

**山东大学（威海）**

**药店进销存人管理系统**

**后台数据库设计与实现**

班级：19级计合02班

作者：钱克洋

学号：201900800151 \_\_\_\_\_

指导老师：王文玉 \_\_\_\_\_\_

编写日期：2021 年 4 月 26 日

目录

[一．引言 4](#_Toc69808925)

[1.1 选题说明 4](#_Toc69808926)

[1.2 需求介绍 4](#_Toc69808927)

[二．系统功能模块 5](#_Toc69808928)

[2.1 系统功能需求分析 5](#_Toc69808929)

[2.2 系统数据流图 6](#_Toc69808930)

[2.3 系统功能模块总图 7](#_Toc69808931)

[2.4 系统功能模块子图 8](#_Toc69808932)

[2.5 数据字典 12](#_Toc69808933)

[三．数据库概念结构设计 23](#_Toc69808934)

[3.1 局部E-R图 23](#_Toc69808935)

[3.2 全局E-R图 26](#_Toc69808936)

[四．逻辑结构设计 28](#_Toc69808937)

[4.1 仓库表 28](#_Toc69808938)

[4.2 订单表 28](#_Toc69808939)

[4.3 供应商表 29](#_Toc69808940)

[4.4 客户表 29](#_Toc69808941)

[4.5 入库表 30](#_Toc69808942)

[4.6 药品表 31](#_Toc69808943)

[4.7 员工表 32](#_Toc69808944)

[4.8 范式分析 33](#_Toc69808945)

[五．物理结构设计 33](#_Toc69808946)

[5.1 表的存储与实现 33](#_Toc69808947)

[5.2 视图、触发器与存储过程实现 39](#_Toc69808948)

[5.3 数据库安全性设计 47](#_Toc69808949)

[5.4 数据库完整性设计 49](#_Toc69808950)

[5.5 索引设计 53](#_Toc69808951)

[六．数据库完成效果 54](#_Toc69808952)

[七．总结 56](#_Toc69808953)

[八．参考资料 57](#_Toc69808954)

[九．致谢 57](#_Toc69808955)

# 一．引言

## 1.1 选题说明

**作为数据库系统学习的终点，课程设计的选题应为构建一个功能较完整的商店/公司数据库系统；而由于课设作业是单人完成，不需要像软件工程实验那样设计多个复杂子系统。故而我选取了药店进销存人管理系统。由于药店需求较简单，该系统不会过于复杂，同时又根据使用者、使用目的的不同分有多个功能模块，结构较为完备。**

## 1.2 需求介绍

**药店进销存人管理系统具有如下需求：**

**实现药品仓库、订单、供应商、客户、药品、员工、入库记录等关系的管理；**

**实现药品信息的管理，入库时能自动记录并修改库存（使用触发器实现），购入新药品时能自动更新药品表并修改库存（使用触发器实现）；**

**实现药品退货功能，退货时能自动删除入库记录并修改库存（使用触发器实现）；**

**假定药品数量只能是整数，并用check子句检查员工类型存储是否符合要求；**

**定义视图供客户查询订单信息；**

**定义视图供员工查询在售药品信息；**

**定义视图查询客户、员工与管理员登陆所需信息；**

**定义视图查询仓库中存放的药品信息和数量；**

**定义视图供管理员查询月销售明细；**

**定义存储过程查询月总体销售情况；**

**建立数据库相关表之间的参照完整性约束；**

**建立表间关系；**

**存储员工照片信息；**

**实现关系模型规范化；**

**保证数据库安全性和完整性，确定关系的主键与索引，授予客户、员工与管理员等不同用户类型不同权限。**

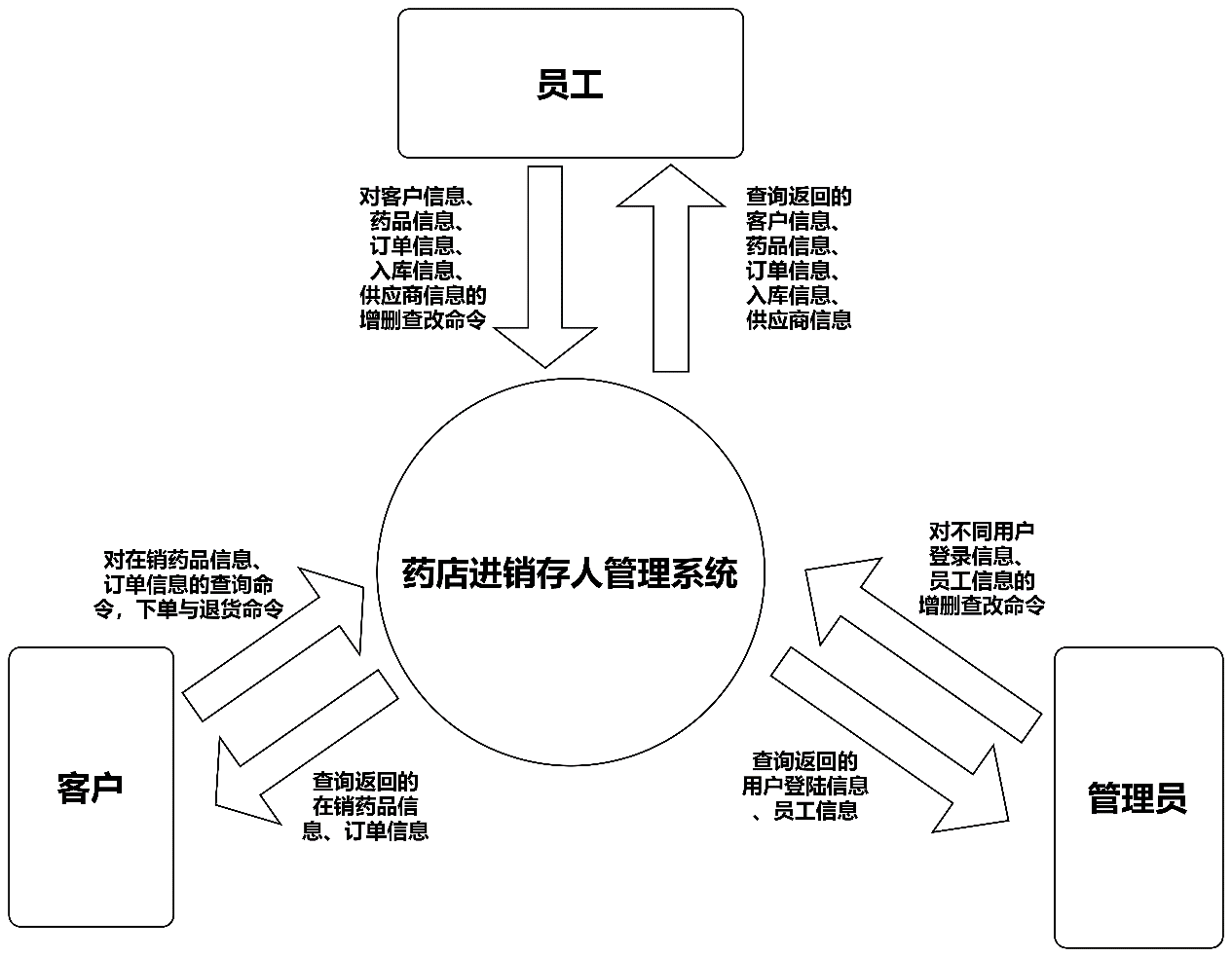
# 二．系统功能模块

## 2.1 系统功能需求分析

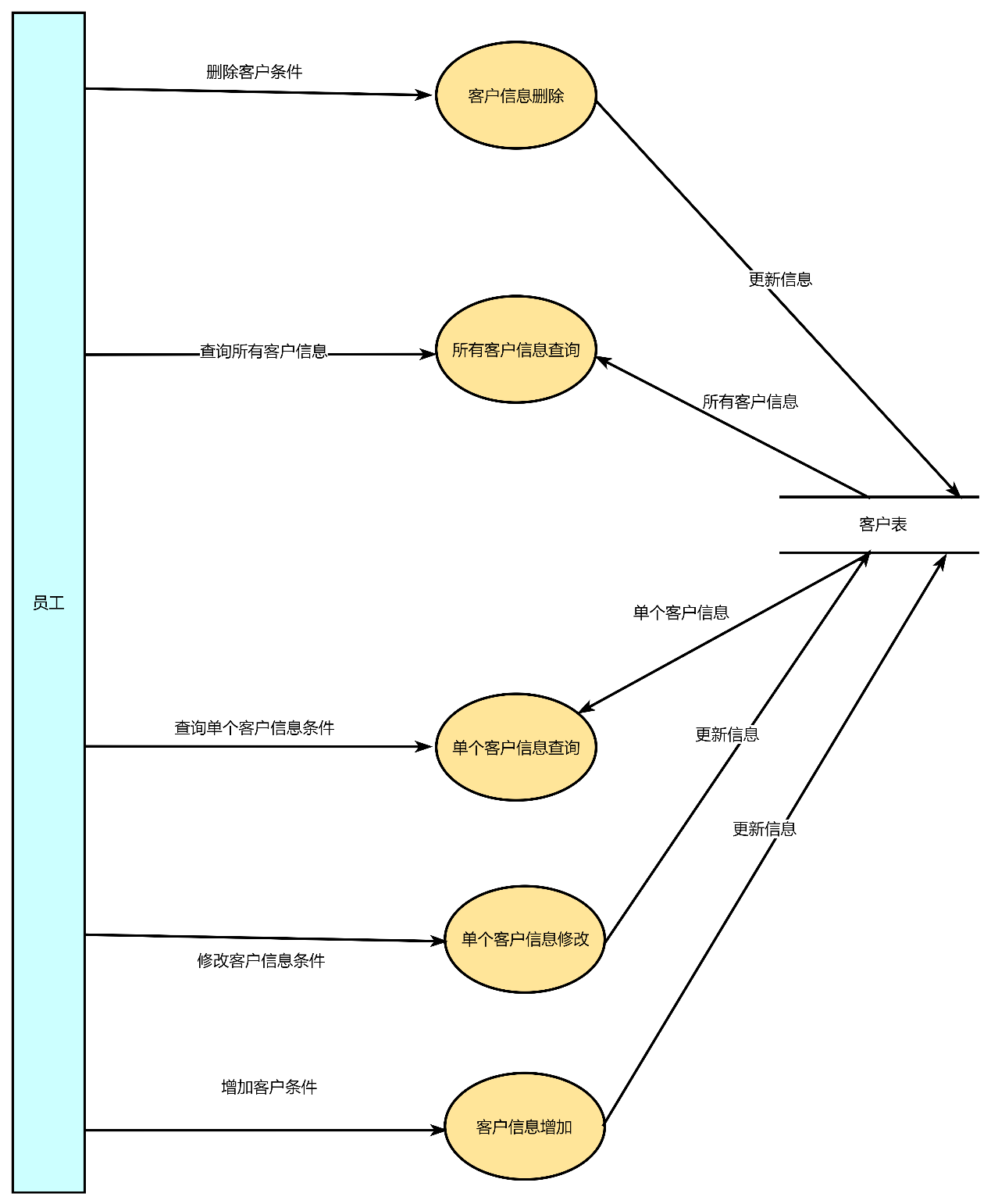
系统按照使用者划分，可分为客户系统、员工系统与管理员系统；按照功能分，则可分为客户信息操作功能、药品信息操作功能、员工信息操作功能、登陆信息操作功能、客户操作功能、订单管理功能、入库信息操作功能、供应商信息操作功能、统计信息操作功能共9个功能模块子系统。

## 2.2 系统数据流图

使用数据流图对系统需求进行分析。由于若以功能作为子系统划分依据，则各子系统数据流图将多有重复；并且9个功能子系统数据库间联系薄弱，其实现也仅为较简单的数据库操作，故从实际考虑，只绘制了顶层数据流图与一张作为范例的二层数据流图。

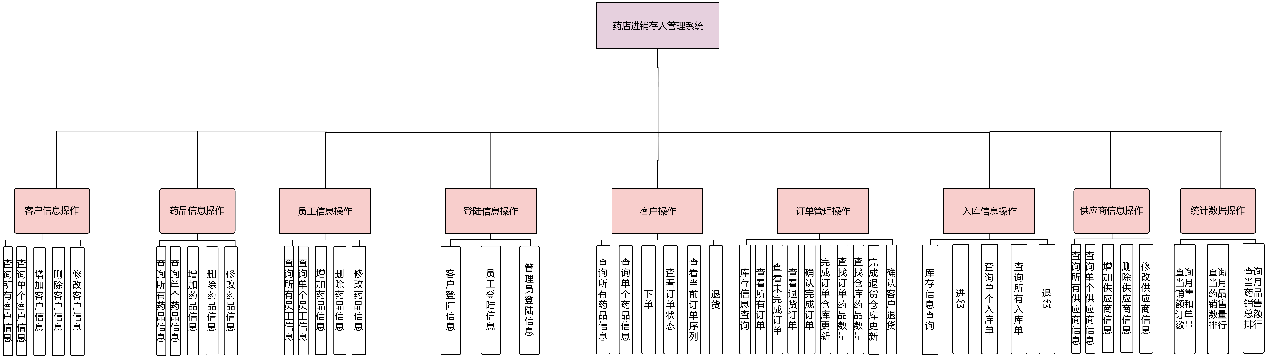


下图为客户信息操作子系统数据流图。



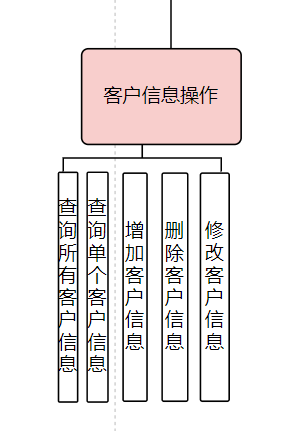
由此已可确定系统功能模块与各子系统所负责的操作，见以下所列系统功能模块图。

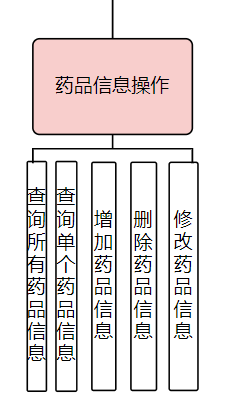
## 2.3 系统功能模块总图

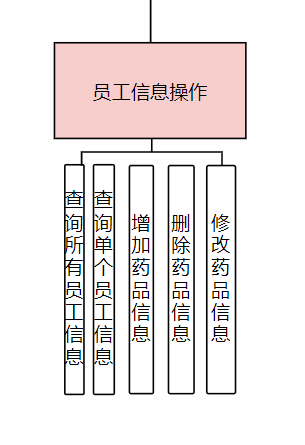


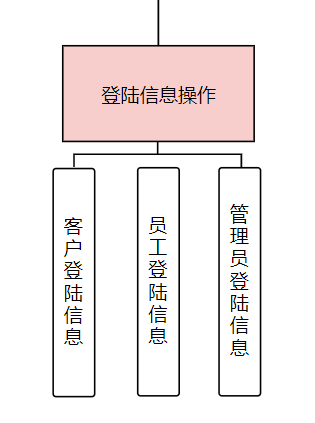
为便于查看，以下列举各功能子系统功能模块图。

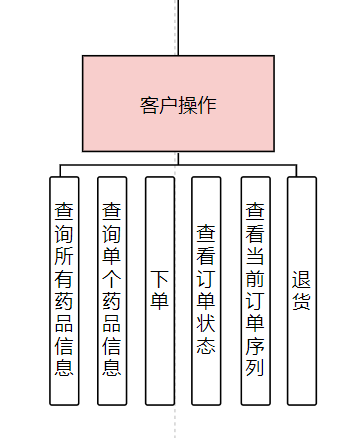
## 2.4 系统功能模块子图

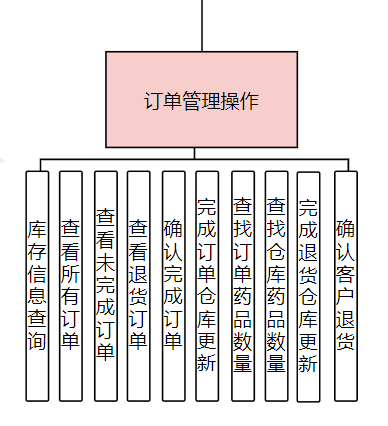


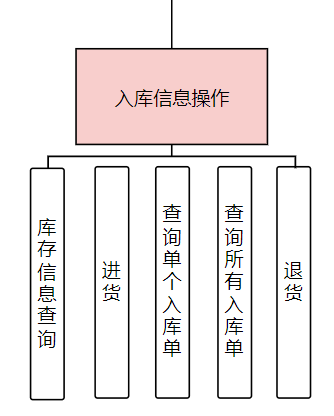


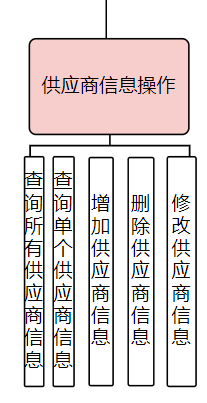


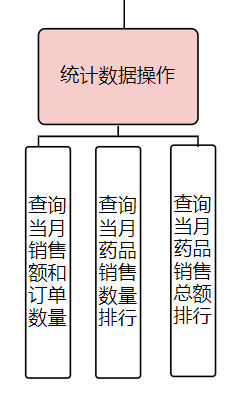










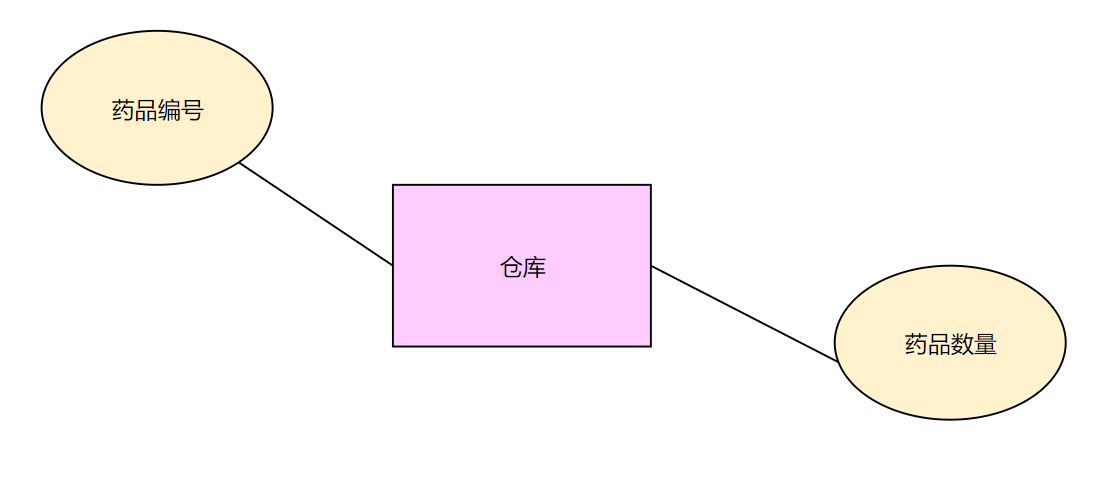


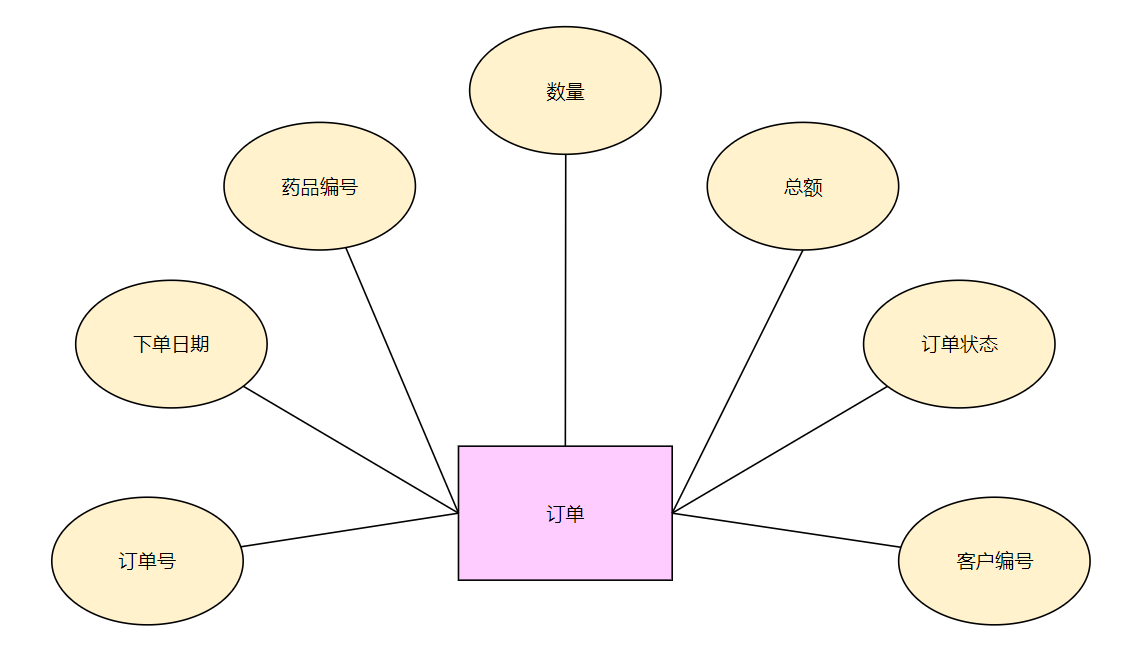
## 2.5 数据字典

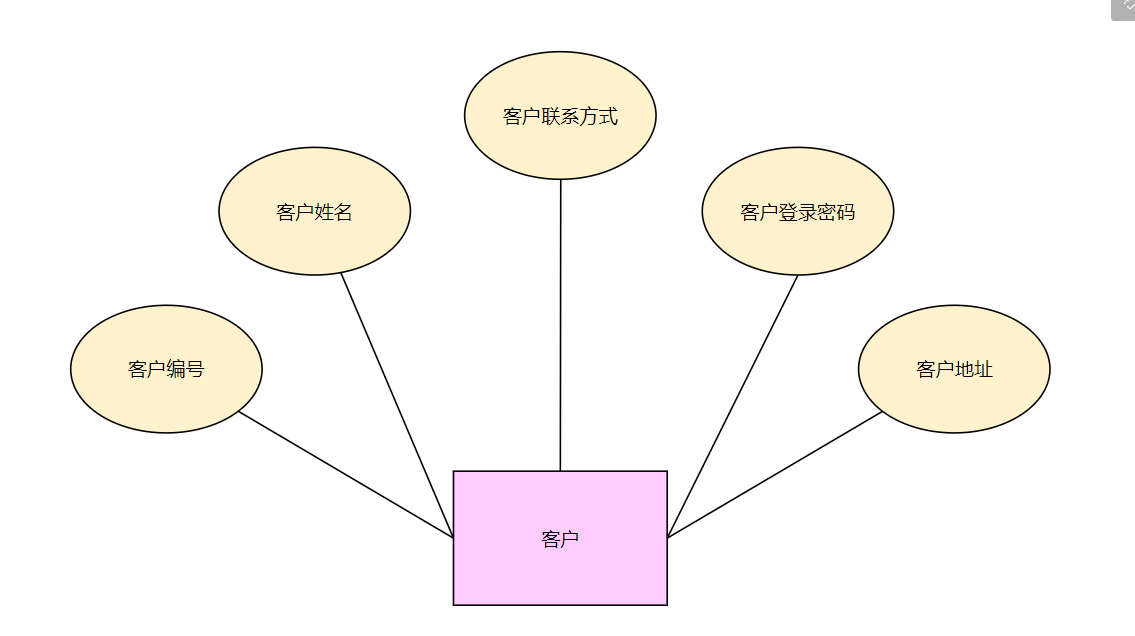
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 数据类型 | 取值范围 | 取值含义 | 相关的数据元素及数据结构 |
| 仓库药品编号 | 仓库药品编号 | varchar(8) | NOT NULL |  | 药品编号，药品数量 |
| 药品数量 | 药品数量 | int | NOT NULL,DEFAULT(0) |  | 仓库药品编号 |
| 订单号 | 订单号 | varchar(8) | NOT NULL |  | 下单日期，总额，订单状态，订单药品编号，订单药品，订单药品数量，订单客户编号 |
| 下单日期 | 下单日期 | date | NOT NULL |  | 订单号 |
| 总额 | 总额 | float(53) | NOT NULL |  | 订单号 |
| 订单状态 | 指示订单状态 | varchar(1) | NOT NULL,('0','1'),DEFAULT(‘0’) | ‘0’表示未完成，‘1’表示已完成 | 订单号 |
| 订单药品编号 | 订单药品编号 | varchar(8) | NOT NULL |  | 药品编号，订单号 |
| 订单药品数量 | 订单药品数量 | int | NOT NULL |  | 订单号 |
| 订单客户编号 | 客户编号 | varchar(8) | NOT NULL |  | 客户编号，订单号 |
| 供应商编号 | 供应商编号 | varchar(8) | NOT NULL |  | 供应商名称，供应商联系方式，供应商地址，药品供应商编号 |
| 供应商名称 | 供应商名称 | varchar(90) | NOT NULL |  | 供应商编号 |
| 供应商联系方式 | 供应商联系方式 | varchar(90) | NOT NULL |  | 供应商编号 |
| 供应商地址 | 供应商地址 | varchar(90) | NOT NULL |  | 供应商编号 |
| 客户编号 | 客户编号 | varchar(8) | NOT NULL |  | 客户姓名，客户联系方式，客户登录密码，客户地址，订单客户编号 |
| 客户姓名 | 客户姓名 | varchar(30) | NOT NULL |  | 客户编号 |
| 客户联系方式 | 客户联系方式 | varchar(30) | NOT NULL |  | 客户编号 |
| 客户登录密码 | 客户登录密码 | varchar(30) | NOT NULL |  | 客户编号 |
| 客户地址 | 客户地址 | varchar(90) | NOT NULL |  | 客户编号 |
| 入库单号 | 入库单号 | varchar(8) | NOT NULL |  | 入库药品编号，入库数量，入库日期 |
| 入库药品编号 | 入库药品编号 | varchar(8) | NOT NULL |  | 药品编号，入库单号 |
| 入库数量 | 入库数量 | int | NOT NULL |  | 入库单号 |
| 入库日期 | 入库日期 | date | NOT NULL |  | 入库单号 |
| 药品编号 | 药品编号 | varchar(8) | NOT NULL |  | 药品名称，药品功效，药品单价，有效期，药品类型，药品供应商编号，仓库药品编号，订单药品编号，入库药品编号 |
| 药品名称 | 药品名称 | varchar(90) | NOT NULL |  | 药品编号 |
| 药品功效 | 药品功效 | varchar(90) | NOT NULL |  | 药品编号 |
| 药品单价 | 药品单价 | float(53) | NOT NULL |  | 药品编号 |
| 有效期 | 有效期 | varchar(30) | NOT NULL |  | 药品编号 |
| 药品类型 | 药品类型 | varchar(30) | NOT NULL |  | 药品编号 |
| 药品供应商编号 | 药品供应商编号 | varchar(8) | NOT NULL |  | 供应商编号，药品编号 |
| 员工照片 | 员工照片 | image | DEFAULT NULL |  | 员工工号 |
| 员工工号 | 员工工号 | varchar(8) | NOT NULL |  | 员工照片，员工姓名，员工联系方式，员工登录密码，员工职务，是否管理员 |
| 员工姓名 | 员工姓名 | varchar(30) | NOT NULL |  | 员工工号 |
| 员工联系方式 | 员工联系方式 | varchar(30) | NOT NULL |  | 员工工号 |
| 员工登录密码 | 员工登录密码 | varchar(30) | NOT NULL |  | 员工工号 |
| 员工职务 | 员工职务 | varchar(30) | NOT NULL |  | 员工工号 |
| 是否管理员 | 指示是否为管理员 | varchar(1) | NOT NULL,('0','1') ,DEFAULT(‘0’) | ‘0’表示一般员工，‘1’表示管理员 | 员工工号 |

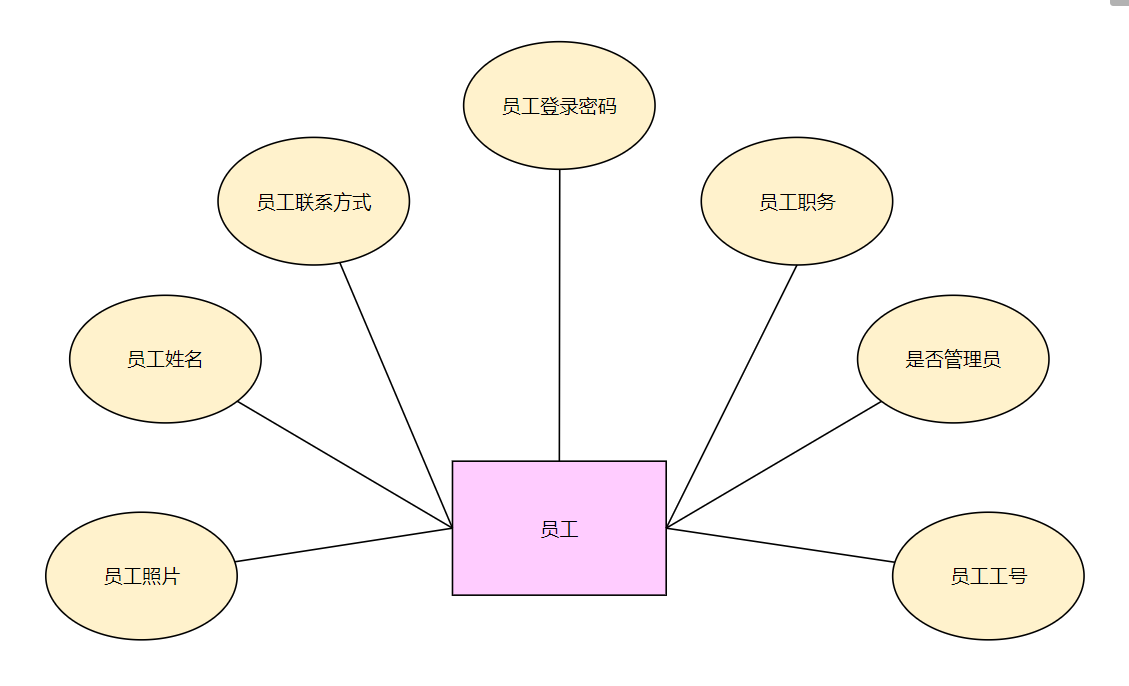
# 三．数据库概念结构设计

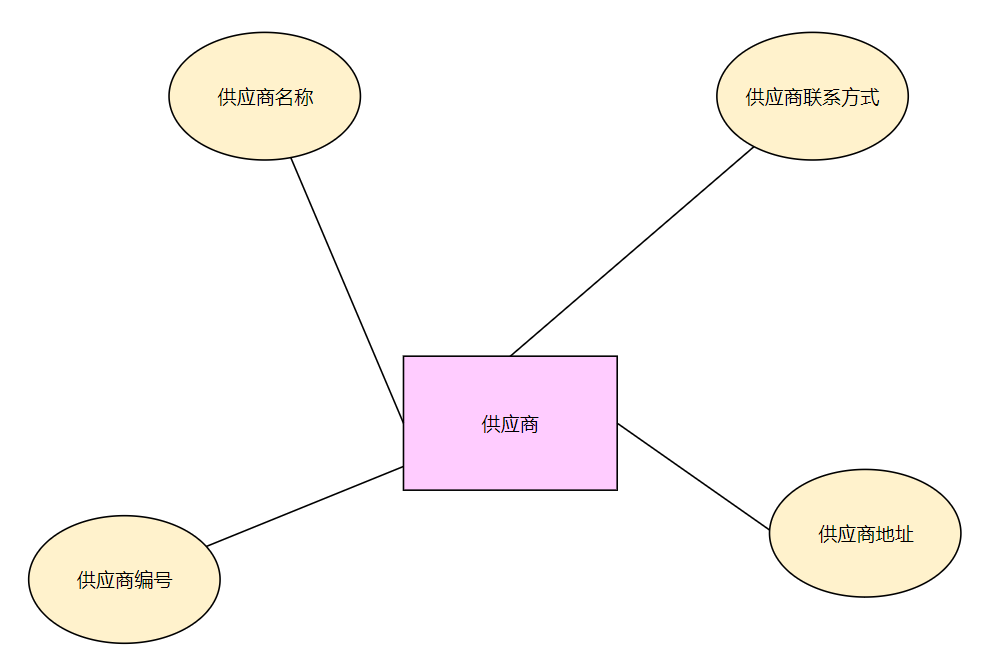
## 3.1 局部E-R图

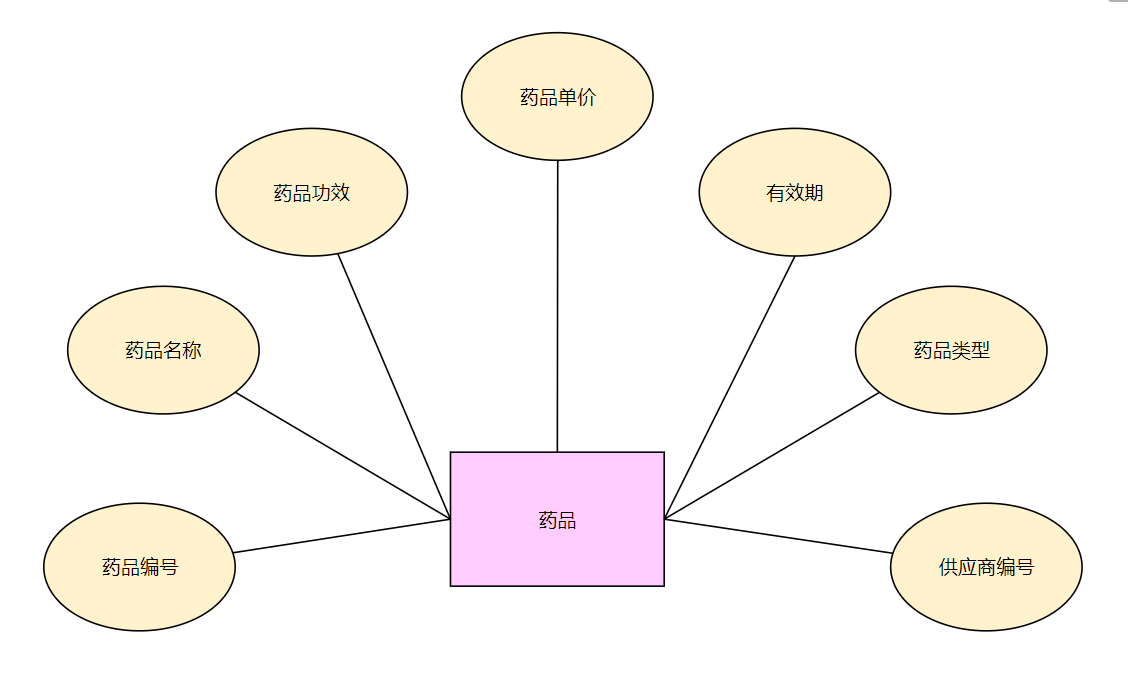


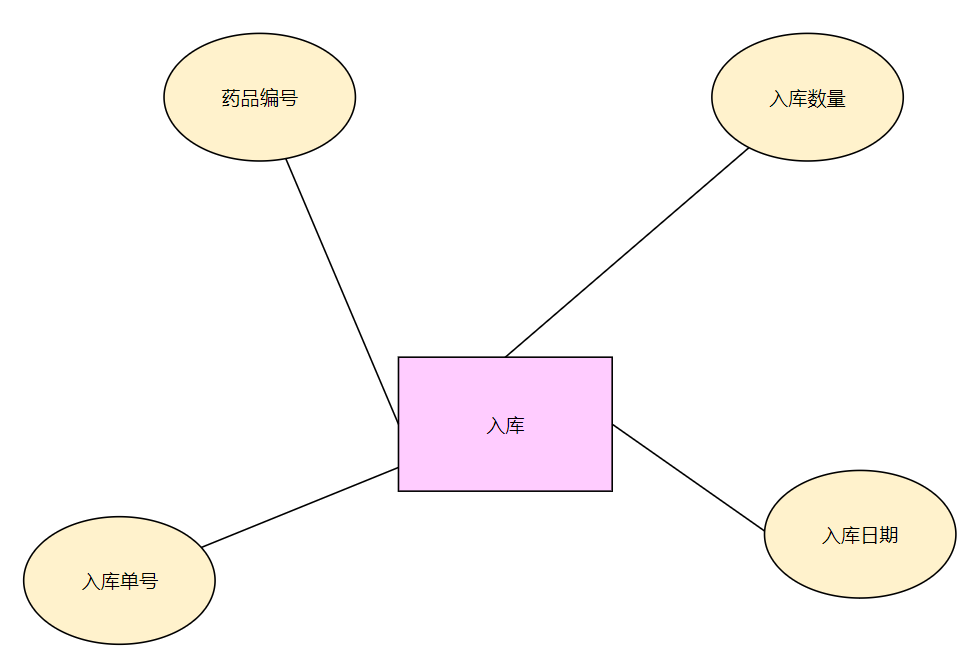




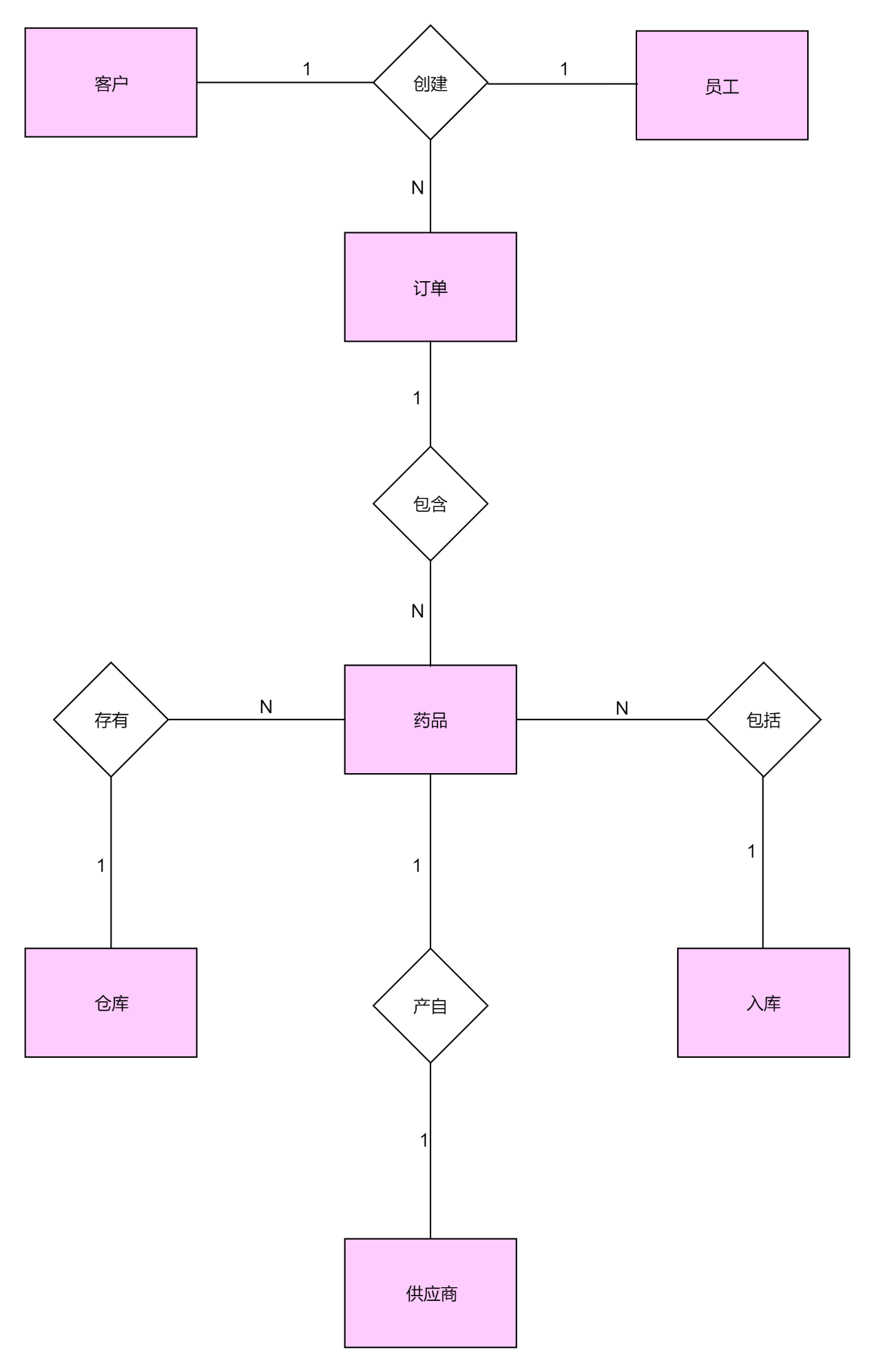








## 3.2 全局E-R图



# 四．逻辑结构设计

## 4.1 仓库表

仓库表：仓库（药品编号、药品数量）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 存储结构 | 是否可为空 | 是否主键 | 是否外键 |
| 药品编号 | 仓库药品编号 | varchar(8) | 不可 | 是 |  |
| 药品数量 | 药品数量 | int | 不可 |  |  |

## 4.2 订单表

订单表：订单（订单号、下单日期、总额、订单状态、药品编号、数量、客户编号）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 存储结构 | 是否可为空 | 是否主键 | 是否外键 |
| 订单号 | 订单号 | varchar(8) | 不可 | 是 |  |
| 下单日期 | 下单日期 | date | 不可 |  |  |
| 总额 | 订单总额 | float(53) | 不可 |  |  |
| 订单状态 | 订单状态 | varchar(1) | 不可 |  |  |
| 药品编号 | 订单药品编号 | varchar(8) | 不可 |  | 是 |
| 数量 | 订单药品数量 | int | 不可 |  |  |
| 客户编号 | 订单客户编号 | varchar(8) | 不可 |  | 是 |

## 4.3 供应商表

供应商表：供应商（供应商编号、供应商名称、供应商联系方式、供应商地址）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 存储结构 | 是否可为空 | 是否主键 | 是否外键 |
| 供应商编号 | 供应商编号 | varchar(8) | 不可 | 是 |  |
| 供应商名称 | 供应商名称 | varchar(90) | 不可 |  |  |
| 供应商联系方式 | 供应商联系方式 | varchar(90) | 不可 |  |  |
| 供应商地址 | 供应商地址 | varchar(90) | 不可 |  |  |

## 4.4 客户表

客户表：客户（客户编号、客户姓名、客户联系方式、客户登录密码、客户地址）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 存储结构 | 是否可为空 | 是否主键 | 是否外键 |
| 客户编号 | 客户编号 | varchar(8) | 不可 | 是 |  |
| 客户姓名 | 客户姓名 | varchar(30) | 不可 |  |  |
| 客户联系方式 | 客户联系方式 | varchar(30) | 不可 |  |  |
| 客户登录密码 | 客户登录密码 | varchar(30) | 不可 |  |  |
| 客户地址 | 客户地址 | varchar(90) | 不可 |  |  |

## 4.5 入库表

入库表：入库（入库单号、药品编号、入库数量、入库日期）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 存储结构 | 是否可为空 | 是否主键 | 是否外键 |
| 入库单号 | 入库单号 | varchar(8) | 不可 | 是 |  |
| 药品编号 | 入库药品编号 | varchar(8) | 不可 |  | 是 |
| 入库数量 | 入库数量 | int | 不可 |  |  |
| 入库日期 | 入库日期 | date | 不可 |  |  |

## 4.6 药品表

药品表：药品（药品编号、药品名称、药品功效、药品单价、有效期、药品类型、供应商编号）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 存储结构 | 是否可为空 | 是否主键 | 是否外键 |
| 药品编号 | 药品编号 | varchar(8) | 不可 | 是 |  |
| 药品名称 | 药品名称 | varchar(90) | 不可 |  |  |
| 药品功效 | 药品功效 | varchar(90) | 不可 |  |  |
| 药品单价 | 药品单价 | float(53) | 不可 |  |  |
| 有效期 | 有效期 | varchar(30) | 不可 |  |  |
| 药品类型 | 药品类型 | varchar(30) | 不可 |  |  |
| 供应商编号 | 药品供应商编号 | varchar(8) | 不可 |  | 是 |

## 4.7 员工表

员工表：员工（员工照片、员工工号、员工姓名、员工联系方式、员工登录密码、员工职务、是否管理员）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 存储结构 | 是否可为空 | 是否主键 | 是否外键 |
| 员工照片 | 员工照片 | image | 可 |  |  |
| 员工工号 | 员工工号 | varchar(8) | 不可 | 是 |  |
| 员工姓名 | 员工姓名 | varchar(30) | 不可 |  |  |
| 员工联系方式 | 员工联系方式 | varchar(30) | 不可 |  |  |
| 员工登录密码 | 员工登录密码 | varchar(30) | 不可 |  |  |
| 员工职务 | 员工职务 | varchar(30) | 不可 |  |  |
| 是否管理员 | 指示是否为管理员 | varchar(1) | 不可 |  |  |

## 4.8 范式分析

在本系统中，各关系的主码均只包含一个主属性，并且不存在多值依赖；非主属性对码的部分函数依赖、传递函数依赖已消除，主属性对码的部分和传递函数依赖也已消除，故关系模式应为4NF。

# 五．物理结构设计

## 5.1 表的存储与实现

CREATE DATABASE drugsystem

ON PRIMARY

(NAME = 'tsg\_data',

FILENAME =

'C:\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\tsg\_data.mdf',

SIZE = 5MB,

MAXSIZE = 500MB,

FILEGROWTH = 10%)

LOG ON

(NAME = 'tsg\_log',

FILENAME =

'C:\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\tsg\_log.ldf',

SIZE = 3MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1MB)

-- ----------------------------

-- Table structure for 仓库

-- ----------------------------

GO

CREATE TABLE 仓库 (

药品编号 varchar(8) NOT NULL ,

药品数量 int NOT NULL DEFAULT (0)

)

GO

-- ----------------------------

-- Table structure for 订单

-- ----------------------------

GO

CREATE TABLE 订单 (

订单号 varchar(8) NOT NULL ,

下单日期 date NOT NULL ,

药品编号 varchar(8) NOT NULL ,

数量 int NOT NULL ,

总额 float(53) NOT NULL ,

订单状态 varchar(1) NOT NULL DEFAULT ('0') CHECK(订单状态 in ('0','1')),

客户编号 varchar(8) NOT NULL

)

GO

-- ----------------------------

-- Table structure for 供应商

-- ----------------------------

GO

CREATE TABLE 供应商 (

供应商编号 varchar(8) NOT NULL ,

供应商名称 varchar(90) NOT NULL ,

供应商联系方式 varchar(90) NOT NULL ,

供应商地址 varchar(90) NOT NULL

)

GO

-- ----------------------------

-- Table structure for 客户

-- ----------------------------

GO

CREATE TABLE 客户 (

客户编号 varchar(8) NOT NULL ,

客户姓名 varchar(30) NOT NULL ,

客户联系方式 varchar(30) NOT NULL ,

客户登录密码 varchar(30) NOT NULL ,

客户地址 varchar(90) NOT NULL

)

GO

-- ----------------------------

-- Table structure for 入库

-- ----------------------------

GO

CREATE TABLE 入库 (

入库单号 varchar(8) NOT NULL ,

药品编号 varchar(8) NOT NULL ,

入库数量 int NOT NULL ,

入库日期 date NOT NULL

)

GO

-- ----------------------------

-- Table structure for 药品

-- ----------------------------

GO

CREATE TABLE 药品 (

药品编号 varchar(8) NOT NULL ,

药品名称 varchar(90) NOT NULL ,

药品功效 varchar(90) NULL ,

药品单价 float(53) NOT NULL ,

有效期 varchar(30) NOT NULL ,

药品类型 varchar(30) NOT NULL ,

供应商编号 varchar(8) NOT NULL

)

GO

-- ----------------------------

-- Table structure for 员工

-- ----------------------------

GO

CREATE TABLE 员工 (

员工照片 image default NULL,

员工工号 varchar(8) NOT NULL ,

员工姓名 varchar(30) NOT NULL ,

员工联系方式 varchar(30) NOT NULL ,

员工登录密码 varchar(30) NOT NULL ,

员工职务 varchar(30) NOT NULL ,

是否管理员 varchar(1) NOT NULL DEFAULT ('0') CHECK(是否管理员 in ('0','1'))

)

GO

## 5.2 视图、触发器与存储过程实现

为方便软件功能实现，部分相关视图、触发器与存储过程的设计是必要的。

-- ----------------------------

-- View structure for 仓库查询

-- ----------------------------

GO

CREATE VIEW 仓库查询 AS

SELECT

仓库.药品编号,

药品.药品名称,

仓库.药品数量

FROM 仓库

INNER JOIN 药品 ON 仓库.药品编号 = 药品.药品编号

GO

-- ----------------------------

-- View,procedure structure for 当月统计

-- ----------------------------

GO

CREATE VIEW 当月统计 AS

SELECT

订单.下单日期,

药品.药品名称,

订单.数量,

订单.总额

FROM 订单

INNER JOIN 药品 ON 订单.药品编号 = 药品.药品编号

WHERE

订单.订单状态 = 1

and MONTH(下单日期)=MONTH(GETDATE());

GO

GO

CREATE PROCEDURE 月统计

@订单数量 int, @销售额 float(53)

AS

BEGIN

SET @订单数量=(SELECT COUNT(\*) FROM 当月统计)

SET @销售额=(SELECT COUNT(总额) FROM 当月统计)

END

GO

-- ----------------------------

-- View structure for 管理员登录

-- ----------------------------

GO

CREATE VIEW 管理员登录 AS

SELECT

员工.员工工号 AS 管理员工号,

员工.员工姓名 AS 管理员姓名,

员工.员工登录密码 AS 管理员登录密码

FROM 员工

WHERE

员工.是否管理员 = 1

GO

-- ----------------------------

-- View structure for 客户登录

-- ----------------------------

GO

CREATE VIEW 客户登录 AS

SELECT

客户.客户编号,

客户.客户姓名,

客户.客户登录密码

FROM 客户

GO

-- ----------------------------

-- View structure for 客户看订单

-- ----------------------------

GO

CREATE VIEW 客户看订单 AS

SELECT

订单.订单号,

订单.下单日期,

订单.药品编号,

药品.药品名称,

客户.客户姓名,

客户.客户地址,

订单.客户编号,

订单.数量,

订单.总额,

订单.订单状态

FROM 订单

INNER JOIN 药品 ON 订单.药品编号 = 药品.药品编号

INNER JOIN 客户 ON 订单.客户编号 = 客户.客户编号

GO

-- ----------------------------

-- View structure for 员工登录

-- ----------------------------

GO

CREATE VIEW 员工登录 AS

SELECT

员工.员工工号,

员工.员工姓名,

员工.员工登录密码

FROM 员工

GO

-- ----------------------------

-- View structure for 在售药品

-- ----------------------------

GO

CREATE VIEW 在售药品 AS

SELECT

药品.药品编号,

药品.药品名称,

药品.药品功效,

药品.药品单价,

药品.有效期,

药品.药品类型,

仓库.药品数量 AS 药品余量

FROM

药品

INNER JOIN 仓库 ON 仓库.药品编号 = 药品.药品编号

WHERE

仓库.药品数量 > 0

GO

-- ----------------------------

-- Triggers structure for table 入库

-- ----------------------------

GO

CREATE TRIGGER 入库触发器

ON 入库

AFTER INSERT

AS

declare @数量 int,@药品 varchar(8)

--在inserted表中查询已经插入记录信息

select @数量 = 入库数量,@药品 = 药品编号 from inserted;

update 仓库 set 药品数量 = 药品数量 + @数量 where 药品编号=@药品;

GO

GO

CREATE TRIGGER 退货触发器

ON 入库

AFTER DELETE

AS

declare @数量 int,@药品 varchar(8)

--在inserted表中查询已经插入记录信息

select @数量 = 入库数量,@药品 = 药品编号 from deleted;

update 仓库 set 药品数量 = 药品数量 - @数量 where 药品编号=@药品;

GO

-- ----------------------------

-- Triggers structure for table 药品

-- ----------------------------

GO

CREATE TRIGGER 仓库更新

ON 药品

AFTER INSERT

AS

declare @药品 varchar(8)

--在inserted表中查询已经插入记录信息

select @药品 = 药品编号 from inserted;

insert into 仓库(药品编号) values(@药品);

GO

## 5.3 数据库安全性设计

如前所述，本系统可分为客户、员工、管理员三种角色，具有不同的控制权限。管理员对整个数据库均具有操作权限，员工对仓库、药品、订单、供应商、客户、入库表具有操作权限，客户仅具有部分视图的查看权限与订单的插入权限。

GO

EXEC sp\_addlogin 'system','yww\_39485184'

GO

EXEC sp\_grantdbaccess 'system','客户'

GO

EXEC sp\_grantdbaccess 'system','员工'

GO

EXEC sp\_grantdbaccess 'system','管理员'

GO

EXEC sp\_addrolemember 'db\_owner', '管理员'

GO

GRANT SELECT,INSERT,UPDATE ON 仓库 TO 员工

GO

GRANT SELECT,INSERT,UPDATE ON 药品 TO 员工

GO

GRANT SELECT,INSERT,UPDATE ON 订单 TO 员工

GO

GRANT SELECT,INSERT,UPDATE ON 供应商 TO 员工

GO

GRANT SELECT,INSERT,UPDATE ON 客户 TO 员工

GO

GRANT SELECT,INSERT,UPDATE ON 入库 TO 员工

GO

GRANT SELECT ON 客户看订单 TO 客户

GO

GRANT SELECT ON 在售药品 TO 客户

GO

GRANT INSERT ON 订单 TO 客户

## 5.4 数据库完整性设计

在前述建表过程中，并未建立主码外码，这是因为数据库完整性设计被独立进行。代码如下。

-- ----------------------------

-- Primary Key structure for table 仓库

-- ----------------------------

ALTER TABLE 仓库 ADD PRIMARY KEY (药品编号)

GO

-- ----------------------------

-- Primary Key structure for table 订单

-- ----------------------------

ALTER TABLE 订单 ADD PRIMARY KEY (订单号)

GO

-- ----------------------------

-- Primary Key structure for table 供应商

-- ----------------------------

ALTER TABLE 供应商 ADD PRIMARY KEY (供应商编号)

GO

-- ----------------------------

-- Primary Key structure for table 客户

-- ----------------------------

ALTER TABLE 客户 ADD PRIMARY KEY (客户编号)

GO

-- ----------------------------

-- Indexes structure for table 入库

-- ----------------------------

-- ----------------------------

-- Primary Key structure for table 入库

-- ----------------------------

ALTER TABLE 入库 ADD PRIMARY KEY (入库单号)

GO

-- ----------------------------

-- Primary Key structure for table 药品

-- ----------------------------

ALTER TABLE 药品 ADD PRIMARY KEY (药品编号)

GO

-- ----------------------------

-- Primary Key structure for table 员工

-- ----------------------------

ALTER TABLE 员工 ADD PRIMARY KEY (员工工号)

GO

-- ----------------------------

-- Foreign Key structure for table 仓库

-- ----------------------------

ALTER TABLE 仓库 ADD FOREIGN KEY (药品编号) REFERENCES 药品 (药品编号) ON DELETE CASCADE ON UPDATE CASCADE

GO

-- ----------------------------

-- Foreign Key structure for table 订单

-- ----------------------------

ALTER TABLE 订单 ADD FOREIGN KEY (客户编号) REFERENCES 客户 (客户编号) ON DELETE CASCADE ON UPDATE CASCADE

GO

ALTER TABLE 订单 ADD FOREIGN KEY (药品编号) REFERENCES 药品 (药品编号) ON DELETE CASCADE ON UPDATE CASCADE

GO

-- ----------------------------

-- Foreign Key structure for table 入库

-- ----------------------------

ALTER TABLE 入库 ADD FOREIGN KEY (药品编号) REFERENCES 药品 (药品编号) ON DELETE CASCADE ON UPDATE CASCADE

GO

-- ----------------------------

-- Foreign Key structure for table 药品

-- ----------------------------

ALTER TABLE 药品 ADD FOREIGN KEY (供应商编号) REFERENCES 供应商 (供应商编号) ON DELETE CASCADE ON UPDATE CASCADE

GO

## 5.5 索引设计

索引设计部分也被独立出来，每个表都建立了相应索引。代码如下。

CREATE INDEX A ON 仓库(药品编号 ASC)

CREATE INDEX B ON 药品(药品编号 ASC)

CREATE INDEX C ON 订单(下单日期 DESC)

CREATE INDEX D ON 供应商(供应商编号 ASC)

CREATE INDEX E ON 客户(客户编号 ASC)

CREATE INDEX F ON 入库(入库单号 ASC)

CREATE INDEX G ON 员工(员工工号 ASC)

# 六．数据库完成效果

数据库部分完成效果如下所示。在软件设计阶段可能会有进一步的增加和调整。

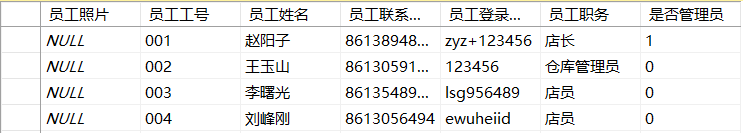














# 七．总结

药店进销存人管理系统后台数据库的设计最终结果较好地达到了预期的目标，并实现了功能子系统较多、不同用户权限区别较大的特点。通过这次数据库课程设计的实践，我也获得了较完整的数据库-后端系统设计实施经验。然而，由于是单人完成，囿于课余时间限制，也留下了一些遗憾。例如，未能为软件的每个功能子系统各自实现全部的视图、触发器、存储过程和函数来充分满足其功能，而将部分任务留给了后端实现；未能使员工、订单等表所具有的信息更完整，未能更多录入图片数据，未能更精细地保证数据库安全性，表间联系存在不足，软件系统设计不够完备，等等。

这些不足也是后续的改进方向。不过，设计实施计算机软件系统总是会有所取舍，总是会留下遗憾。取舍、斟酌的过程，大概也是我可以获得的一种开发经验吧！

# 八．参考资料

[1] 郑人杰，马素霞，王素琴，齐林海。《软件工程概论》（第3版）。北京机械工业出版社。2020。

[2] (美)凯 S.霍斯特曼。《Java核心技术》 （第10版）。2016。

[3] 王珊，萨师煊。《数据库系统概论》（第5版）。高等教育出版社。2014。

[4] 中国国家医保目录2020版。<https://www.yaopinnet.com/tools/yibao2020/index.htm>

[5] Sql Server image类型数据使用。<https://blog.csdn.net/tgbsqliuying/article/details/44306611>

[6] 数字字段varchar与int选择。<https://blog.csdn.net/banghu8816/article/details/101780625?utm\_medium=distribute.pc\_relevant.none-task-blog-baidujs\_title-0&spm=1001.2101.3001.4242>

[7] Sql Server—— 用户权限操作。<https://www.cnblogs.com/xulinjun/p/12000516.html>

# 九．致谢

在本次课程设计的过程中，毫无疑问首先应当感谢王文玉老师。没有她的尽心指导和多次答疑，我既不可能、也不会有足够的动力去尝试学习和研究数据库系统。同时，我也要感谢我学习资料的创作者，郑人杰、王珊、萨师煊、凯 S.霍斯特曼等老师，以及网民徐林俊、banghu8816、刘颖90，他们分享了自己所学，我等后辈因此而能自主学习、成长。

我还要感谢我的室友们和家人们，感谢他们的鼓励与陪伴、付出与汗水。