《数据库系统概论》

课程设计报告

《电脑销售信息管理系统》

班级：

学号：

姓名：

目 录

[一、 系统需求分析 3](#_Toc517822238)

[（一） 需求概述 3](#_Toc517822239)

[（二） 业务流分析 3](#_Toc517822240)

[（三） 数据流分析 5](#_Toc517822246)

[（四） 数据字典 6](#_Toc517822248)

[二、 数据库概念结构设计 9](#_Toc517822249)

[（一） 实体分析 9](#_Toc517822250)

[（二） 属性分析 9](#_Toc517822251)

[（三） 联系分析 13](#_Toc517822252)

[（四） 概念模型分析（.CDM图） 14](#_Toc517822253)

[三、数据库逻辑结构设计 15](#_Toc517822254)

[（一）概念模型转化为逻辑模型 15](#_Toc517822255)

[（二）逻辑模型设计（.PDM图） 15](#_Toc517822263)

[四、数据库物理实现 16](#_Toc517822265)

[（一） 表设计 16](#_Toc517822266)

[（二） 创建表和完整性约束代码设计 19](#_Toc517822279)

[（三） 创建视图、索引、存储过程和触发器 22](#_Toc517822292)

[1. 创建视图 22](#_Toc517822293)

[2. 创建索引 22](#_Toc517822297)

[3. 创建存储过程 23](#_Toc517822299)

[五、数据库功能调试 24](#_Toc517822302)

[（一） 采购员信息查询模块 24](#_Toc517822303)

[（二） 仓库管理员信息查询模块 25](#_Toc517822308)

[（三） 营业员信息查询模块 26](#_Toc517822319)

[（四） 系统管理员管理模块 27](#_Toc517822328)

[六、设计系统前台软件 38](#_Toc517822341)

[（一）开发软件选择 39](#_Toc517822342)

[（二）软件功能要求与设计 40](#_Toc517822344)

[（三）软件功能实现 40](#_Toc517822345)

[（四）系统测试 47](#_Toc517822346)

[七、设计总结 50](#_Toc517822347)

# 系统需求分析

1. 需求概述

本次课程设计的题目是电脑销售信息管理系统，系统要求做到电脑销售信息系统中的一系列的信息管理：商品的采购、进货、配送、销售和存储。完成这五项销售信息的管理还需要有商品、供应商、仓库、门店、采购员、管理员和营业员的基本信息。做到可以对商品的流动信息以及7个实体的信息进行修改、查询。

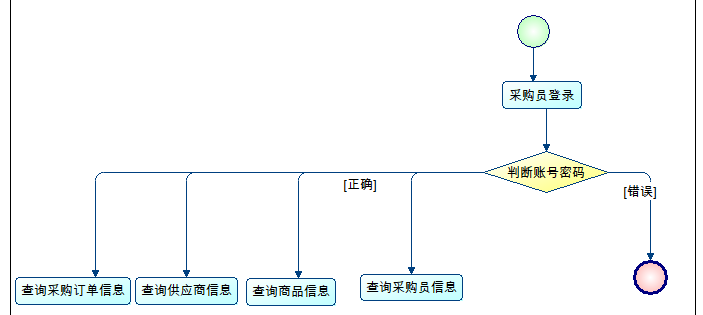
系统的使用者分为管理人员和普通用户。他们对该数据库系统有不同的操作：

管理人员：可以对供货商、仓库、门店、商品进行删除或者增添，也可以更改它们的信息，如更改供货商的地址，更改商品单位等；可以在对采购员、管理员、营业员进行删除或者增添，也可以更改其中的信息，比如，更改管理员管理的仓库，修改采购员姓名等；可以增添删除和修改采购、进货、配送、销售和存储订单，可以增添订单单号，修改订单日期、数量等等。总之，管理人员可以对数据库系统进行增加、删除、更新、查询各种操作。

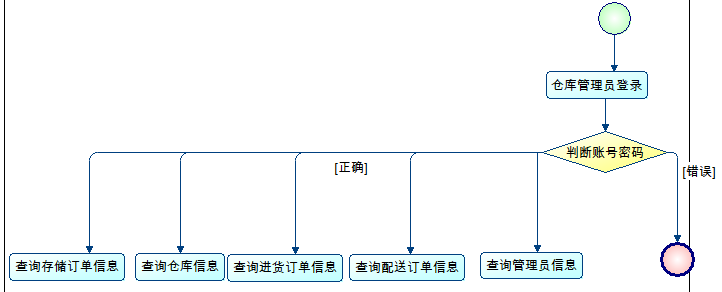
普通用户：采购员可以查询采购订单信息、供货商信息、商品信息；仓库管理员可以查询存储订单信息和仓库信息以及进货订单息和配送订单信息；营业员可以查询门店信息、销售订单信息和配送订单信息；总之，普通用户可以对数据库系统进行查询操作。

1. 业务流分析

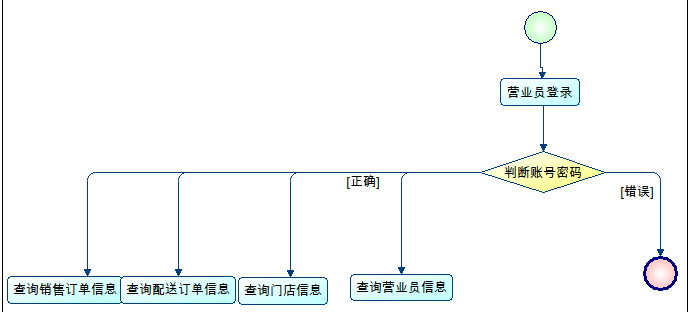
从采购员角度：



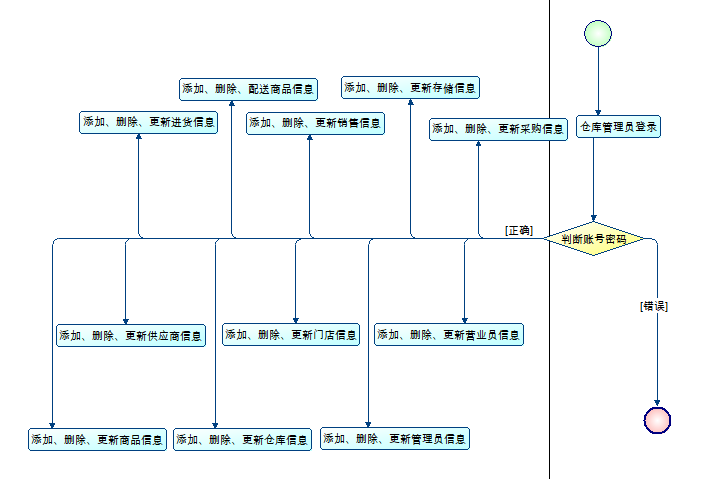
从仓库管理员角度：



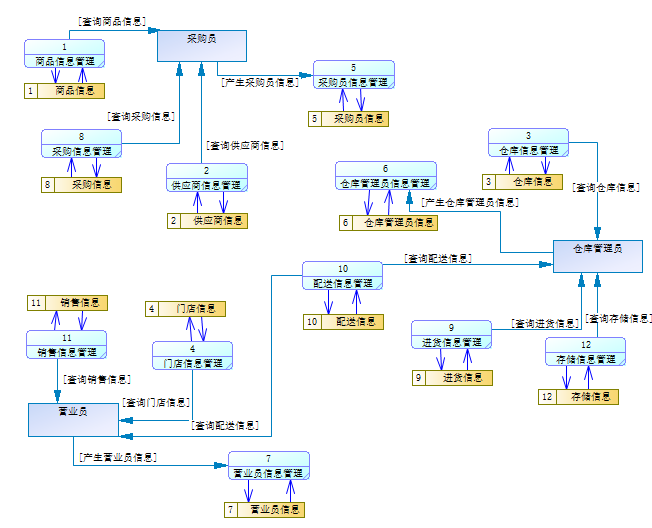
从营业员角度：



从管理员角度：



1. 数据流分析



1. 数据字典

数据项：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 含义说明 | 数据类型 | 长度 | 取值范围 | 数据项联系 |
| 数据项名 | 含义说明 | 数据类型 | 长度 | 取值范围 | 数据项联系 |
| 商品编号 | 区别商品 | char | 4 | 0000-9999 | 一个商品对应一个商品名 |
| 商品名称 | 区别商品 | varchar | 20 |  |  |
| 类别 | 标注商品 | varchar | 10 |  |  |
| 单位 | 标注商品 | varchar | 4 |  |  |
| 单价 | 标注商品，每件商品的价格 | char | 4 | 0-9999 |  |
| 供应商编号 | 区分供应商 | char | 4 | 0000-9999 | 一个供应商对应一个供应商号 |
| 供应商名称 | 区分供应商 | varchar | 20 |  |  |
| 账号 | 标注供应商 | char | 7 | 0000000-9999999 |  |
| 供应商地址 | 标注供应商 | varchar | 20 |  |  |
| 仓库编号 | 区分仓库 | char | 4 | 0000-9999 | 一个仓库对应一个仓库编号 |
| 仓库地址 | 标注仓库，仓库所在地址 | varchar | 20 |  |  |
| 负责人 | 标注仓库 | varchar | 8 |  |  |
| 门店编号 | 区别门店 | char | 4 | 0000-9999 | 一个门店对应一个门店编号 |
| 门店名称 | 区别门店 | varchar | 20 |  |  |
| 门店地址 | 标注门店 | varchar | 20 |  |  |
| 采购员编号 | 区别采购员 | char | 4 | 0000-9999 | 一个采购员对应一个采购员编号 |
| 采购员姓名 | 标注采购员 | varchar | 8 |  |  |
| 采购员业绩 | 标注采购员 | int |  |  |  |
| 管理员编号 | 区别管理员 | char | 4 | 0000-9999 | 一个管理员对应一个管理员编号 |
| 管理员姓名 | 标注管理员 | varchar | 8 |  |  |
| 管理员业绩 | 标注管理员 | int |  |  |  |
| 营业员编号 | 区别营业员 | char | 4 | 0000-9999 | 一个营业员对应一个营业员编号 |
| 营业员姓名 | 标注营业员 | varchar | 8 |  |  |
| 营业员业绩 | 标注营业员 | int |  |  |  |
| 采购单号 | 区别采购关系 | char | 7 |  | 一个采购订单对应一个采购单号 |
| 采购数量 |  | int |  | 1-99999 |  |
| 采购日期 |  | datetime |  |  |  |
| 进货单号 | 区别进货关系 | char | 7 |  | 一个进货订单对应一个进货单号 |
| 进货数量 |  | int |  | 1-99999 |  |
| 进货日期 |  | datetime |  |  |  |
| 配送单号 | 区别配送关系 | char | 7 |  | 一个配送订单对应一个配送单号 |
| 配送数量 |  | int |  | 1-99999 |  |
| 配送日期 |  | datetime |  |  |  |
| 销售单号 | 区别销售关系 | char | 7 |  | 一个销售订单对应一个销售单号 |
| 销售数量 |  | int |  | 1-99999 |  |
| 销售日期 |  | datetime |  |  |  |
| 存储日期 | 区别存储关系 | datetime |  |  |  |
| 库存量 | 当前库存量 | int |  | 0-99999 |  |
| 安全库存量 |  | int |  | 0-99999 |  |

数据结构：

|  |  |  |
| --- | --- | --- |
| 数据结构名 | 含义名 | 组成 |
| 商品 | 商品信息 | 商品编号，名称，类别，单位，单价 |
| 供应商 | 供应商信息 | 供应商编号，名称，账号，地址 |
| 仓库 | 仓库信息 | 仓库编号，地址，负责人 |
| 门店 | 门店信息 | 门店编号，名称，地址 |
| 采货员 | 采货员信息 | 采购员编号，姓名，业绩 |
| 管理员 | 管理员信息 | 管理员编号，姓名，业绩 |
| 营业员 | 营业员信息 | 营业员编号，姓名，业绩 |
| 采购 | 采购员从供应商采购商品 | 采购单号，数量，日期，采购员编号，供应商编号，商品编号 |
| 进货 | 供应商从仓库购进商品 | 进货单号，数量，日期，供应商编号，商品编号，仓库编号 |
| 配送 | 商品由仓库被配送到门店 | 配送单号，数量，日期，商品编号，仓库编号，门店编号 |
| 销售 | 营业员在门店中销售商品 | 销售单号，数量，日期，商品编号，门店编号，营业员编号 |
| 存储 | 商品在仓库里的存储情况 | 商品编号，仓库编号，日期，库存量，安全库存量 |

数据流：

|  |  |  |  |
| --- | --- | --- | --- |
| 数据流名 | 数据流来源 | 数据流去向 | 组成 |
| 商品信息管理 | 各个商品的信息 | 商品信息表 | 商品信息 |
| 供应商信息管理 | 供应商 | 供应商信息表 | 供应商信息 |
| 仓库信息管理 | 仓库 | 管理员 | 仓库信息 |
| 门店信息管理 | 门店 | 营业员 | 门店信息 |
| 采购员信息管理 | 采购员 | 采购员信息表 | 采购员信息 |
| 管理员信息管理 | 仓库信息表 | 管理员信息表 | 管理员信息 |
| 营业员信息管理 | 门店信息表 | 营业员信息表 | 营业员信息 |
| 采购信息管理 | 采购员信息表、供应商信息表、商品 信息表 | 采购信息表 | 采购信息 |
| 进货信息管理 | 供应商信息表、商品信息表 | 仓库信息表 | 进货信息 |
| 配送信息管理 | 商品信息表、仓库信息表 | 门店信息表 | 配送信息 |
| 销售信息管理 | 商品信息表、门店信息表 | 营业员信息表 | 销售信息 |
| 存储信息管理 | 商品信息表、仓库信息表 | 存储信息表 | 存储信息 |

数据存储：

|  |  |  |  |
| --- | --- | --- | --- |
| 数据存储名 | 输入的数据流 | 输出的数据流 | 组成 |
| 管理员信息表 | 管理员信息  仓库信息 | 管理员信息 | 管理员信息  仓库信息 |
| 营业员信息表 | 营业员信息  门店信息 | 营业员信息 | 营业员信息  门店信息 |
| 商品采购登记表 | 采购员信息  供应商信息  商品信息 | 采购信息 | 采购员信息  供应商信息  商品信息 |
| 商品进货登记表 | 供应商信息  商品信息 | 仓库信息 | 仓库信息  供应商信息  商品信息 |
| 商品配送登记表 | 商品信息  仓库信息 | 门店信息 | 门店信息  商品信息  仓库信息 |
| 商品销售登记表 | 商品信息  门店信息 | 营业员信息 | 门店信息  商品信息  营业员信息 |
| 商品存储登记表 | 商品信息  仓库信息 | 存储信息 | 存储信息  商品信息  仓库信息 |

# 数据库概念结构设计

1. 实体分析

经需求分析，本次课程设计中包含7个实体，他们分别是：商品实体、供应商实体、仓库实体、门店实体、采购员实体、管理员实体、营业员实体。

1. 属性分析

商品实体属性：商品编号，名称，类别，单位，单价。如下：

供应商实体属性：供应商编号，名称，账号，地址。如下：



仓库实体属性：仓库编号，地址，负责人。如下：



门店实体属性：门店编号，名称，地址。如下：



采货员实体属性：采购员编号，姓名，业绩。如下：



管理员实体属性：管理员编号，姓名，业绩。如下：



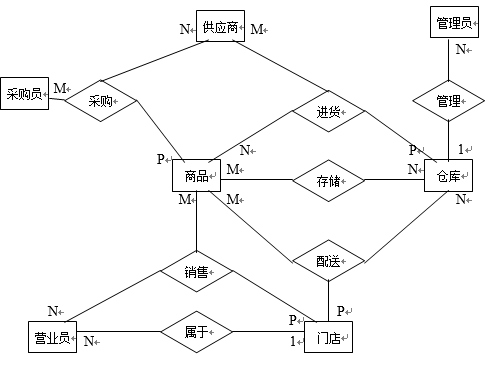
营业员实体属性：营业员编号，姓名，业绩。如下：



1. 联系分析

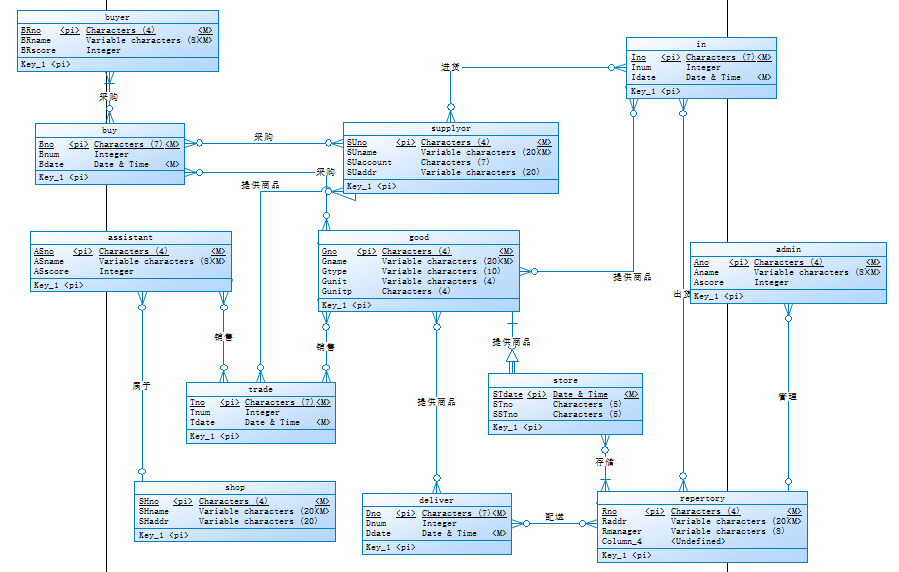
一个管理员只管理一个仓库，但一个仓库有多个管理员，管理员与仓库之间的联系是多对一的联系；一个营业员属于一个门店，一个门店有多个营业员，营业员与门店是多对一的关系；一个采购员能从多个供应商里采购多种商品，一个供应商能被多个采购员采购，一件商品也能被多个采购员采购，采购员与供应商、商品存在M:N:P的关系；一个供应商能从多个供仓库里购进多种商品，一个仓库能为多个供应商提供商品，一件商品也能被多个供应商购进，供应商与仓库、商品存在M:N:P的关系；一个门店能从多个仓库获得多种商品配送，一个仓库能为多个门店配送，一件商品也能被配送到多个门店，采购员与供应商、商品存在M:N:P的关系；一个采购员能从多个供应商里采购多种商品，一个供应商能被多个采购员采购，一件商品也能被多个采购员采购，采购员与供应商、商品存在M:N:P的关系；

E-R图分析如下：



1. 概念模型分析（.CDM图）

在经过需求分析和实体属性的分析，以及各实体之间的关系，最终得到概念模型如下：



# 三、数据库逻辑结构设计

（一）概念模型转化为逻辑模型

1. 一对一关系的转化

在电脑销售信息管理系统中没有一对一关系的转化

1. 一对多关系的转化

一个管理员只管理一个仓库，但一个仓库有多个管理员，管理员与仓库之间的联系是多对一的联系

一个营业员属于一个门店，一个门店有多个营业员，营业员与门店是多对一的关系；

1. 多对多关系的转化

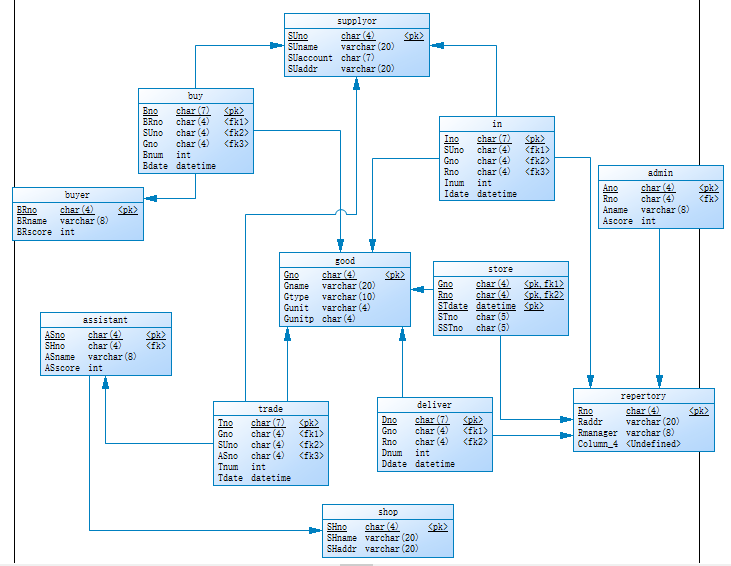
一个采购员能从多个供应商里采购多种商品，一个供应商能被多个采购员采购，一件商品也能被多个采购员采购，采购员与供应商、商品存在M:N:P的关系；

一个供应商能从多个供仓库里购进多种商品，一个仓库能为多个供应商提供商品，一件商品也能被多个供应商购进，供应商与仓库、商品存在M:N:P的关系；

一个门店能从多个仓库获得多种商品配送，一个仓库能为多个门店配送，一件商品也能被配送到多个门店，采购员与供应商、商品存在M:N:P的关系；

一个采购员能从多个供应商里采购多种商品，一个供应商能被多个采购员采购，一件商品也能被多个采购员采购，采购员与供应商、商品存在M:N:P的关系；

（二）逻辑模型设计（.PDM图）



# 四、数据库物理实现

1. 表设计
2. 商品信息表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Gno | 商品编号 | char(4) | 主键 |
| Gname | 商品名称 | varchar(20) | 非空 |
| Gtype | 类别 | varchar(10) |  |
| Gunit | 单位 | Varchar(4) |  |
| Gunitp | 单价 | char(4) |  |

1. 供应商信息表设计：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| SUno | 供应商编号 | char(4) | 主键 |
| SUname | 供应商名称 | varchar(20) | 非空 |
| SUaccount | 账号 | char(7) |  |
| SUaddr | 供应商地址 | varchar(20) |  |

1. 仓库信息表设计：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Rno | 仓库编号 | char(4) | 主键 |
| Raddr | 仓库地址 | varchar(20) | 非空 |
| Rmanager | 负责人 | varchar(8) |  |

1. 门店信息表设计：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| SHno | 门店编号 | char(4) | 主键 |
| SHname | 门店名称 | varcShar(20) | 非空 |
| SHaddr | 地址 | varchar(20) |  |

1. 采购员信息表设计：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| BRno | 采购员编号 | char(4) | 主键 |
| BRname | 采购员姓名 | varchar(8) | 非空 |
| BRscore | 采购员业绩 | int |  |

1. 管理员信息表设计：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Ano | 管理员编号 | char(4) | 主键 |
| Aname | 管理员姓名 | varchar(8) | 非空 |
| Ascore | 管理员业绩 | int |  |
| Rno | 仓库编号 | Char(4) | 外键 |

1. 营业员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| ASno | 营业员编号 | char(4) | 主键 |
| ASname | 营业员姓名 | varchar(8) | 非空 |
| ASscore | 营业员业绩 | int |  |
| SHno | 门店编号 | Char(4) | 外键 |

1. 采购关系表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Bno | 采购单号 | char(7) | 主键 |
| Bnum | 采购数量 | int |  |
| Bdate | 采购日期 | datetime | 非空 |
| BRno | 采购员编号 | char(4) | 外键 |
| SUno | 供应商编号 | char(4) | 外键 |
| Gno | 商品编号 | Char(4) | 外键 |

1. 进货关系表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Ino | 进货单号 | char(7) | 主键 |
| Inum | 进货数量 | int |  |
| Idate | 进货日期 | datetime | 非空 |
| SUno | 供应商编号 | char(4) | 外键 |
| Gno | 商品编号 | char(4) | 外键 |
| Rno | 仓库编号 | Char(4) | 外键 |

1. 配送关系表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Dno | 配送单号 | char(7) | 主键 |
| Dnum | 配送数量 | int |  |
| Ddate | 配送日期 | datetime | 非空 |
| Gno | 商品编号 | char(4) | 外键 |
| Rno | 仓库编号 | char(4) | 外键 |
| SHno | 门店编号 | Char(4) | 外键 |

1. 销售关系表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Tno | 销售单号 | char(7) | 主键 |
| Tnum | 销售数量 | int |  |
| Tdate | 销售日期 | datetime | 非空 |
| Gno | 商品编号 | char(4) | 外键 |
| SHno | 门店编号 | char(4) | 外键 |
| ASno | 营业员编号 | Char(4) | 外键 |

1. 存储关系表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段解释 | 数据类型 | 备注 |
| Gno | 商品编号 | char(4) | 外键 |
| Rno | 仓库编号 | char(4) | 外键 |
| STdate | 存储日期 | datetime | 与Cno、Rno共同构成主键，非空 |
| STno | 库存量 | char(5) |  |
| SSTno | 安全库存量 | char(5) |  |

1. 创建表和完整性约束代码设计
2. 创建商品信息表及完整性约束：

CREATE TABLE good

(

Gno char(4) primary key,

Gname varchar(20)unique not null,

Gtype varchar(10),

Gunit varchar(4),

Gunitp char(4)

)

1. 创建供应商信息表及完整性约束：

CREATE TABLE supplyor

(

SUno char(4) primary key,

SUname varchar(20)unique not null,

SUaccount char(7),

SUaddr varchar(20)

)

1. 创建仓库信息表及完整性约束：

CREATE TABLE repertory

(

Rno char(4) primary key,

Rddr varchar(20)unique not null,

Rmanager varchar(8)

)

1. 创建门店信息表及完整性约束:

CREATE TABLE shop

(

SHno char(4) primary key,

SHname varchar(20)unique not null,

SHaddr varchar(20)unique

)

1. 创建采购员信息表及完整性约束:

CREATE TABLE buyer

(

BRno char(4) primary key,

BRname varchar(8)unique not null,

BRscore int

)

1. 创建管理员信息表及完整性约束:

CREATE TABLE admin

(

Ano char(4) primary key,

Aname varchar(8)unique not null,

Ascore int,

Rno char(4),

foreign key (Rno) references repertory(Rno)

)

1. 创建营业员信息表及完整性约束:

CREATE TABLE assistant

(

ASno char(4) primary key,

ASname varchar(8)unique not null,

ASscore int,

SHno char(4),

foreign key (SHno) references shop(SHno)

)

1. 创建采购信息表及完整性约束:

CREATE TABLE buy

(

Bno char(7) primary key,

Bnum int,

Bdate datetime not null,

BRno char(4),

SUno char(4),

Gno char(4)

foreign key (BRno) references buyer(BRno),

foreign key (SUno) references supplyor(SUno),

foreign key (Gno) references good(Gno)

)

1. 创建进货信息表及完整性约束:

CREATE TABLE ini

(

Ino char(7) primary key,

Inum int,

Bdate datetime not null,

SUno char(4),

Gno char(4),

Rno char(4)

foreign key (SUno) references supplyor(SUno),

foreign key (Gno) references good(Gno),

foreign key (Rno) references repertory(Rno)

)

1. 创建配送信息表及完整性约束:

CREATE TABLE deliver

(

Dno char(7) primary key,

Dnum int,

Ddate datetime not null,

Gno char(4),

Rno char(4),

SHno char(4),

foreign key (Gno) references good(Gno),

foreign key (Rno) references repertory(Rno),

foreign key (SHno) references shop(SHno)

)

1. 创建销售信息表及完整性约束:

CREATE TABLE trade

(

Tno char(7) primary key,

Tnum int,

Tdate datetime not null,

Gno char(4),

SHno char(4),

ASno char(4),

foreign key (Gno) references good(Gno),

foreign key (SHno) references shop(SHno),

foreign key (ASno) references assistant(ASno),

)

1. 创建存储信息表及完整性约束:

CREATE TABLE store

(

Gno char(4),

Rno char(4),

STdate datetime,

STno char(5),

SSTno char(5),

foreign key (Gno) references good(Gno),

foreign key (Rno) references repertory(Rno),

primary key(Gno,Rno,STdate)

)

1. 创建视图、索引、存储过程和触发器
2. 创建视图
3. 创建采购员的视图。

create view P1

as

select buyer.BRno,BRname,BRscore,bno ,bnum ,Bdate,SUno,Gno

from buy,buyer

WHERE buyer.BRno=buy.Brno

1. 创建仓库管理员的视图。

create view P2

as

select admin.Rno,Aname,Ascore,Gno,STdate,STno,SSTno

from admin,repertory ,store

WHERE admin.Rno=repertory.Rno and store.Rno=repertory.Rno

1. 创建营业员的视图。

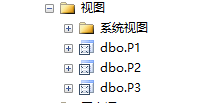
create view P3

as

select ASno,ASname,ASscore,shop.SHno,SHname,SHaddr

from assistant,shop

WHERE shop.SHno=assistant.SHno



1. 创建索引

为改数据库中的good和buyer两个表建立索引。其中good表按照商品单价升序建立唯一索引。buyer表按照采购员业绩升序建立唯一索引。

CREATE UNIQUE INDEX good on good(Gunitp);

CREATE UNIQUE INDEX buyer on buyer(BRscore);

1. 创建存储过程
2. 建立存储过程，输入商品编号得到该商品的所有信息。

create procedure good

@Gno char (4)

as

select \*

from good

where Gno = @Gno

1. 建立存储过程，输入商品得到该商品采购信息。

create procedure gbuy

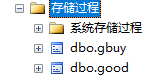
@Gno char (4)

as

select good.\*,buyer.BRno,BRname,BRscore,bno ,bnum ,Bdate,SUno

from good,buy,buyer

where buy.Gno = @Gno and buyer.BRno=buy.Brno



1. 创建触发器

CREATE TRIGGER tri\_UPDATE\_DELETE\_sc

ON good

FOR UPDATE,DELETE

AS

IF UPDATE(Gunitp)

BEGIN

SELECT INSERTED.Gno,DELETED.Gunitp AS原单价,

INSERTED.Gunitp AS 新单价

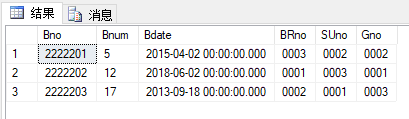
FROM DELETED ,INSERTED

WHERE DELETED.Gno=INSERTED.Gno

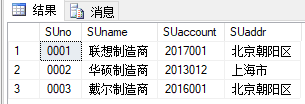
END

# 五、数据库功能调试

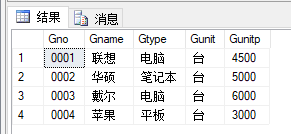
1. 采购员信息查询模块
2. 查询采购信息



1. 查询供应商信息



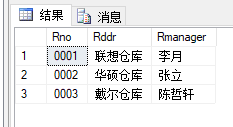
1. 查询商品信息



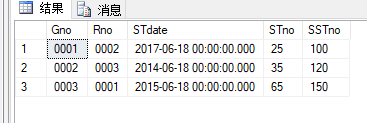
1. 查询采购员信息



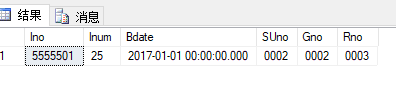
1. 仓库管理员信息查询模块
2. 查询仓库信息



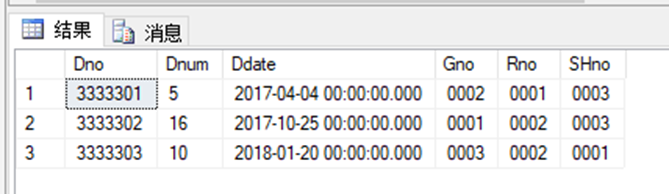
1. 查询存储信息



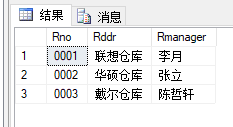
1. 查询进货信息



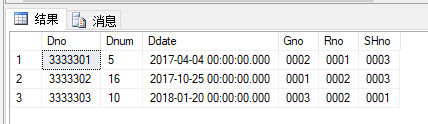
1. 查询配送信息



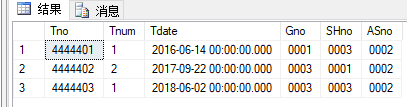
1. 查询仓库管理员信息



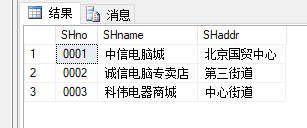
1. 营业员信息查询模块
2. 查询配送信息



1. 查询销售信息



1. 查询门店信息

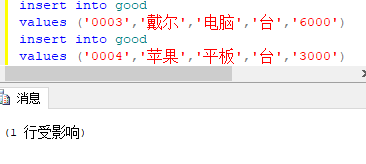


1. 查询营业员信息



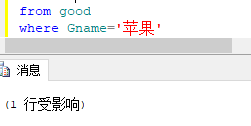
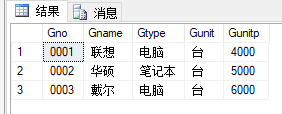
1. 系统管理员管理模块
2. 对商品的信息管理

1.1插入商品信息

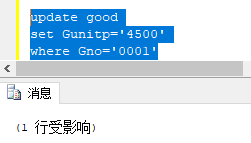
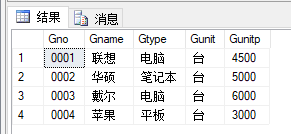




1.2删除商品信息

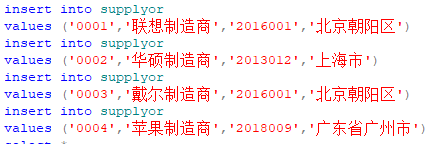
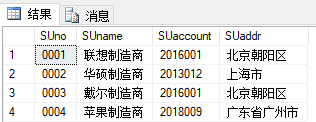
 

1.3修改商品信息

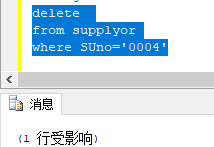
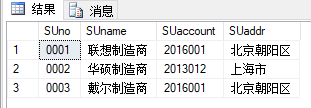
 

1. 对供应商的信息管理

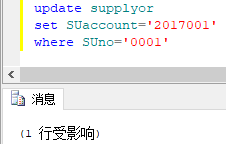
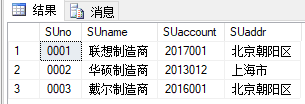
2.1插入供应商信息

2.2删除供应商信息

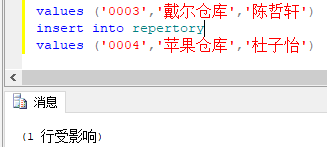
 

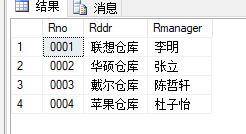
2.3修改供应商信息

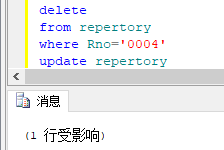
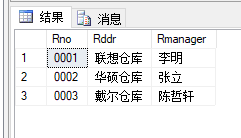
1. 对仓库的信息管理

3.1插入仓库信息

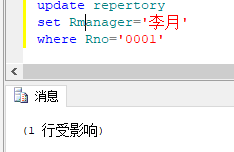
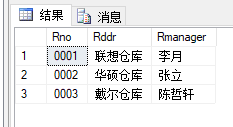




3.2删除仓库信息

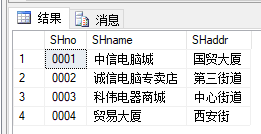
 

3.3修改仓库信息

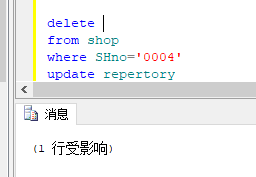
 

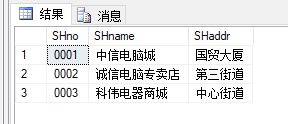
1. 对门店的信息管理

4.1插入门店信息

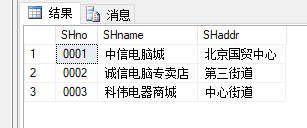
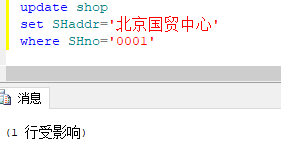
 

4.2删除门店信息



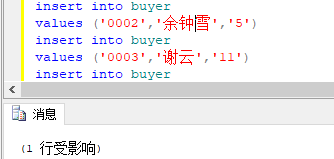


4.3修改门店信息

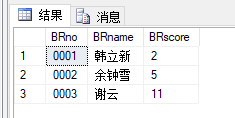
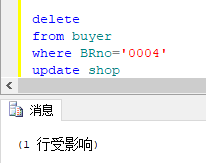


1. 对采购员的信息管理

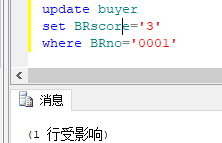
5.1插入设备信息

5.2删除采购员信息

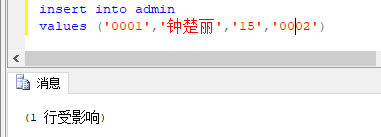
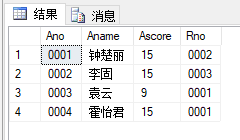


5.3修改采购员信息

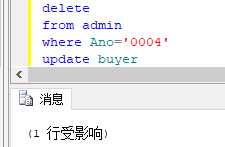


1. 对管理员的信息管理

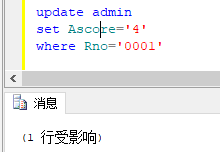
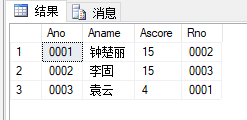
6.1插入管理员关系信息

6.2删除管理员信息

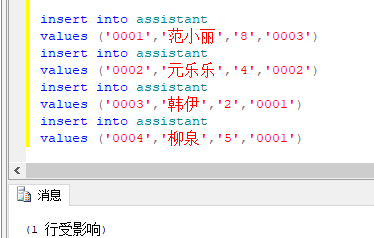
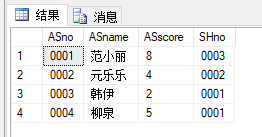
 

6.3修改管理员信息

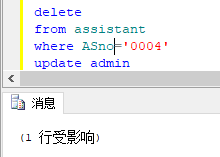
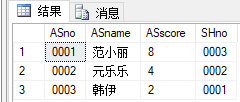
 

1. 对营业员的信息管理

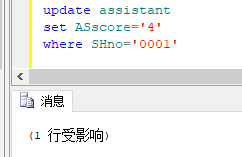
7.1插入营业员信息

7.2删除营业员信息

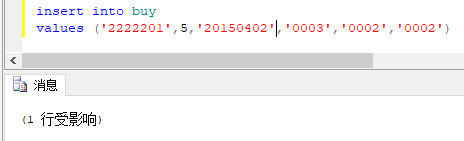
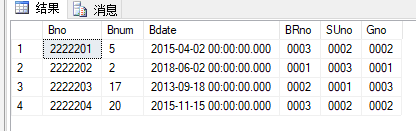
 

7.3修改营业员信息

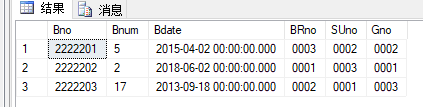
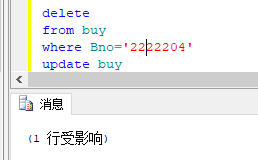
 

1. 对采购关系的信息管理

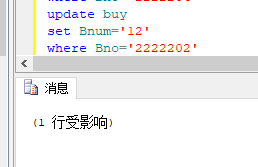
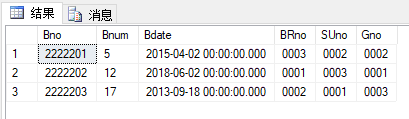
8.1插入采购关系信息

8.2删除采购关系信息

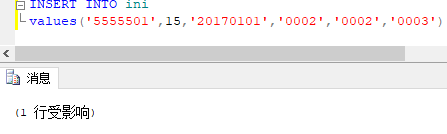
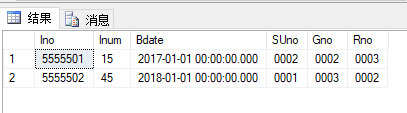


8.3修改采购关系信息

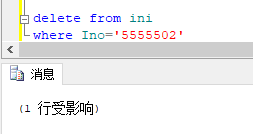
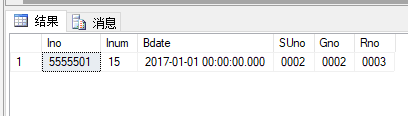
 

1. 对进货关系的信息管理

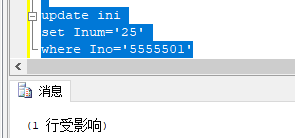
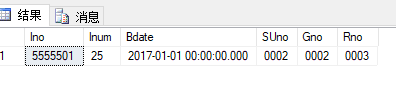
9.1插入进货关系信息

9.2删除进货关系信息

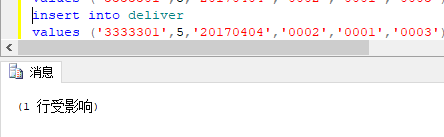
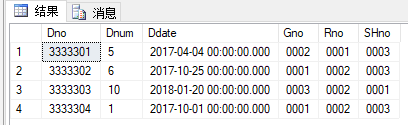
 

9.3修改进货关系信息

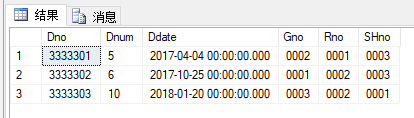
 

1. 对配送关系的信息管理

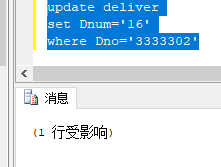
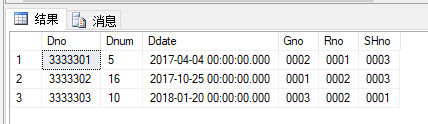
10.1插入配送关系信息

10.2删除配送关系信息

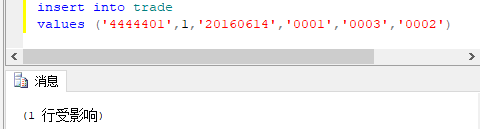
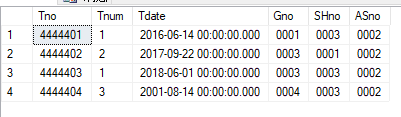
 

10.3修改配送关系信息

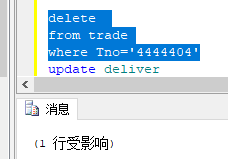
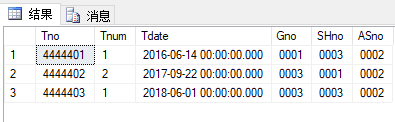
 

1. 对销售关系的信息管理

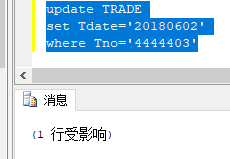
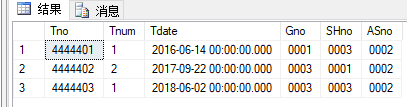
11.1插入销售关系信息

11.2删除销售关系信息

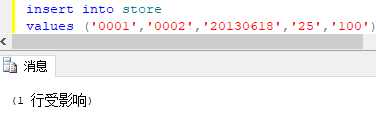
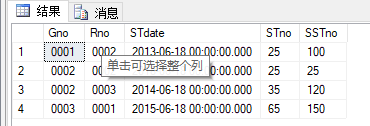
11.3修改销售关系信息

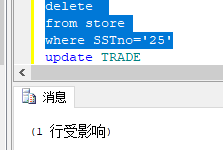
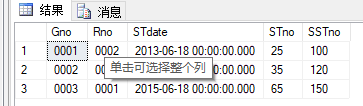
# 六、设计系统前台软件

1. 对存储关系的信息管理

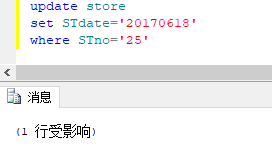
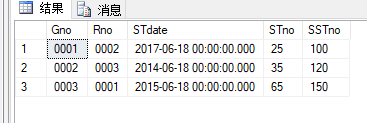
12.1插入存储关系信息

12.2删除存储关系信息

12.3修存储关系信息

（一）开发软件选择

ODBC进行数据源设置

IIS进行ASP配置

Dreamweaver进行站点配置

（二）软件功能要求与设计

实现不同的使用用户登录功能，默认界面信息的展示功能，详细信息的跳转功能，对信息表的查询功能，对信息表项的插入功能，对信息表项的更新功能。

这些具体的功能通过Dreamweaver中选择服务器行为实现，设定，随后通过站点链接到浏览器。通过页面的显示与界面的跳转实现功能要求。

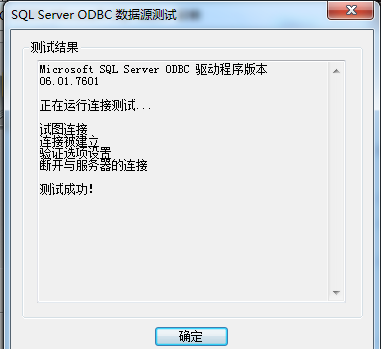
（三）软件功能实现

1.使用DBC进行数据源设置

连接SQL Sever服务器，

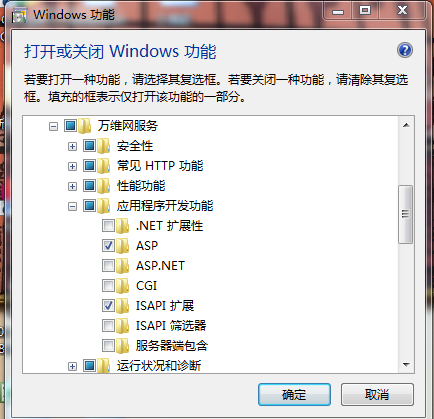


完成后，数据源测试成功。

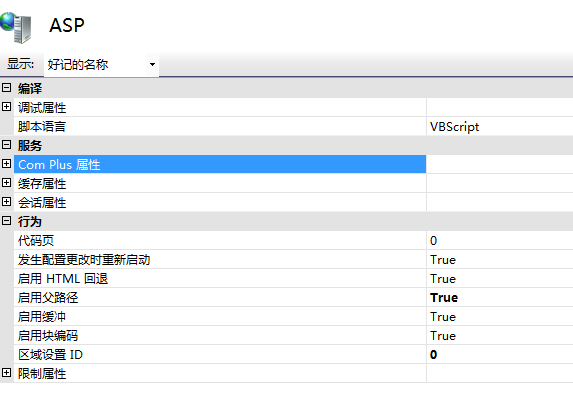


2.使用IIS进行ASP配置

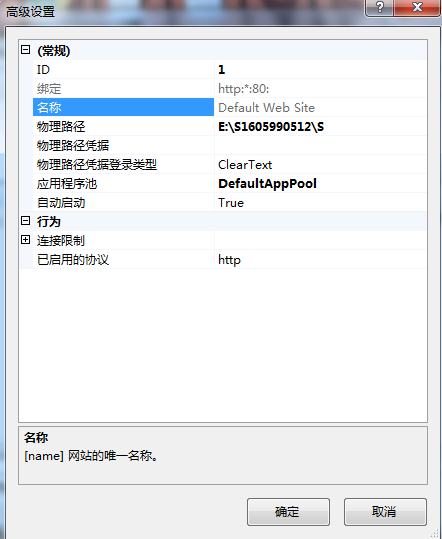
首先，打开控制面板中的打开或关闭Windows功能，勾选ASP等设置：



随后，进入IIS将ASP的启用父路径更改为true

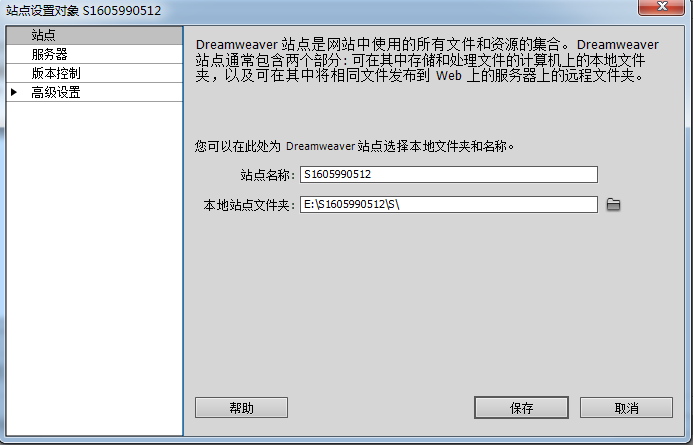


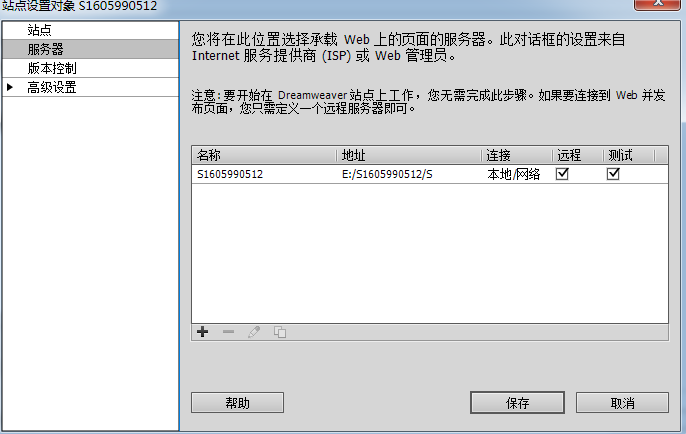
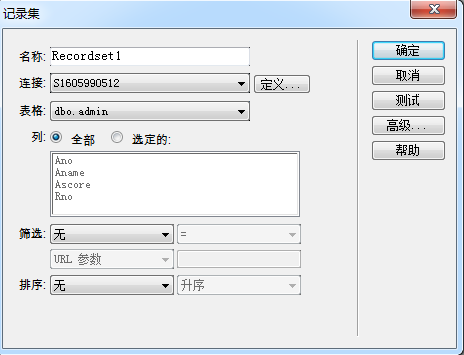
将默认网址的物理路径修改为存放站点的文件夹：



1. 使用Dreamweaver建立站点、记录集

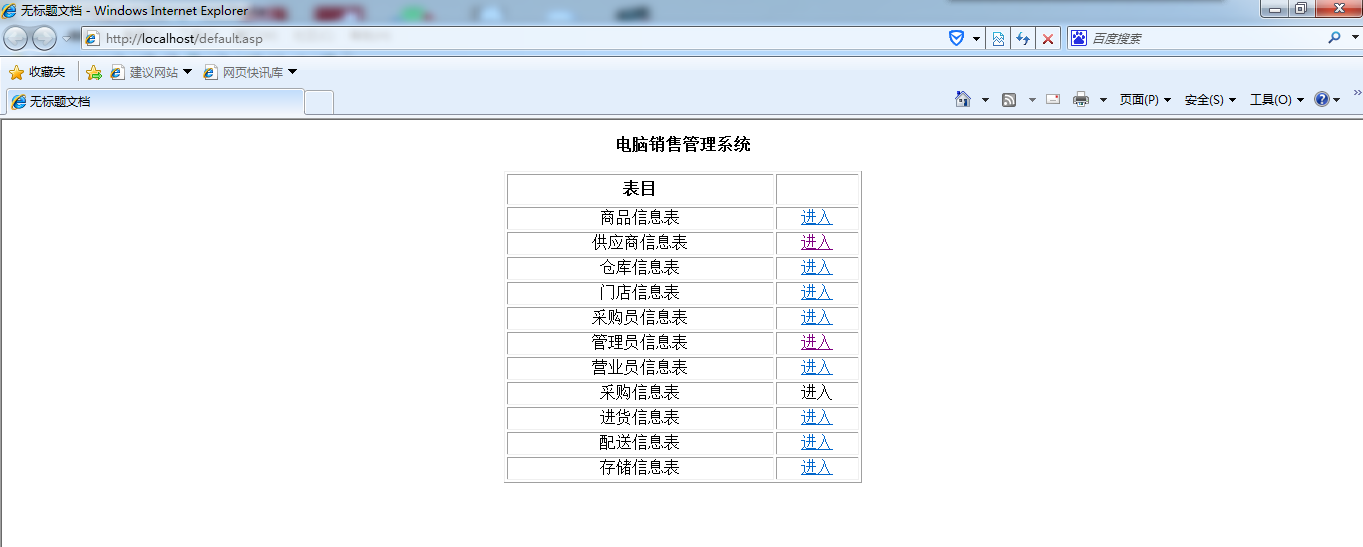
在Dreamweaver里新建一个VBScript文件，完成站点的建立，测试服务器的运行，记录集的建立：





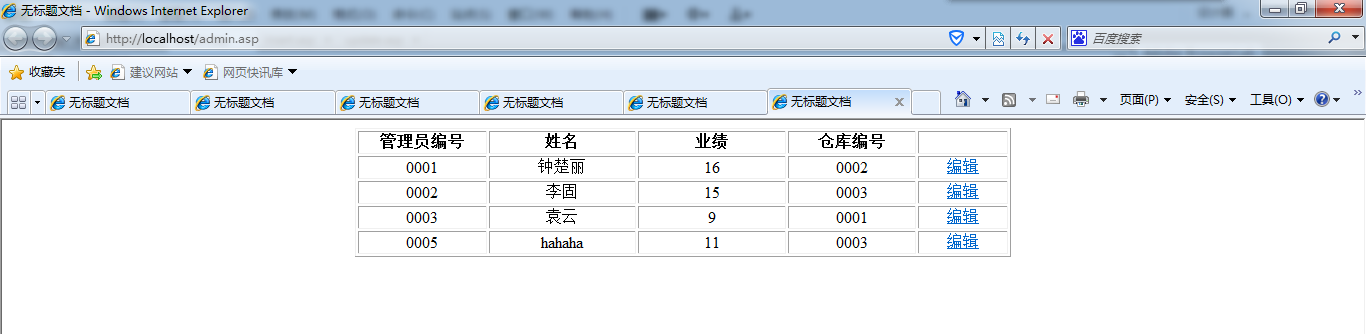
1. 设计default.asp界面：

在VBScript文件中插入一个13行2列的表格。在表格中插入各个信息表，选中“进入”这两个字，添加服务器行为，选择跳转至详情页面，将各个信息表项与各个信息表的主页对应连接。完成界面如图：

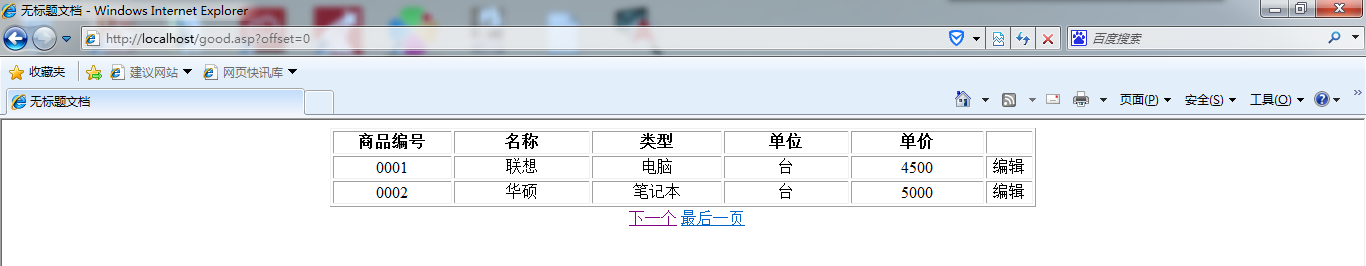


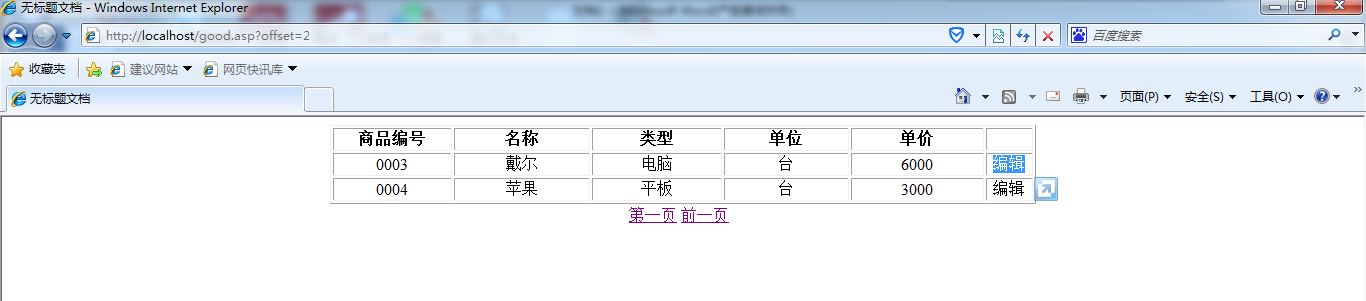
1. 设计各个信息表界面：

重新新建一个VBScript文件，插入两行的表格，在第一行输入表头，将记录集拖入第二行，重复第二行区域，随机就可以在网页中显示整个表的数据信息：



插入记录及导航条之后的页面显示为：

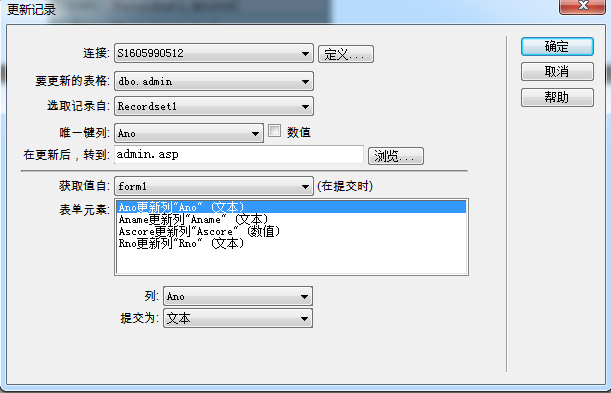


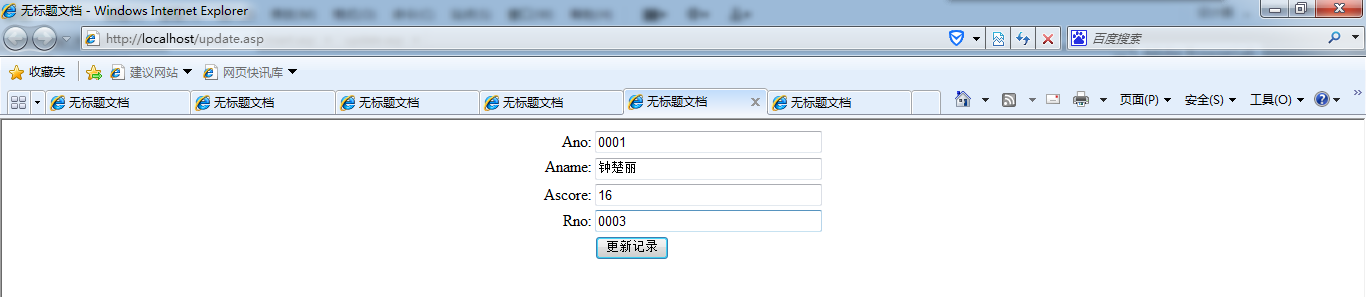


1. 设计更新界面：

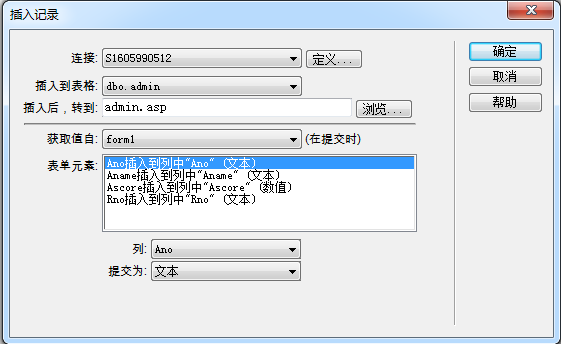
插入服务器行为更新记录，设置update.asp页面跳转至admin.asp页面

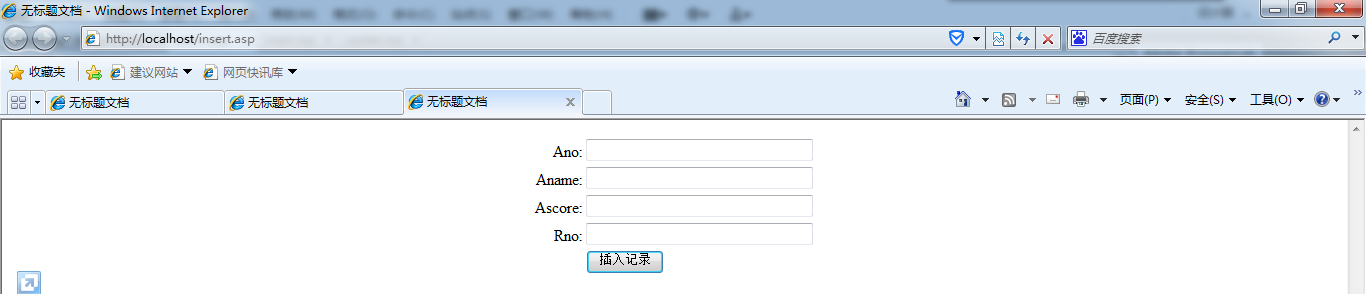




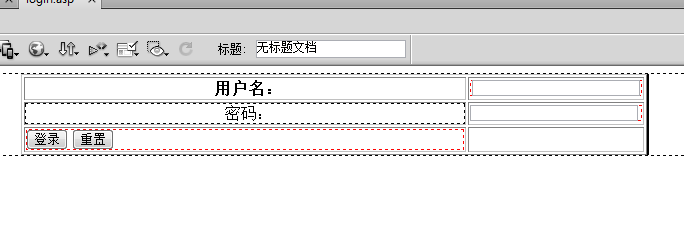
更新页面显示如下：

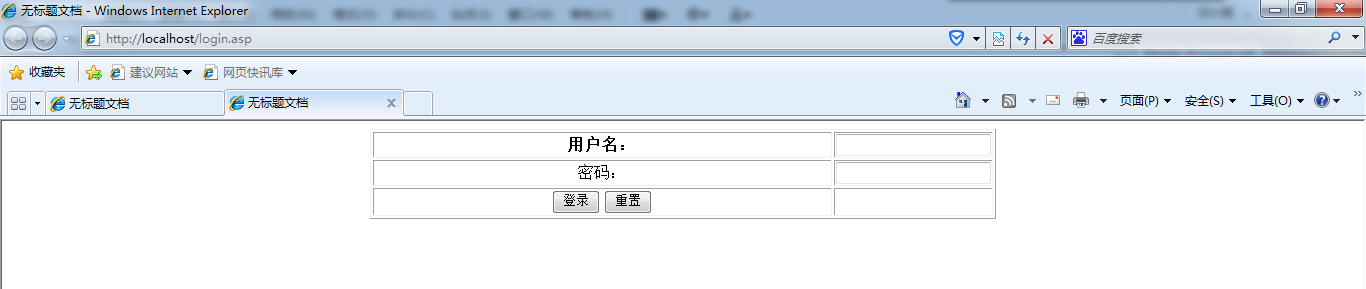
1. 设计插入界面：

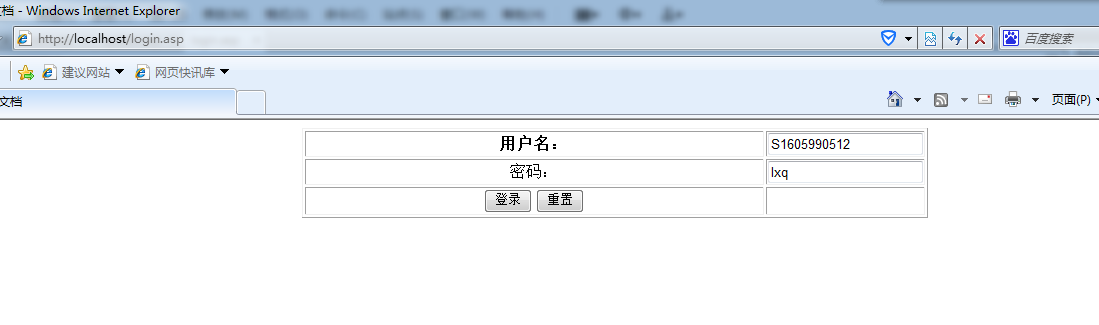




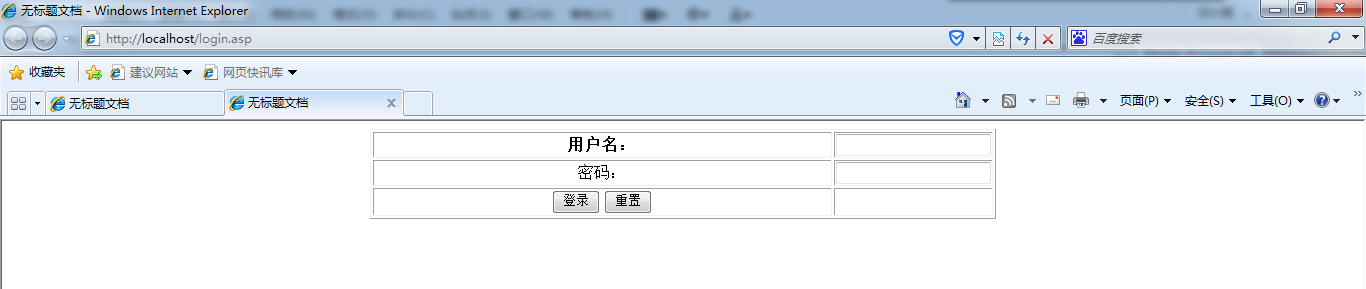
1. 设计登录界面：

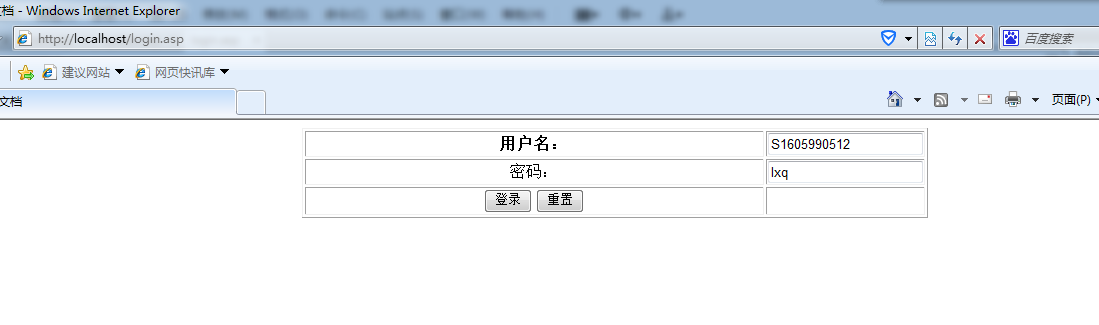




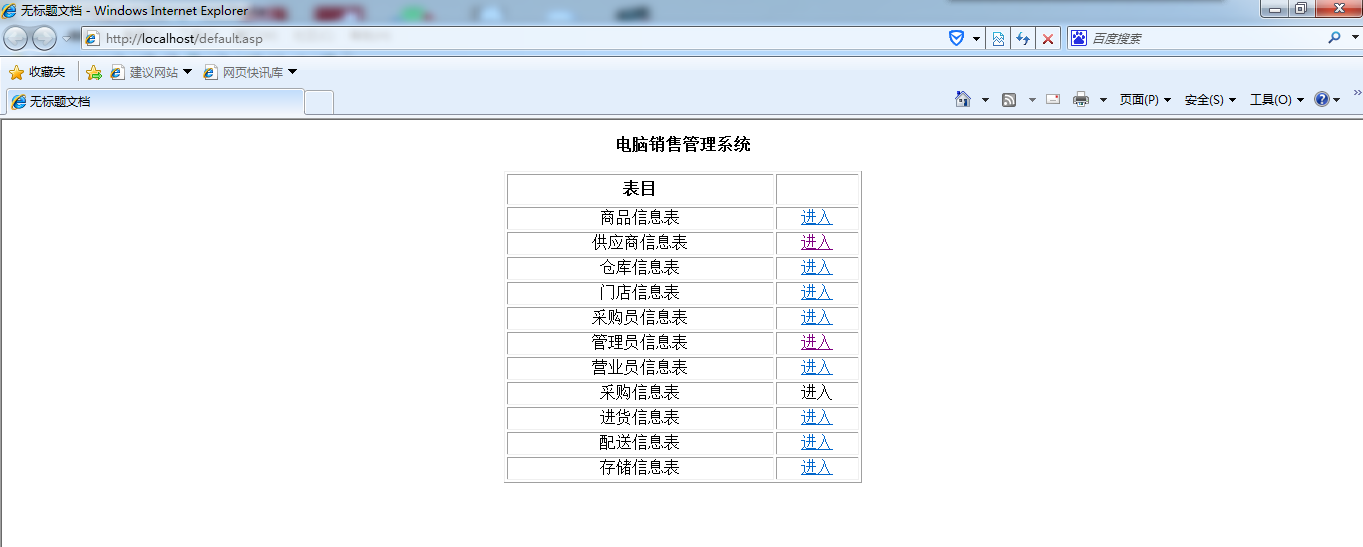


1. 系统测试
2. 登录页面

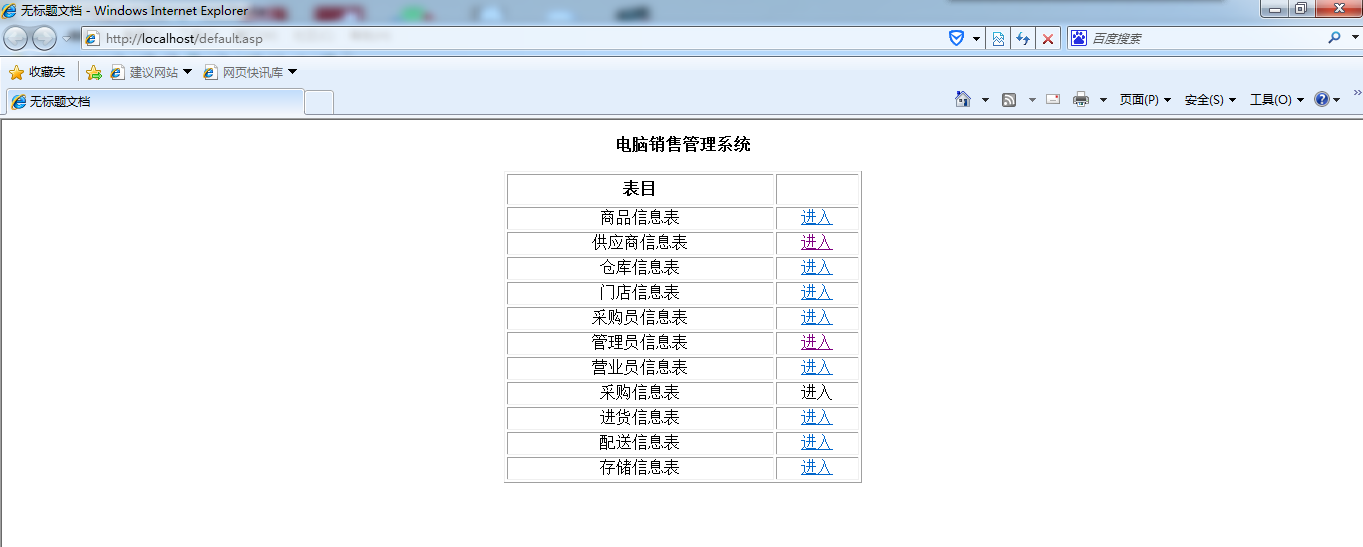




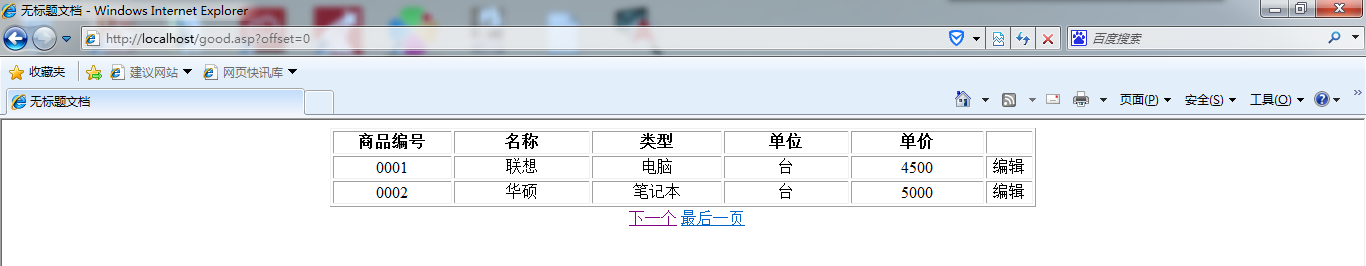
登录后跳转至默认页面

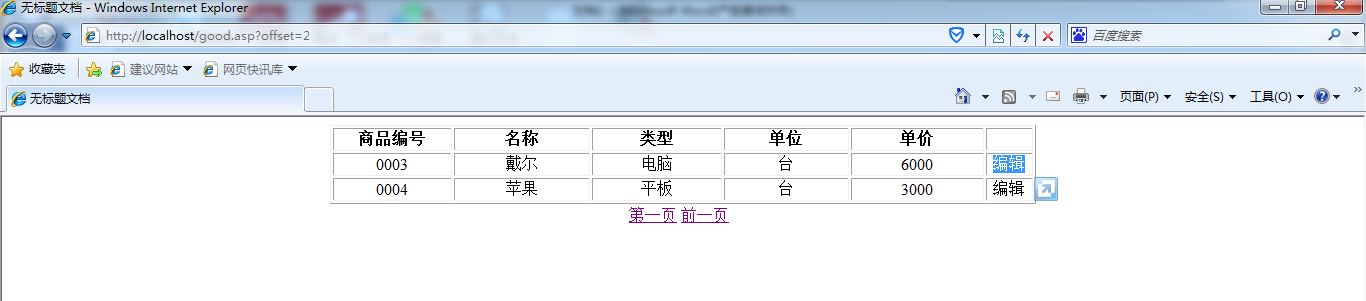


1. 默认界面

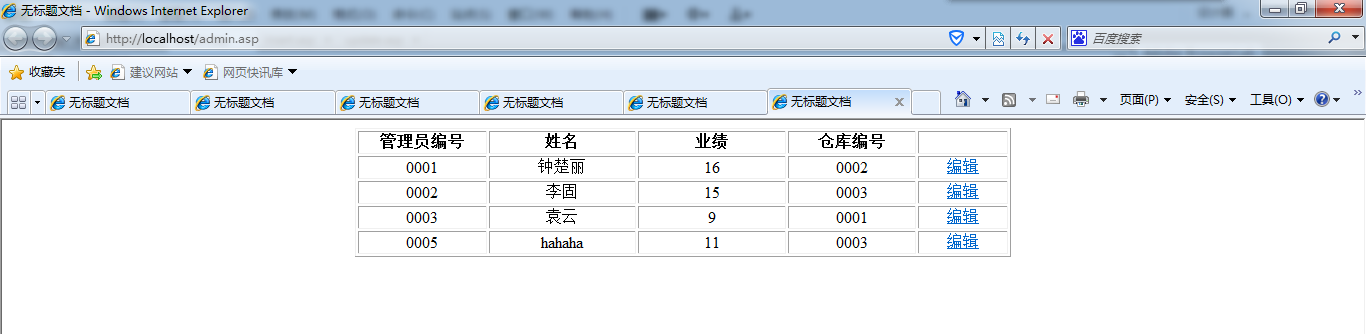


1. good信息表页面

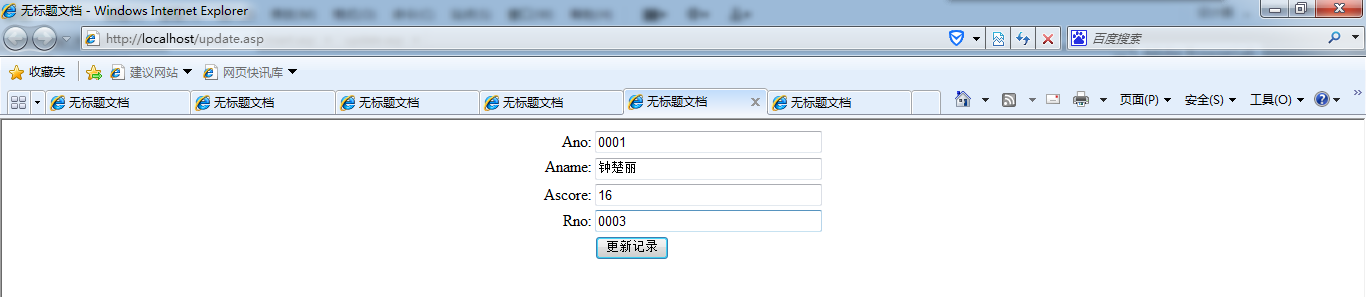


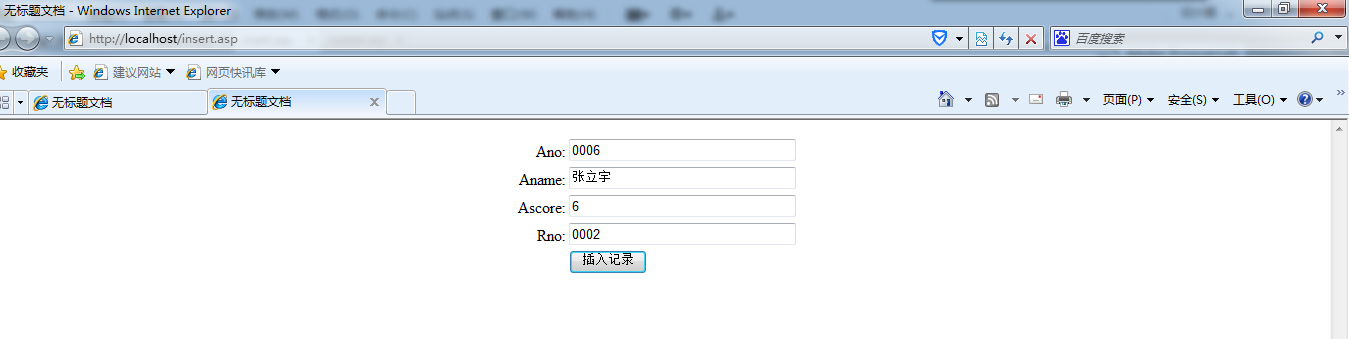


1. admin信息表页面



1. 更新页面

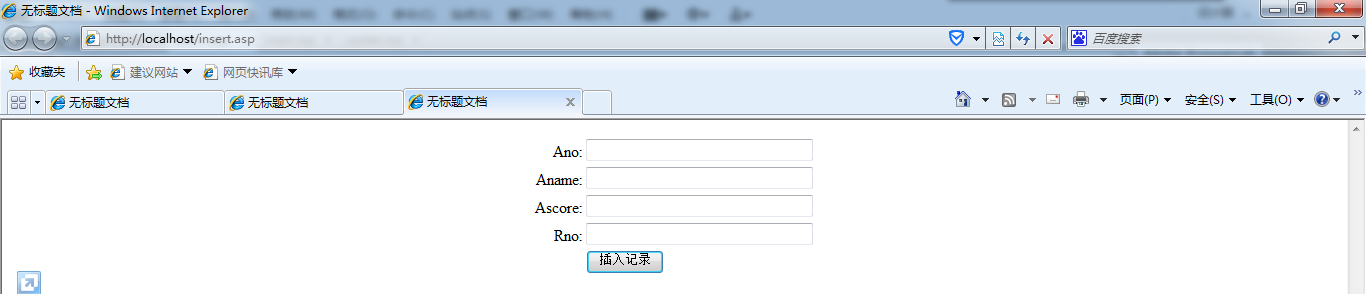


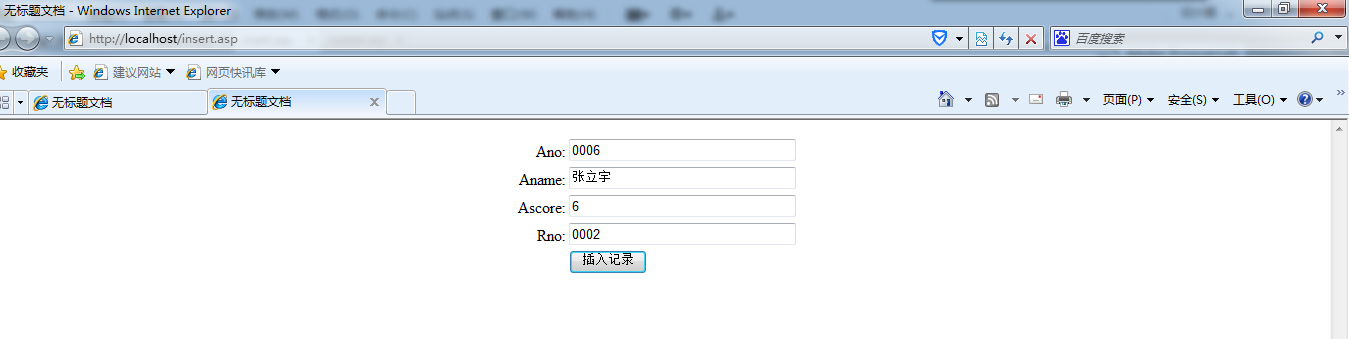


更新后：



1. 插入页面





插入成功后：



# 七、设计总结

这次课程设计，使我体会到了关于设计数据库的各个方面的过程，先是对问题进行需求分析，画出E-R图，进行概念结构设计，完成数据字典，生成概念模型图，再进行逻辑结构设计，将概念模型转换为逻辑模型，逻辑模型图，之后进行数据库的物理实现，也就是形成物理表和在SQL sever里输入语句，随即功能实现，可以进行调试。此外，最后还实现了数据库的前端界面显示，是通过ARP与SQL连接完成的，可在浏览器中访问网页实现对数据库的访问。

特别的是，本次的数据库设计是面向对象完成的，非常符合现代程序的积极特点，通过授予不同用户不同的权限以及信息领域，将他们可以访问的数据明确出来，同时，也可使用户使用更加清楚和简单。

这次的课程设计相对来说是比较规模大的，而且又用到了许多知识，其中也包括一些没学过的知识。所以，我在进行课程设计的过程中，也找到了自己的一些不足，比如说，数据库的语句知识掌握的不够熟练，对陌生软件的适应时间长，许多专业课的拓展知识了解的不够等等。但是，经历过这次数据库课设，我也学到了很多东西，这种东西不仅是留于知识层面的，对我来说还有其他更深远的意义。因为，我不禁巩固了数据库的知识，学到了数据库前端实现的一种方法，学会使用了power designer、visio、dreamweaver等软件的基本操作，还掌握到了短时间熟悉新知识、学习新知识的能力，除此之外，还收获到了提高效率、细心做事考虑周到、泰然处之的心境。这是令我受益匪浅的。

再往后的学习练习中，我要改正发现的缺点，总结本次课程设计的经验，将学到的知识运用到往后的学习中去，不断巩固、创新。