. Classification表数据的录入；



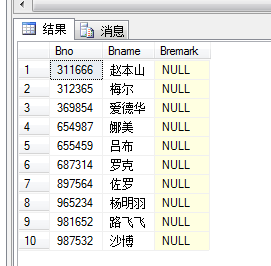
1. Supplier表数据的录入；



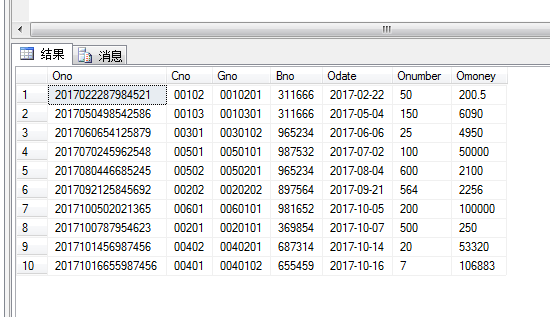
1. Goods表数据的录入；



1. Buyer表数据的录入；



1. Order表数据的录入；



## 5.2建立触发器、索引

（1）创建触发器----商品库存少于10%时自动生成订单。

（2）创建触发器----当相应物品的订单进过采购员同意后，物品表备注改为已记录。

（3）创建索引----商品库存少于10%的商品排在库存多于10%的商品的前面。

# 第六章 测试结果

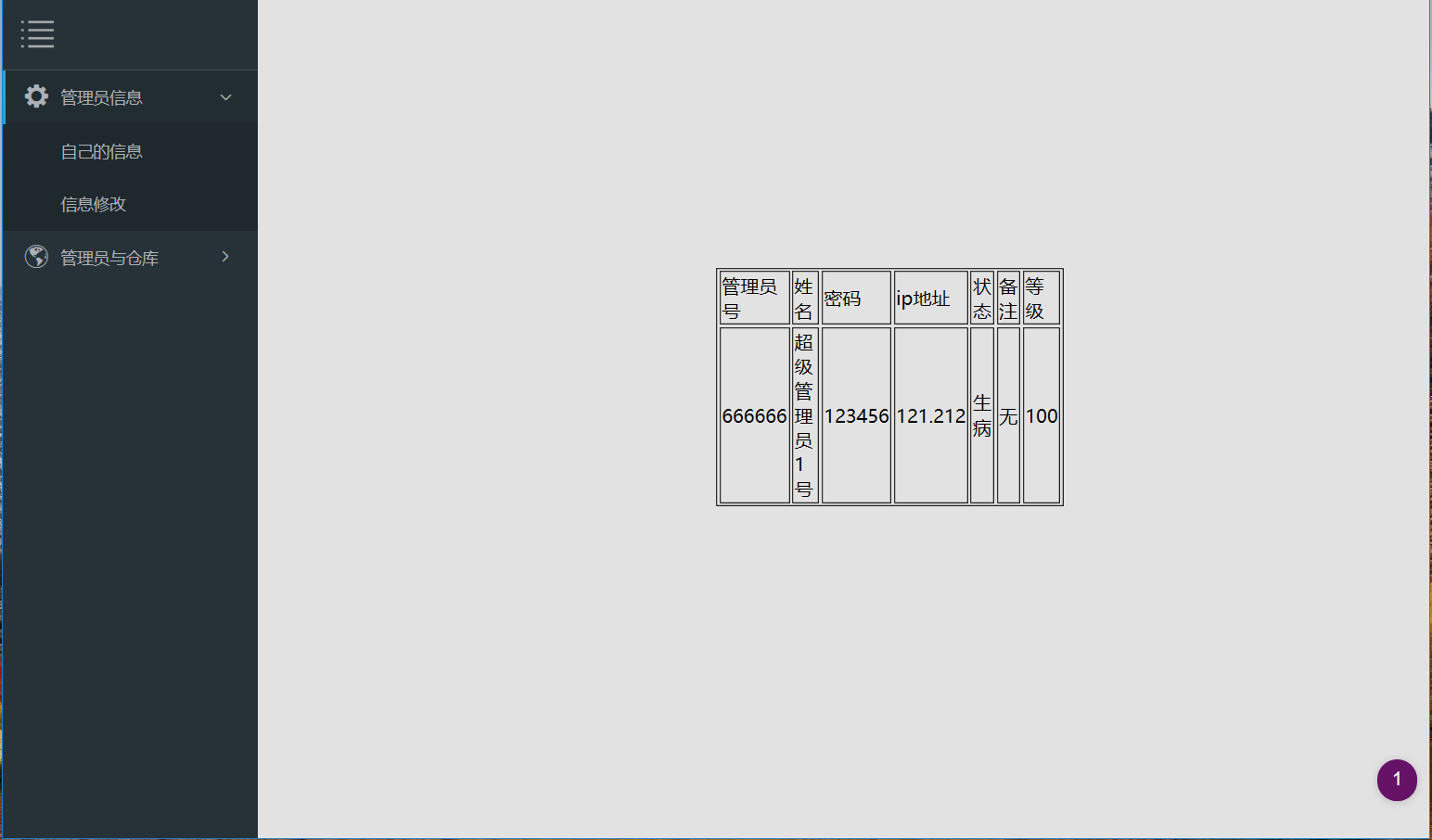
## 6.1界面设计

### 6.1.1 登陆界面

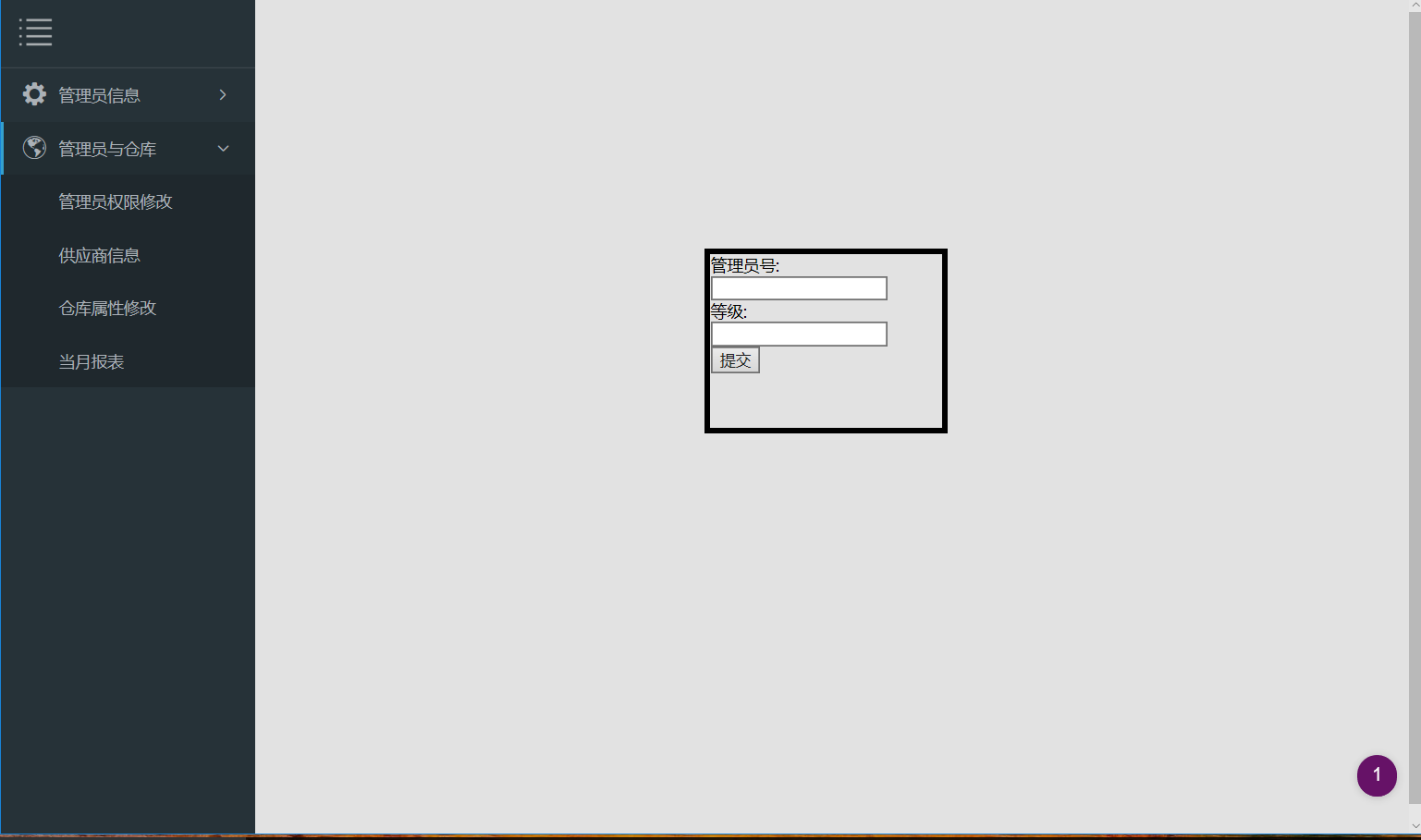


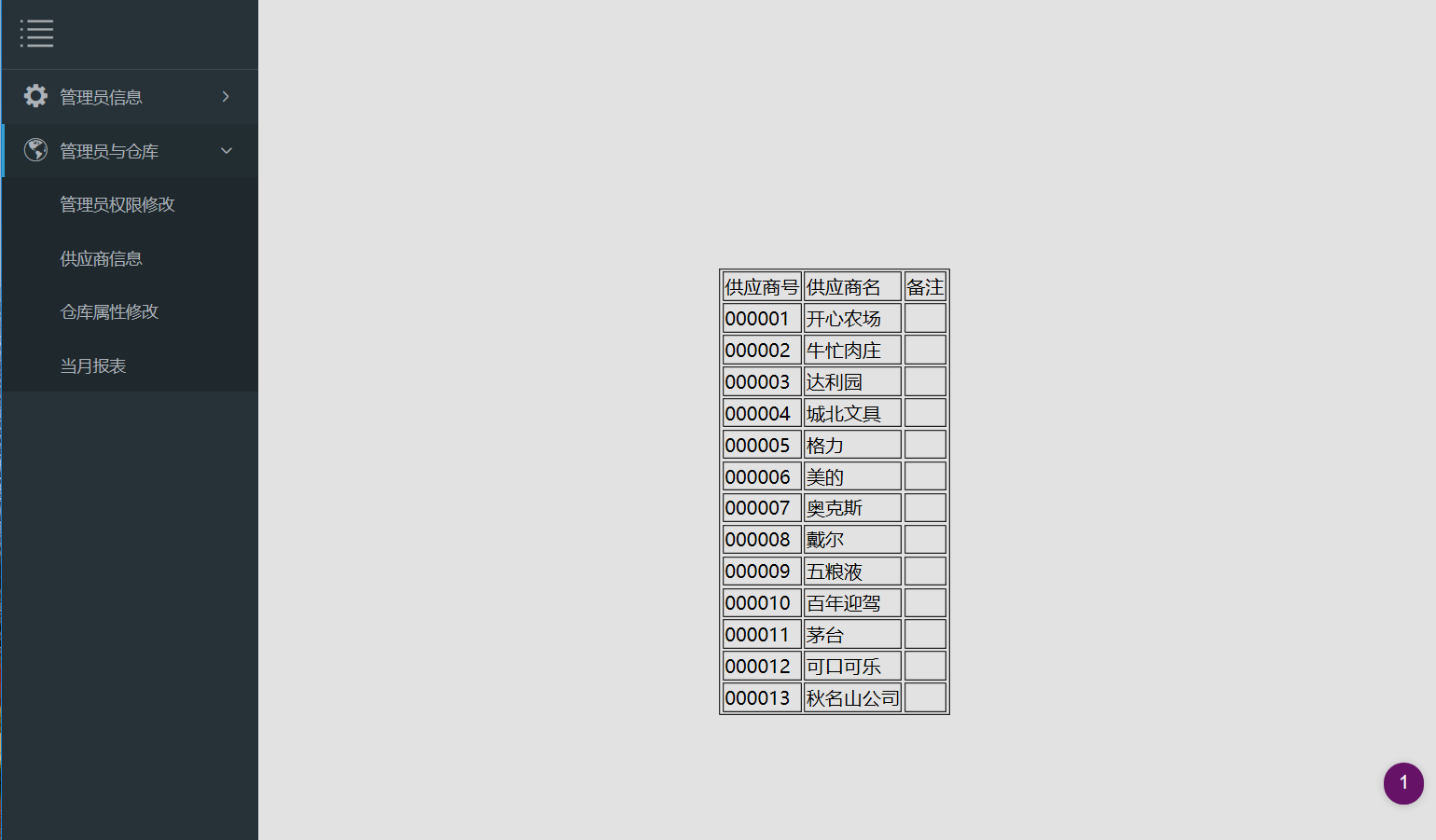


### 6.1.2 超级管理员界面

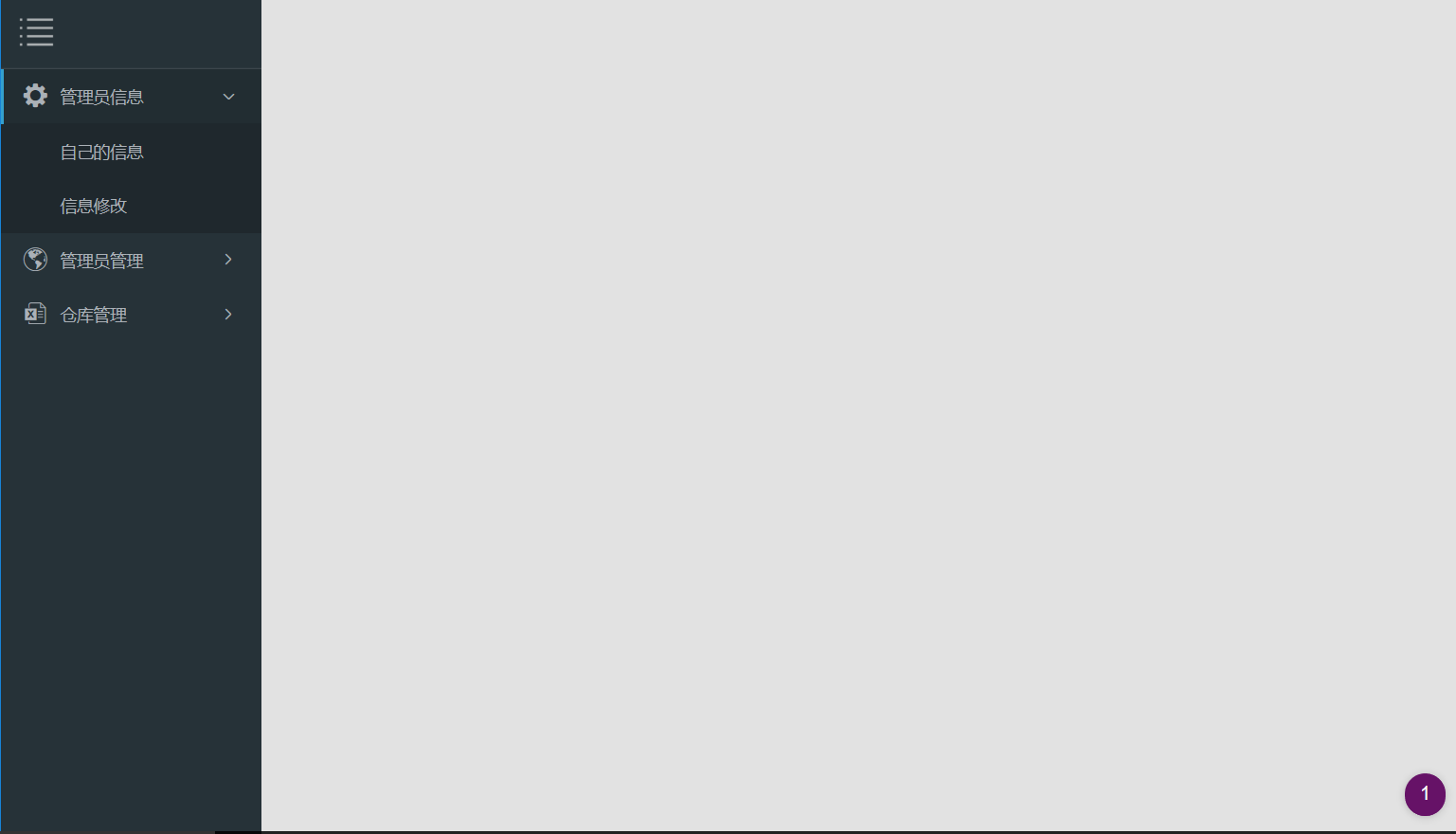


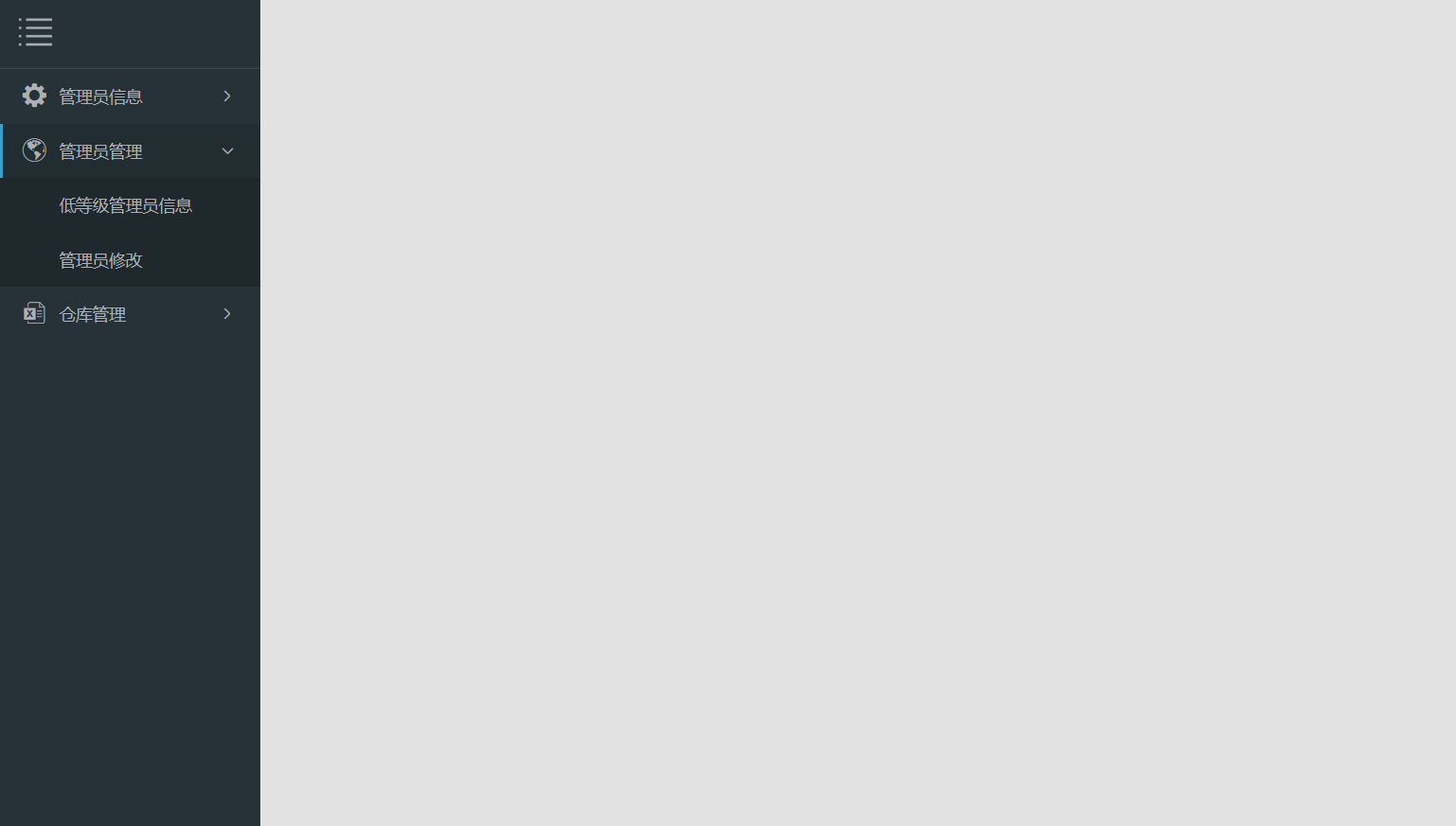


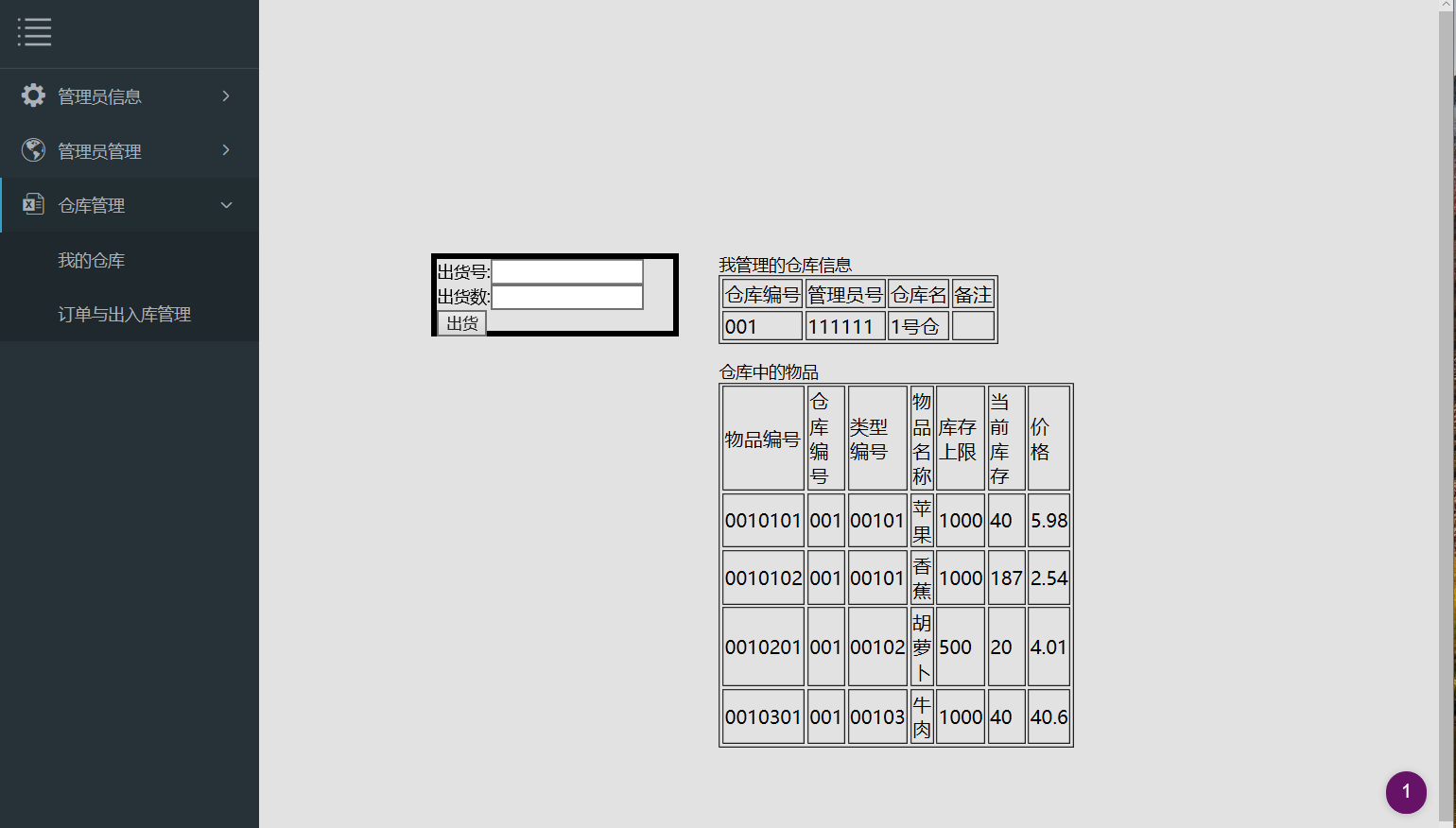


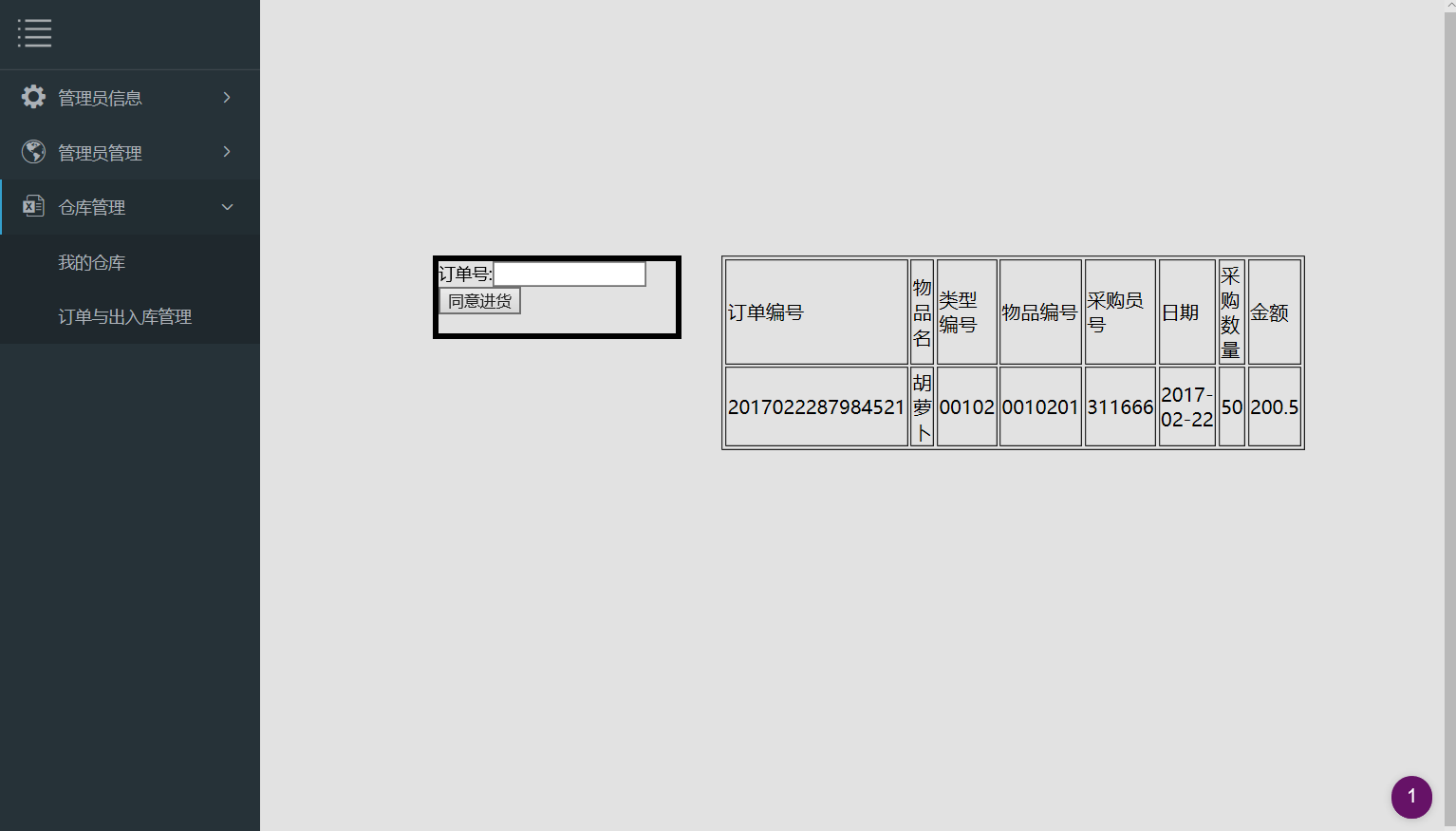


### 6.1.3 普通管理员界面

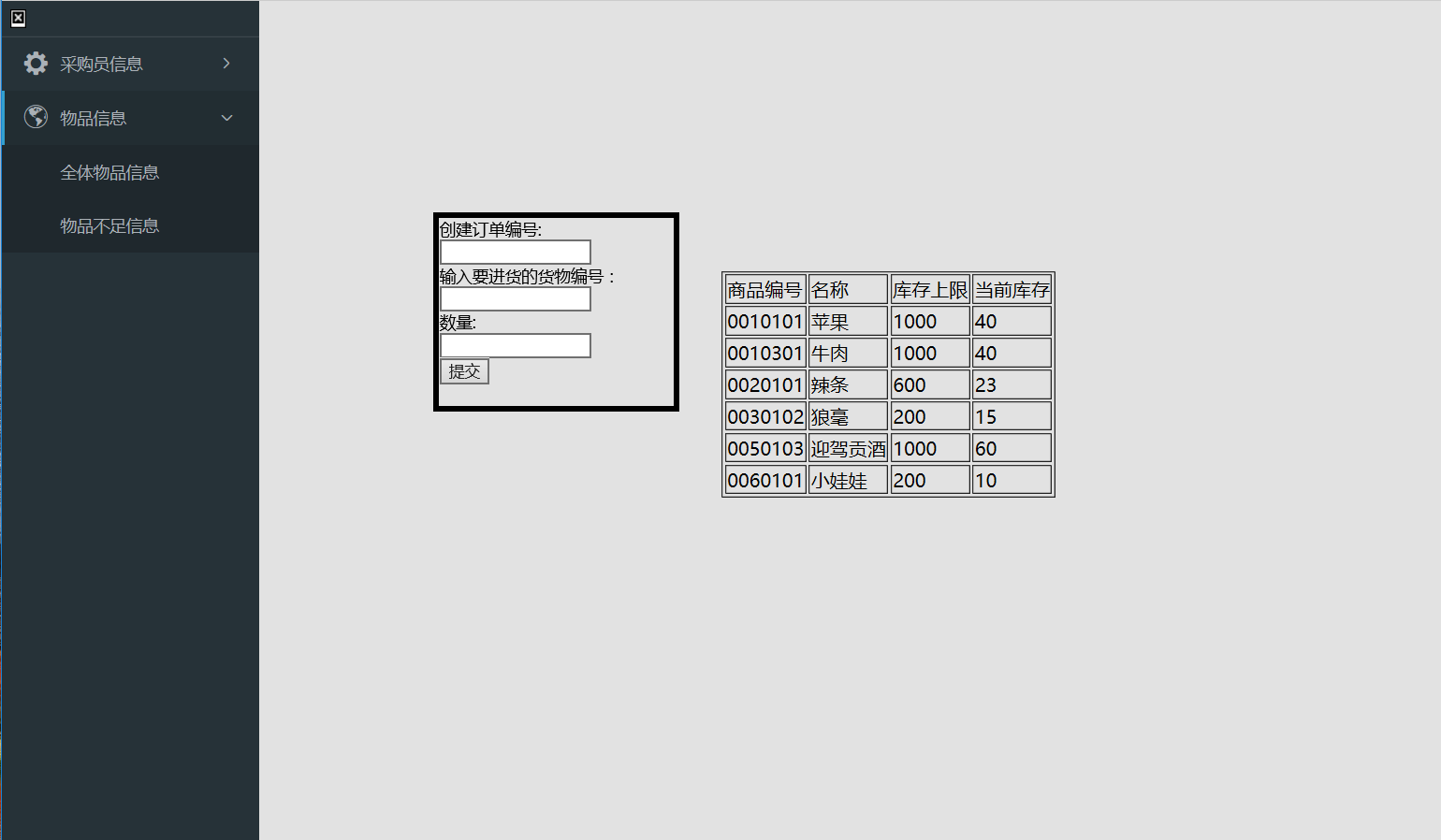
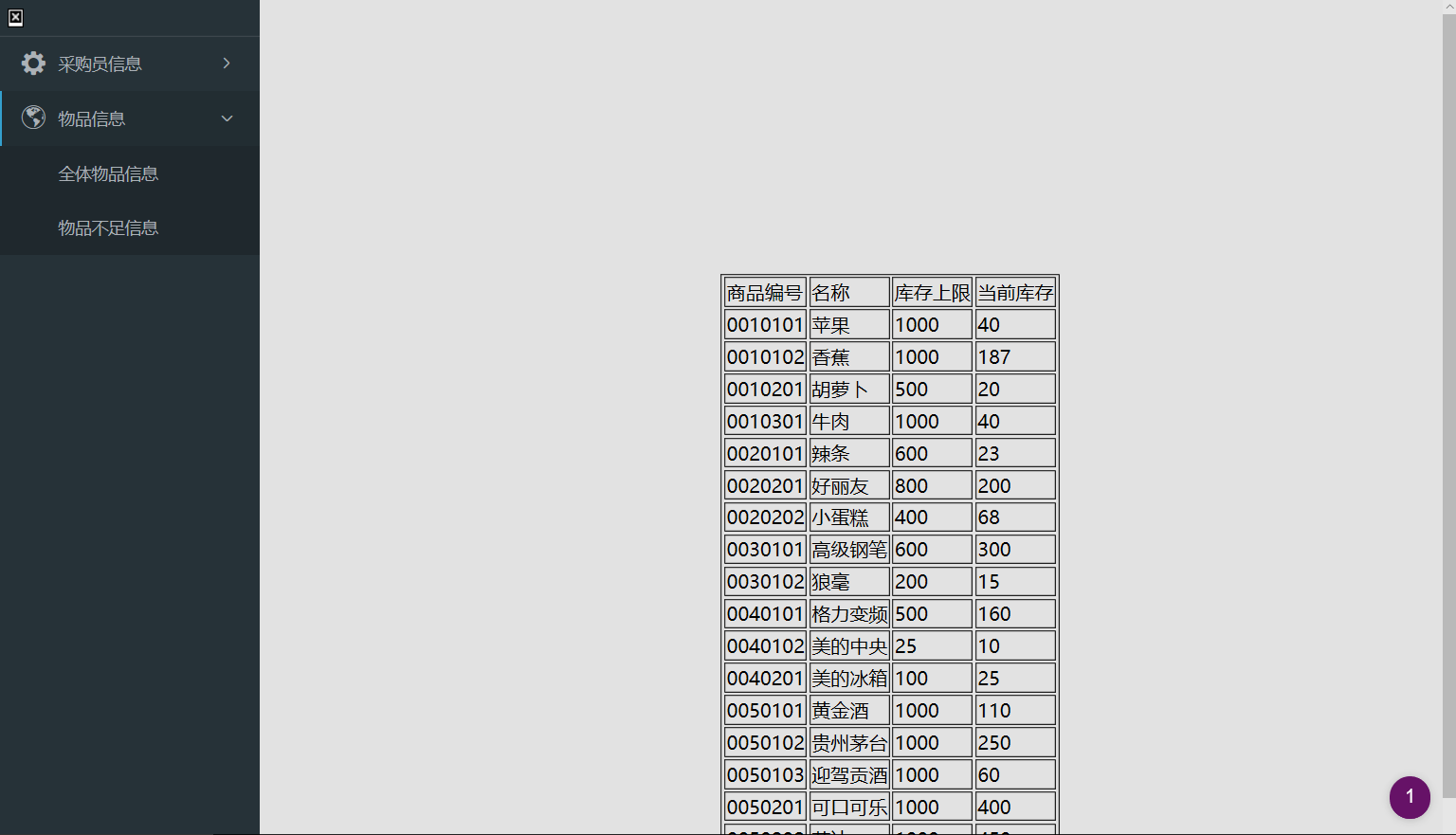
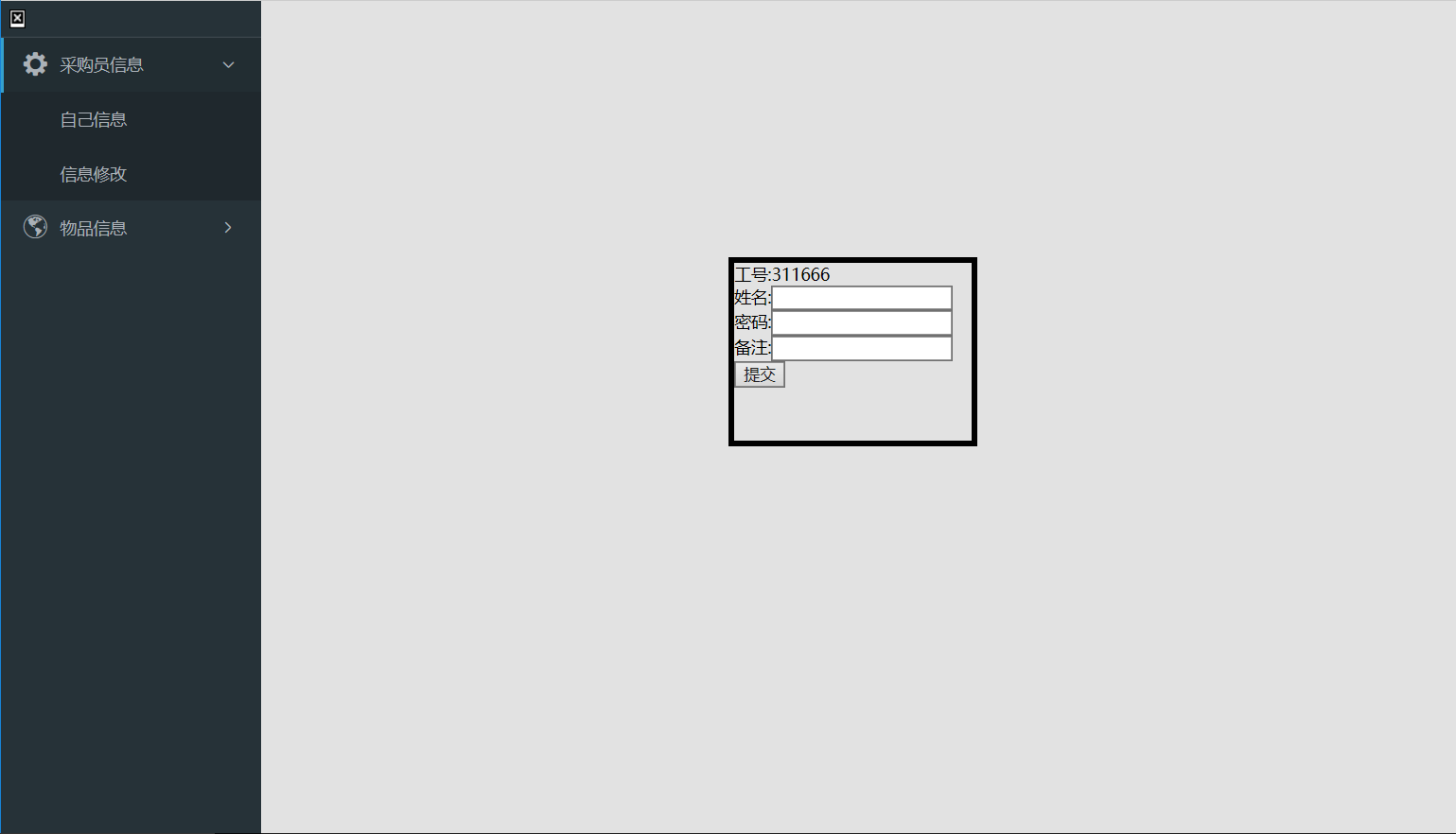








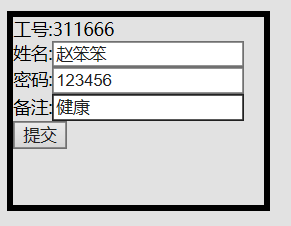
### 6.1.4 采购员界面



## 6.2操作流程

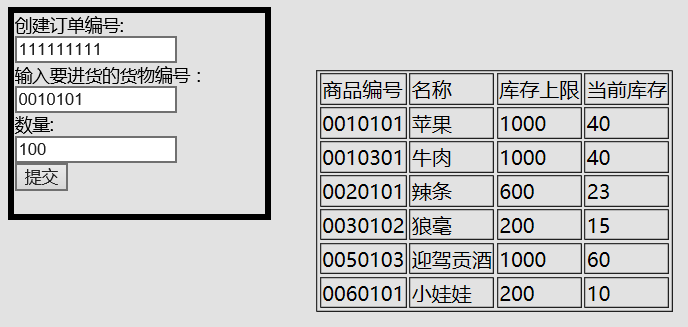
### 6.2.1 信息修改





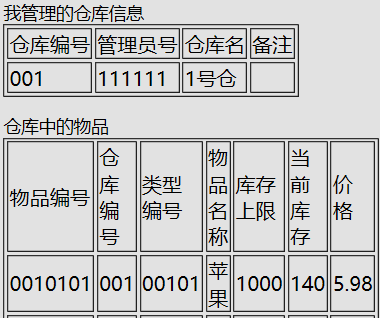


### 6.2.2 创建订单

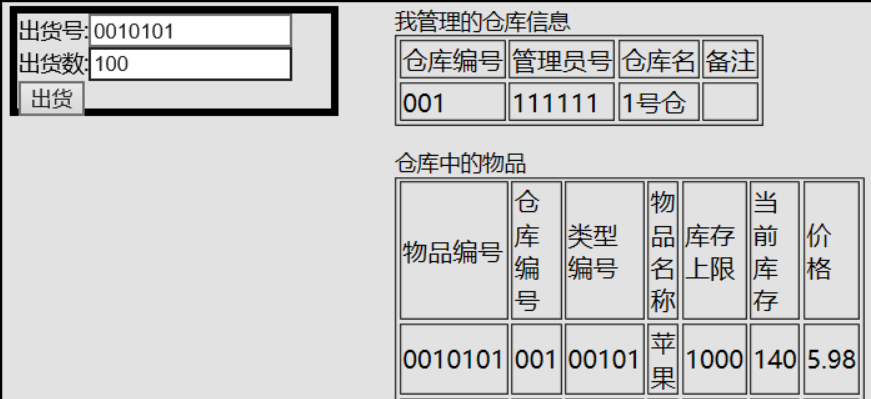


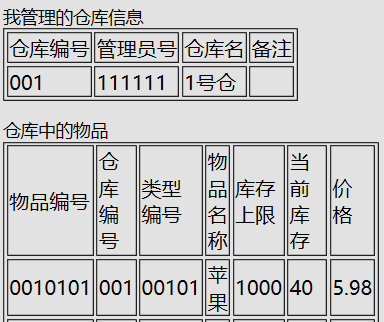
### 6.2.3 进库



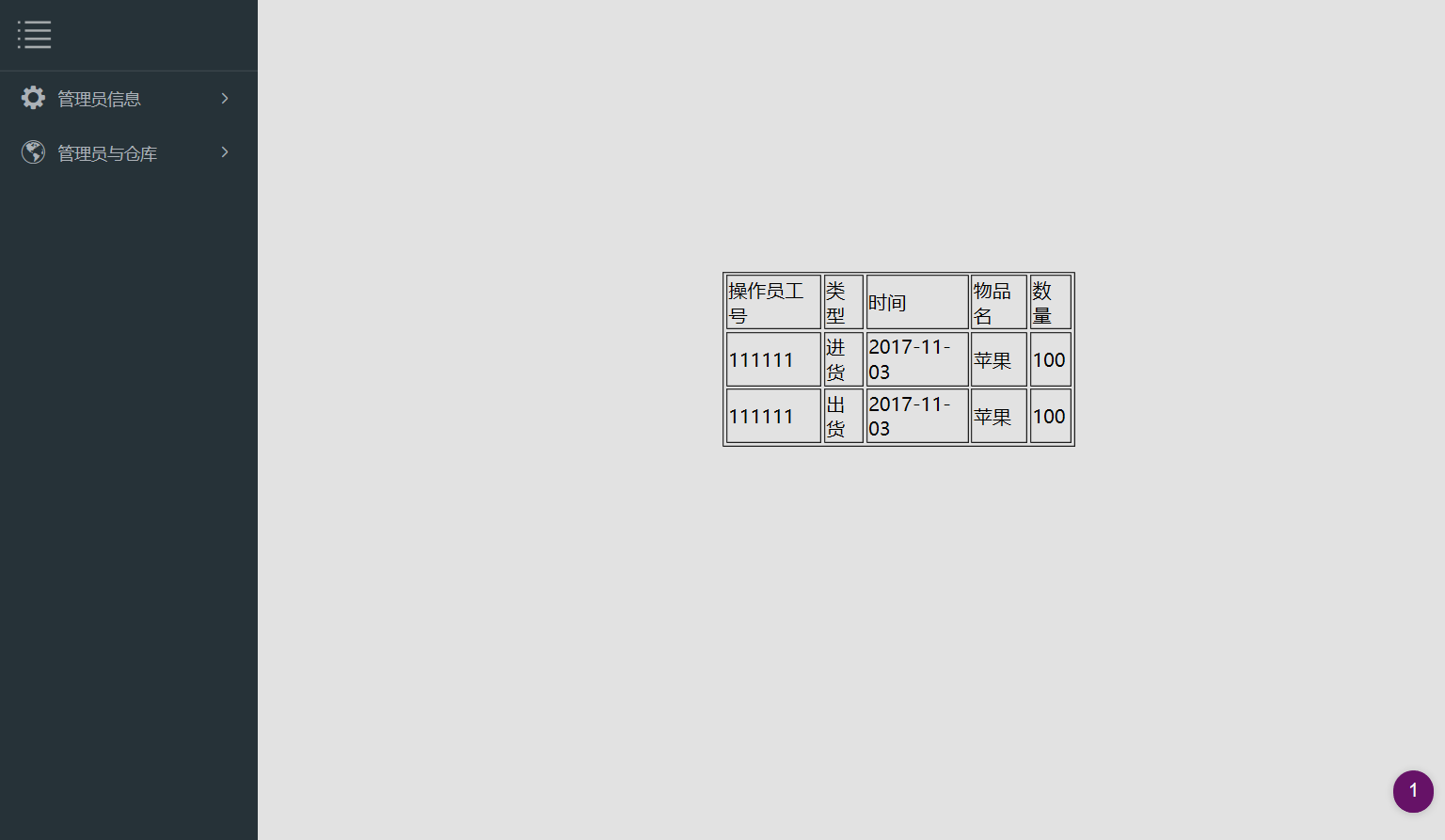


### 6.2.4 出库





### 6.2.5 当月报表



# 第七章 总结

## 7.1任务分工

### 邹梁琦

在本次数据库课程设计中，我主要负责工作分工、调度、并参与了课程设计的各个部分，重点参与了数据库的建立以及界面设计，同时对仓储管理系统进行了测试。

在数据库的实施阶段，我负责修改数据库的不足之处，同时优化数据库。在界面设计中，我负责代码设计的一部分，同时参与各个设计阶段的整体优化。在需求分析阶段，我负责了数据库的系统需求、数据需求、处理需求、安全性需求、完整性需求。在概念设计阶段，我负责的是实体之间的联系设计。在逻辑设计阶段，我负责了概念模型向关系模型的转化。

### 张宝根

我在团队中负责的工作主要有逻辑设计、数据库的实施。

在逻辑设计阶段，我负责对概念模型进行优化，并完成了数据库视图的设计。

在数据库实施阶段，我负责对数据库进行错误修改、补正。

在功能实现阶段，我写了功能实现的所有代码。

### 许哲龙

在本学期数据库课程设计中，我主要负责需求分析、概念设计、逻辑设计、物理设计、以及数据库的建立五个模块。

在需求分析阶段，我负责画数据流图和数据字典。在概念设计阶段，我负责画E-R图。

在逻辑设计阶段，我负责设计数据库的表。在物理设计阶段，我负责设计数据库的存储结构和存储方法。在数据库的实施阶段，我负责编写数据库表的代码。

### 张银磊

本人参与数据的定义，数据库的创建。主要负责数据库和表的建立，视图的实现，数据的录入，建立触发器、存储过程、索引，数据库测试、更新等。并且参与查找与登录界面设计的相关代码的工作中。

## 7.2个人总结

### 邹梁琦

完成这个超小型的仓储管理系统之后，的确有很多的感想。最深的感触就是：学习专业知识，必须去实践，否则一切学习只能是纸上谈兵。

这次课程设计给我最大的感受就是做一个领导者是很不容易的。在每次实验中，我都努力为团队分配各种任务，同时还要做好宏观的调控。这次课程设计是一次很好的锻炼，相信以后再遇到这样的项目时我一定能更从容地应对。

### 张宝根

在此期间，我碰到了很多的问题，通过小组的讨论与上网查询，我解决了一系列的问题，熟悉了网页界面的设计与php的运用，收获颇丰，其次，我对于数据库设计过程中的需求分析有了深刻的认识，在设计数据库时一定要确定好完整的数据库需求。

### 许哲龙

通过本次课程设计，我感受到了团队工作的力量。在课程设计的一开始，我认为这是一个巨大的工程，很难完成。但是，在实验的过程中，通过大家的不断努力，通过分工，我们发现每个人的工作量不是特别多。总之，这次课程设计非常有意义，我学到了很多知识，也学会了如何去合作完成一个较大的项目。

### 张银磊

通过九次课的工作，我们四人成功完成了仓库管理系统的数据库设计，四人分工明确，合作进行数据库的设计，在完成基础任务的基础上，添加了属于我们团队的创新点。总之，通过本次设计，对于数据库这门学科有了近一步的认识和掌握，丰富了自己的知识体系，有了很大的收获。

## 7.3团队总结

首先：开始的时候，我总是在想我们的学习还是不够，这个仓库管理系统该怎么弄呢？总是徘徊之中，不知道从何处下手。

然后：经过讨论以及根据每个人的能力状况我们选择了做网页端的仓储管理系统，并开始了对这个仓库管理系统的思考，首先我对这个系统的界面进行了一定的思考。也查询了许多的学习资料。

与此同时复习的数据库理论，已经有一定的火侯了。通过数据库的理论学习中，学到了E-R图，数据库的设计过程：需求分析，概要设计，逻辑设计，详细设计，数据库的维护。

这个系统只是我们一个小小的尝试，是一个学习的过程，如果要真正的完善这个系统还有许多的事情要去做，还需继续努力。