

三幺八医院管理信息系统实验报告

小组成员：2107 cyj、lx、lhq、why

一.库分析与的建立部分

1. 系统需求分析

1.1 需求概述和系统边界

随着医疗机构规模的扩大和病患人数的增多,医院信息管理已经成为每个医院必不可少的工作,使用数据库能够对信息进行更好的管理,本实验欲建立医院信息数据库,提供一种数据库设计的方法,以其达到便捷高效地对医院的信息进行管理。

医院管理信息系统是通过将医院中的病人、医生、科室、诊室、病房、病床药品、值班安排以及在治疗和住院过程中产生的挂号单、处方单、缴费单、用药清单、诊疗方案、住院记录和住院档案等信息存储到数据库中,用户能够在系统界面或登录系统后进行信息的查看和维护。同时系统可以根据诊疗方案等信息自动计算住院时的每日费用,实现住院管理的自动化、系统化,最大限度地为医生和医院提供方便和提高管理效率。本医院管理信息系统的需求描述如下:

该信息管理系统支持三类用户:病人、医生、管理员。在进行注册后,病人可以利用自己的病案号和密码登陆系统,在登陆系统后,可以进行挂号、缴费、就诊,查询就诊记录、本人治疗情况和费用明细的操作;医生可以利用管理员分配的工号和密码登陆系统,登陆系统后,可以查看自己的排班情况、接诊病人、开具处方、查看病人的治疗情况,并且主任医生还可统计不同科室的排班情况、不同医生的工作量情况等;而管理员可以维护医院的基本信息、科室的基本信息、

医生的基本信息、药品的基本信息等。

本实验将按照数据库设计流程设计医院管理信息系统数据库 E-R 图和关系模式，保存所需的全部信息，并高效地对上述功能实现及应用。

1.2 主要业务处理流程

医院管理信息系统的主要业务包括：门诊治疗、住院治疗、药品信息。我们只给出医院管理信息系统的核心业务“缴费”及“接诊”的处理流程，如图 1 所示：

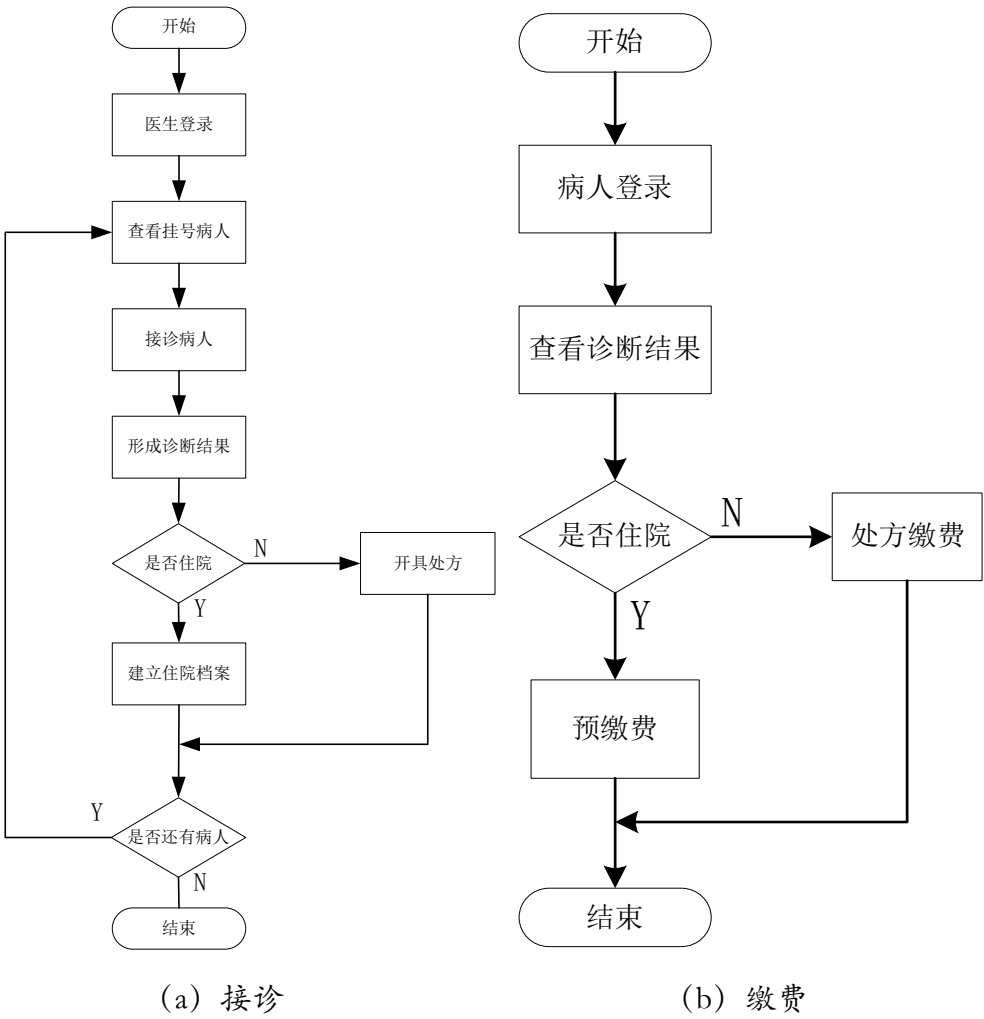


图 1 医院管理信息系统的主要业务流程

1.3 功能需求分析

根据上述需求概述和业务流程,通过与医院管理信息系统用户沟通,医院管理信息系统主要功能需求分析如下:

(1) **用户管理**。提供病人的基本信息录入、管理员对医生信息的录入以及查看用户本身信息等功能。包括:

- 病人注册信息录入、维护与查询;
- 管理员对医生基本信息的录入、维护和查询;
- 所有用户能够查看自身的基本信息;
- 病人能够查看自己的就诊记录和费用明细;

(2) **药品管理**。提供对药品信息的录入与维护,以及药品采购、入库、消耗等功能,包括:

- 药品信息查询:管理员能够查询到所有药品的具体信息。
 - 药品入库管理:管理员为新增的药品办理药品入库,并增加新药品信息、更新药品库存数量;
 - 药品消耗管理:当药品被使用或被购买后进行药品消耗,管理员进行药品的库存数量更新;

(3) **医生排班管理**。提供对医生进行排班以及统计工作量的功能,包括:

- 排班查询:医生能够查询到自己当日是否排班、排班的诊室及排班时间;
- 工作量统计:主任医生能够对医生接诊的病人数量及排班次数查看;

(4) **挂号管理**。能够提供挂号信息录入、查询的功能,包括:

- 挂号信息录入:病人在进行挂号后,系统将挂号信息进行录入;

- 挂号信息查询：医生能够对病人的挂号信息进行查询；

(5) **诊疗管理**。提供病人就诊、医生接诊、开具处方以及建立住院档案等功能。包括：

- 病人就诊：病人根据挂号单上的信息到特定的诊室由指定医生接诊；
- 医生接诊：医生能够根据病人的挂号时间顺序对病人进行接诊；
- 开具处方：医生能够根据对接诊的病人开具处方；

(6) **科室信息管理**。主要提供对科室信息的录入、维护和查询以及统计科室排班情况的功能。包括：

- 管理员能够对科室信息进行录入、查询和维护；
- 主任医生能够查询不同科室的排班情况；

(7) **缴费管理**。提供处方缴费、住院预缴费以及记录就诊和费用的功能。包括：

- 处方缴费：病人可以根据医生开具的处方单，进行处方缴费；
- 住院预缴费：需要住院的病人能够在住院前进行预缴费；
- 系统根据病人的挂号和处方所缴费用，记录本次就诊的所有费用；

(8) **住院管理**。提供对住院档案信息的录入、维护和查询，安排病床、开具诊疗方案以及存档住院记录的功能。包括：

- 医生能够根据诊断结果对住院病人建立住院档案，并能够进行后续维护和查询；
- 医生能够为需要住院的病人分配病床；
- 主治医生能够根据住院病人当天的情况建立诊疗方案；
- 系统能够将建立的诊疗方案存档为住院记录；

医院管理信息系统主要功能模块如下

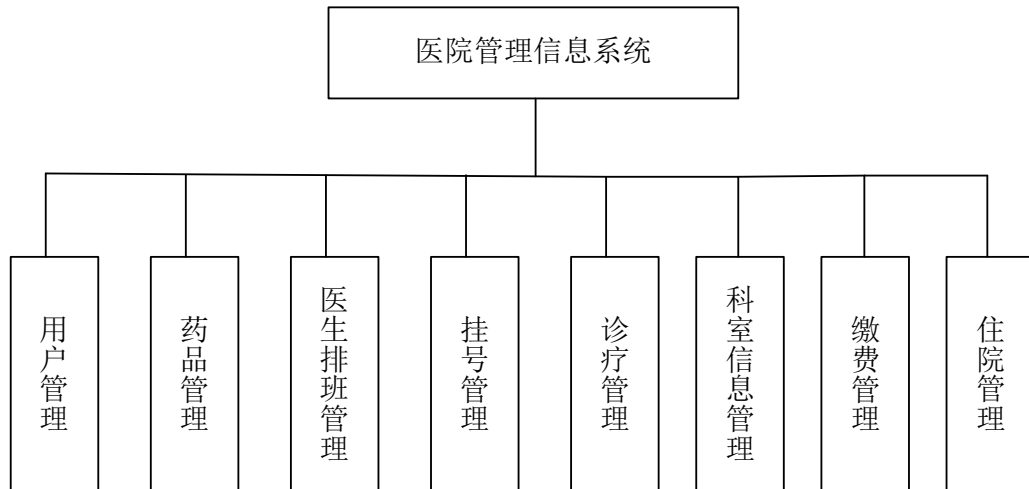


图 2 所示：

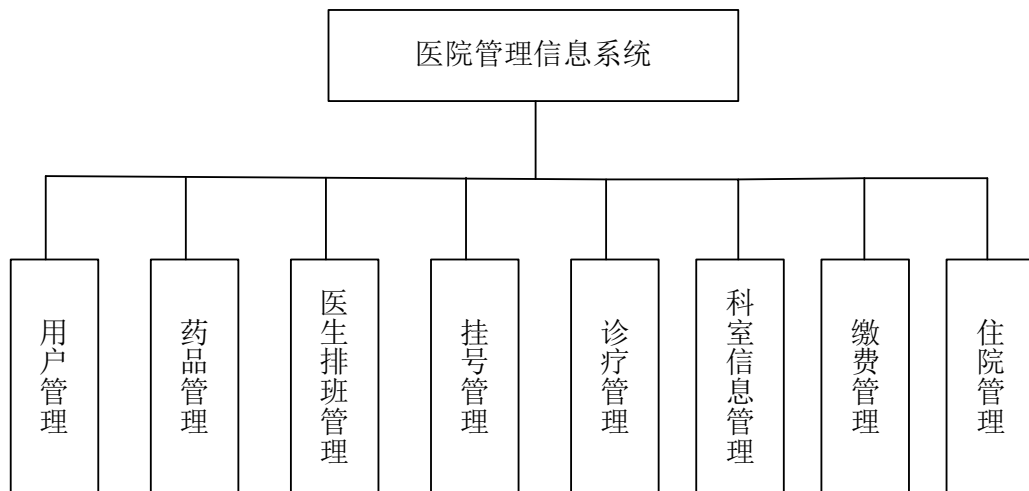


图 2 医院管理信息系统功能模块

1.4 数据需求分析

根据功能需求分析的结果，医院管理信息系统的数据需求分析如下：

(1) **病人信息**：包括病案号、密码、姓名、性别、电话号码、地址等信息。

当病人的所有信息填写正确后提示病人注册成功，并且返回病人的病案号（唯一标识）和密码等信息。

(2) **医生信息**：包括工号、登陆密码、姓名、性别、职称、电话号码等信息。系统检查医生填写的工号和登陆密码正确后能够进行登录，工号为医生的唯

一标识。医生的职称记录了医生是否为主任医生，主任医生有权限统计科室和其他医生的总体情况。

(3) **管理员信息**：包括工号、登陆密码、姓名、性别、电话号码等信息。系统检查管理员填写的工号和登陆密码正确后能够进行登录。管理员的工号为管理员的唯一标识。

(4) **挂号单**：需存储就诊单号、挂号时间等信息。就诊单号为挂号单的唯一标识，由系统按时间顺序生成。

(5) **处方单**：包括处方单号、症状描述、诊疗费用等信息。处方单号为处方单的唯一标识。诊疗费用包括了挂不同职称医生号的费用以及药品费用，由药品费用和挂号费派生而来。

(6) **药品**：包括药品编号、药品名、单价、库存数量、用法等信息。药品编号为药品的唯一标识。药品库存数量能够决定医生在开处方的时候能否开出该药品，病人进行处方缴费时药品实际上进行出库，当药品库存不够时，管理员需要进行库存管理。

(7) **用药清单**：包括清单号、药品总价等信息，清单号为用药清单的唯一标识。

(8) **诊室**：包括诊室号、诊室名称、诊室电话等信息。诊室编号为诊室的唯一标识。

(9) **科室**：包括科室号、科室名称、科室电话、主任医生等信息。科室号为科室的唯一标识。主任医生能够管理本科室的排班情况和本科室医生的排班情况。

(10) **住院档案**：包含档案号、入院时间、出院时间等信息，档案号为住院

档案的唯一标识。

(11) **住院记录**：它是依赖于住院档案的弱实体集，需存储记录号、病人情况、**诊疗方案**(包括方案号、方案细则、方案费用等)、记录时间等信息，记录号为部分码。

(12) **病房**：包含病房编号、地点以及收费标准等信息，其中，病房编号为病房的唯一标识。

(13) **病床**：它是依赖于病房的弱实体集，需要存储病床号和病床状态等信息，其中病床号为部分码。

(14) **缴费单**：包含了流水号、缴费状态、金额以及缴费日期，其中流水号为费用的唯一编号。

(15) **值班安排**：它是依赖于诊室的弱实体集，需存储值班时间信息，值班时间为部分码。

1.5 业务规则及完整性约束分析

基于上述功能需求和数据需求，通过进一步了解，医院管理系统规则及完整性约束如下：

- (1) 门诊分不同科室，一个科室有若干名医生，一位医生只属于一个科室；
- (2) 看医生时，病人根据挂号进入相应诊室；
- (3) 医生坐诊的时间不定，根据排班，有时坐诊，有时需要去住院部治疗住院病人（排班时，门诊坐诊时间和住院部巡诊时间不能冲突）；
- (4) 医生为病人治疗，开出处方（本实验忽略医疗检查环节）；处方内容包含症状描述及用药清单，清单中包括药品、价格、数量、用法等信息，还包括

诊疗费用，不同职称的医生诊疗费不同；

(5) 病人根据处方缴费（本实验假设一律采用线上支付）去药房取药；

(6) 药房药品有库存问题，无库存的药品不会出现在医生开处方上，药房管理人员可以查询库存情况（本实验不考虑药房业务）。

(7) 病人根据门诊部科室的诊断情况，决定是否住院（办理住院手续：建立住院档案）；

(8) 住院部不同科室有固定病房，病人住院时需要安排相应病房的病床；

(9) 住院部每位病人都有一位主治医生，而每一位医生可能给多名病人治病；

(10) 一位病人可能多次住院，每次住院都建一个住院档案，而一份住院档案只能记载一个病人的情况；

(11) 每间病房有多个床位，能住多位病人，而每一位病人只能安排在一间病房中的一个床位；

(12) 医生每天巡检病人情况，根据病人情况开具诊疗方案，并依据诊疗方案对病人进行治疗；每日诊疗方案需要存档（插入住院记录）；

(13) 病人办理住院手续时需要预缴纳住院费，计费系统根据每天的诊疗方案等信息计算当日费用（包括病房床位费等），不足时第二天停止医疗。

2. 数据库概念设计

2.1 确定基本实体集及属性

由 1.4 节的分析可知，医院管理系统中出现的“名词”主要有：病人、医生、管理员、挂号单、处方单、用药清单、药品、值班安排、诊室、科室、住院档案、

住院记录、病房、病床等。

显然，病人、医生、用药清单、药品、诊室、科室、病房等都是对应为有形的人、物或单位，且都具有一组属性且部分属性能唯一标识每个实体，而且它们都需要存储到数据库中供查询用，因此可以直接建模为基本实体集。

医院的基本信息包括名称、咨询电话、位置等，由于本系统并不与其他医院进行信息共享，只对某一医院的内部信息进行管理，因此，我们不必将医院建模为一个实体集，我们将直接把医院的信息显示在系统界面中，既能够达成管理医院信息的要求，也可以达到突出显示的效果。

我们将伴随着业务发生而形成的处方单、缴费单、住院档案、诊疗方案、挂号单、值班安排、住院记录、病床等建模为依赖实体集或弱实体集。我们将这些伴随着主要业务建模而形成的依赖实体集、弱实体集都放在 2.2 节主要业务局部概念建模中去分析。

综上所述，病人、医生、用药清单、药品、诊室、科室、病房等可建模为基本实体集。我们将与应用相关的特征建模为实体集的属性。

各基本实体集的属性定义如下：

(1) 病人(Patient)实体集。其属性有：病案号(patNo)、密码(patPassword)、姓名(patName)、性别(sex)、电话号码(telephone)、地址(address)等。表 1 为病人实体集的数据字典。

表 1 病人(Patient)实体集的数据字典

属性名	含义	类别	域及约束
patPassword	密码		char(20)，不能少于 6 位；不可以取空值；默认为 123456
<u>patNo</u>	病案号	主码	char(13)，由大写字母 P 加上 12 位数字组成，其中数字为入院年份、月

			份、日，后按当日入院时间顺序记录 病人四位编号
patName	姓名		varchar(10)，不允许取空值
sex	性别		char(2)，取值范围：{'男'，'女'}
telephone	电话号码		char(11)，由数字字符组成
address	地址		varchar(100)，不允许取空值

(2) **医生(Doctor)**实体集。其属性有：工号(docNo)、登陆密码(docPassword)、姓名(docName)、性别(sex)、职称(degree)、电话号码(telephone)、诊疗费用(fee)等。

表 2 为医生实体集的数据字典。

表 2 医生(Doctor)实体集的数据字典

属性名	含义	类别	域及约束
<u>docNo</u>	工号	主码	char(10)，不允许取空值；由大写字母 D 加 9 位数字组成，前四位数字表示入职年份，后两份数字表示入职月份，后三位数值按入职的时间顺序递增表示。
docPassword	登录密码		char(20)，不能少于 6 位；不能为空
docName	姓名		varchar(10)，不允许取空值
sex	性别		char(2)，取值范围：{'男'，'女'}
degree	职称		varchar(20)
telephone	电话号码		char(11)，由数字字符组成
fee	诊疗费用		int

(3) **用药清单(MedicationList)**实体集。其属性有：清单号(recipeNo)、药品总价(medicineTotal)等。表 3 为用药清单实体集的数据字典。

表 3 用药清单(MedicationList)实体集的数据字典

属性名	含义	类别	域及约束
listNo	清单号	主码	char(7)，不允许取空值
medicineTotal	药品总价	派生	numeric，从具有联系集中得到

(4) **药品**(Medicine)实体集。其属性有：药品编号(medicineNo)、药品名(medName)、单价(price)、库存数量(medLeft)、用法(usage)等。表 4 为药品实体集的数据字典。

表 4 药品(Medicine)实体集的数据字典

属性名	含义	类别	域及约束
<u>medicineNo</u>	药品编号	主码	char(10)，不允许取空值；又两位字母加 6 位数字组成，两位字母统一为 YP 表示“药品”，接着四位字母表示药品转正年份，最后两位数字为转正号
medName	药品名		varchar(30)，不允许取空值
price	单价		numeric
medLeft	库存数量		int
usage	用法		varchar(100)

(5) **诊室**(Room)实体集。其属性有：诊室号(roomNo)、诊室名称(roomName)、诊室电话(telephone)等。表 5 为诊室实体集的数据字典。

表 5 诊室(Room)实体集的数据字典

属性名	含义	类别	域及约束
<u>roomNo</u>	诊室号	主码	varchar(4)，不可为空值
roomLocation	诊室地点		varchar(100)
telephone	诊室电话		varchar(11)

(6) **科室**(Office)实体集。其属性有：科室号(officeNo)、科室名称(officeName)、科室电话(telephone)等。表 6 为科室实体集的数据字典。

表 6 科室(Office)实体集的数据字典

属性名	含义	类别	域及约束
<u>officeNo</u>	科室号	主码	Int，不可为空值
officeName	科室名称		varchar(20)
telephone	科室电话		varchar(11)

(7) **病房(Ward)**实体集。其属性有:病房编号(wardNo), 地点(address), 收费标准(chargeStand)等。表 7 为病房实体集的数据字典。

表 7 病房(Ward)实体集的数据字典

属性名	含义	类别	域及约束
<u>wardNo</u>	病房编号	主码	char(4), 不允许取空值
address	地点		varchar(10), 不允许取空值
chargeStand	收费标准		INT, 不允许取空值

2.2 主要业务局部概念建模

由需求分析可知, 医院管理系统中的主要业务有: 病人挂号、门诊治疗、开具处方、取药、值班安排、建立住院档案、病床安排、诊疗方案开具、存档诊疗方案等。下面分别对它们进行建模分析。

1. 病人挂号

病人挂号操作涉及**医生、病人**基本实体集, 并会伴随着生成**挂号单**。根据依赖约束的分析可知, 伴随着“挂号”业务而形成的**挂号单(BookingForm)**需要单独建模为**依赖实体集**, 属性有: 就诊单号(bookingNo)、挂号时间(bookingTime)、挂号状态(bookingStatu)等, 挂号联系集为挂号单与病人之间的多对一的联系集。同时, 医生实体集与挂号单实体集之间存在着一对多的根据业务, 挂号单的数据字典如下表 8 所示:

表 8 挂号单(BookingForm)实体集的数据字典

属性名	含义	类别	域及约束
<u>bookingNo</u>	就诊单号	主码	char(13), 不可为空
bookingTime	挂号时间	联系	Date, 默认为当前时间

bookingStatu	挂号状态		varchar(10)
--------------	------	--	-------------

挂号业务的 E-R 图如下图 3 所示：

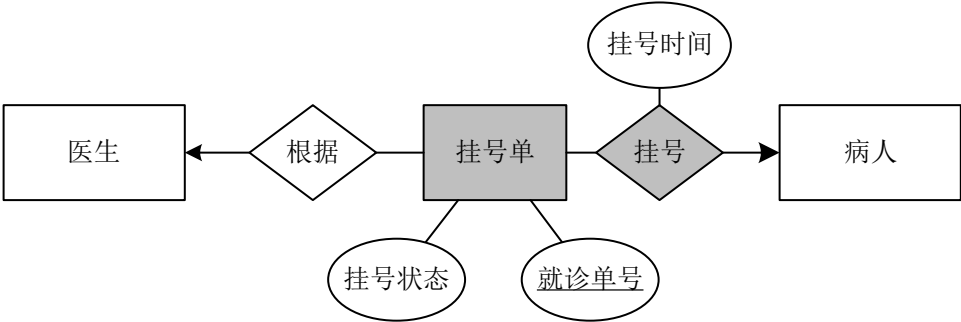


图 3 病人挂号业务的建模

2. 门诊治疗与开具处方

门诊治疗涉及**医生**和**病人**基本实体集，医生和病人之间存在着一对多的**接诊**联系集。伴随着开具处方业务生成的处方单(Recipe)需要单独建模为**依赖实体集**，其属性有：处方单号(recipeNo)、症状描述(symptom)、诊疗费用(money)等。开具处方应该是接诊联系集与处方单实体集之间的联系集，而不是医生实体集或病人实体集之间的联系集，所以我们将联系实体集治疗参与到开具处方联系集中，于是开具处方联系集为治疗联系实体集与依赖实体集处方单之间一对一的联系集。

处方单实体集的数据字典如下表 9 所示：

表 9 处方单(Recipe)实体集的数据字典

属性名	含义	类别	域及约束
<u>recipeNo</u>	处方单号	主码	char(7)，不可为空值
symptom	症状描述		varchar(100)
money	诊疗费用	派生	Numeir，由接诊费用和药品费用组成

门诊治疗与开具处方业务的 E-R 图如下图 4 所示：

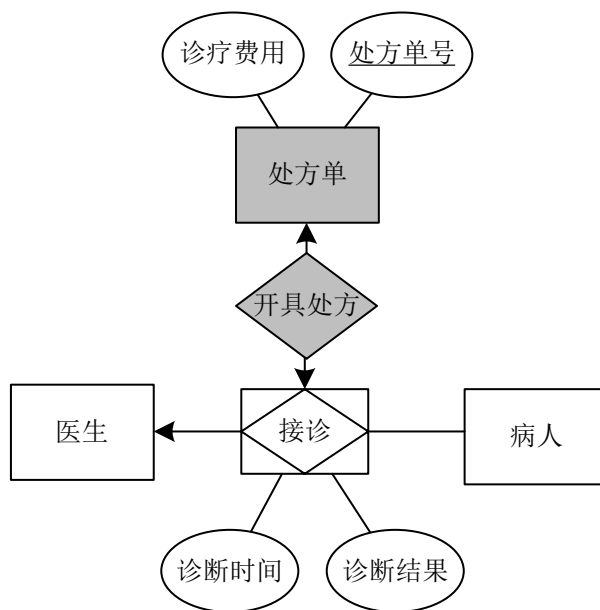


图 4 门诊治疗与开具处方业务的建模

3. 取药

取药业务涉及了**药品**这一基本实体集，另外，由于**缴费单**是伴随着**生成**这一业务而形成的，所以我们需要将缴费单(Payment)建立为依赖实体集，其属性有：流水号(paymentNo)、金额(fee)、缴费状态(paymentState)、缴费日期(paymentDate)等，其数据字典如下表 10 所示：

表 10 缴费单(Payment)实体集的数据字典

属性名	含义	类别	域及约束
<u>paymentNo</u>	流水号	主码	char(16)，不允许取空值
fee	金额		float，不允许取空值
paymentState	缴费状态		char(6)，仅可从“已缴费”、“未缴费”中选择
paymentDate	缴费日期		date

缴费单和处方单具有一对一的生成联系集，通过对缴费单进行缴费，能够根据处方单等取得对应的药品。取药业务的 E-R 图如下图 5 所示：

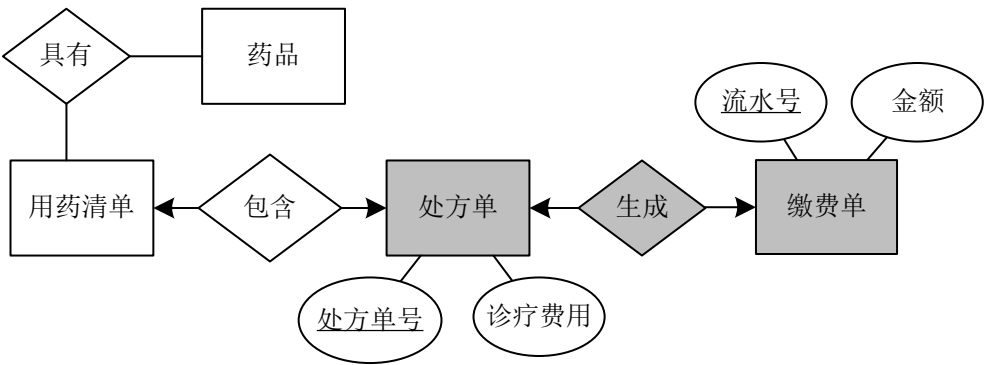


图 5 取药业务的建模

4. 值班安排

值班安排业务涉及了医生和诊室两个基本实体集，伴随着排时间业务形成了值班安排这一依赖实体集，医生实体集和值班安排实体集之间存在着一对多的排时间联系集，同时值班安排是依附于诊室的弱实体集，排诊室是存在于值班安排弱实体集与诊室基本实体集之间的一对多的标识联系集，因此我们将值班安排建模为弱实体集和依赖实体集。值班安排(DutyArrangement)弱实体集属性有：值班日期(dutyDate)等，其数据字典如下表 11 所示：

表 11 值班安排(DutyArrangement)弱实体集的数据字典

属性名	含义	类别	域及约束
dutyDate	值班日期	部分码	Date，不可为空
dutyShift	值班班次		ENUM

值班安排业务的 E-R 图如下图 6 所示：

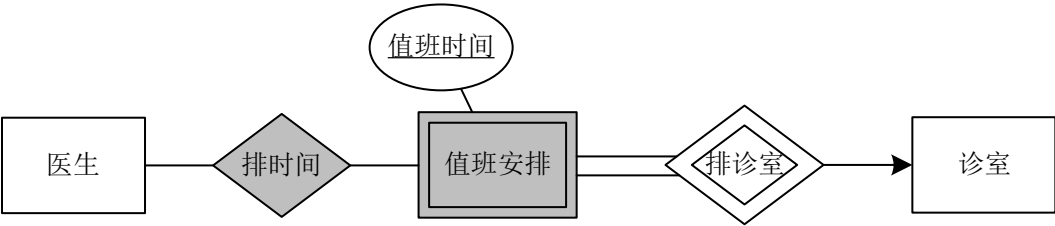


图 6 值班安排业务的建模

5. 建立住院档案及病床安排

建立住院档案业务涉及**接诊**这一联系实体集,并且伴随着**建档**业务会产生住院档案**依赖实体集**,建档为住院档案实体集与联系实体集接诊之间的一对一联系集。依赖实体集住院档案(DepartDocument)属性有:档案号(departNo)、入院时间(indepartDate)、出院时间(outdepartDate)等,依赖实体集住院档案的数据字典如下表 12 所示:

表 12 住院档案(DepartDocument)实体集的数据字典

属性名	含义	类别	域及约束
<u>departNo</u>	档案号	主码	char(16), 不允许取空值
indepartDate	入院时间		date, 不允许取空值
outdepartDate	出院时间		date, 可以取空值

病床安排业务涉及基本实体集**病房**以及依赖实体集**住院档案**,由于病床依附于病房实体集,因此我们需要将**病床**建模为弱实体集。病床(Bed)弱实体集属性有:病床编号.bedNo)、病床状态.bedState)等,其数据字典如下表 13 所示:

表 13 病床(Bed)弱实体集的数据字典

属性名	含义	类别	域及约束
<u>bedNo</u>	病床编号	部分码	char(3), 不允许取空值
bedState	病床状态		char(4), 仅可从“有人”, “空闲”中选择

建立住院档案及病床安排业务的 E-R 图如下图 7 所示：

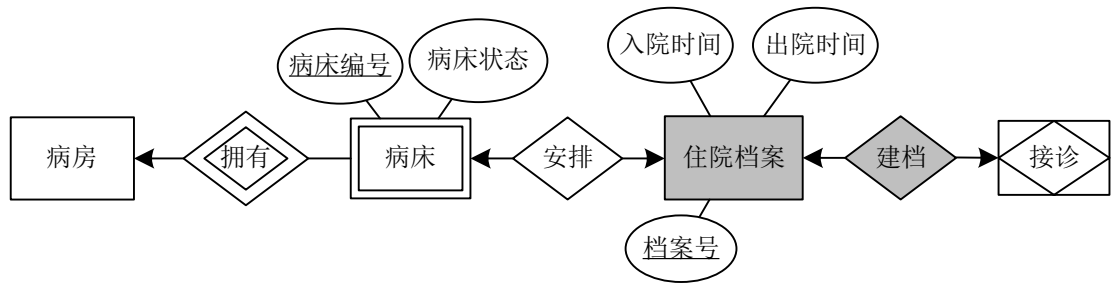


图 7 建立住院档案及病床安排业务的建模

6. 开具诊疗方案

开具治疗方案业务涉及医生和病人基本实体集,由于诊疗方案的开具业务是基于病人和医生的多对一联系集**巡检**,而不是医生和诊疗方案或者病人和诊疗方案之间,所以我们将巡检设定为**联系实体集**。同时,诊疗方案这一实体集是伴随开具这个关于联系实体集巡检以及治疗方案的一对多的业务产生的,因此我们将诊疗方案(Treatment)设定为**依赖实体集**,其属性为: 方案号(planNo)、方案描述(description)、收费标准(money)。诊疗方案实体集的数据字典如下表 14 所示:

表 14 诊疗方案(Treatment)依赖实体集的数据字典

属性名	含义	类别	域及约束
<u>planNo</u>	方案编号	主码	char(5), 不允许为空值
description	方案描述		varchar(100), 可以为空
money	收费标准		float

开具诊疗方案业务的 E-R 图如下图 8 所示：

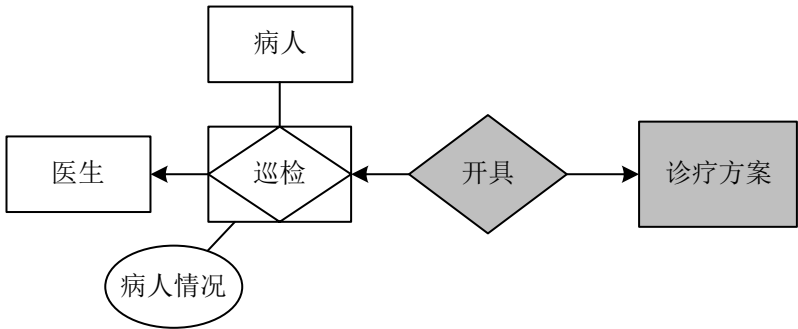


图 8 开具诊疗方案业务的建模

7. 诊疗方案存档

诊疗方案存档这一业务涉及了**诊疗方案**以及**住院档案**这两个依赖实体集。由于一份住院档案中会包含多份住院记录,住院记录是依附于住院档案的弱实体集,因此我们将住院记录(DepartRecord)建立为弱实体集,其属性为:记录号(aidNo)、病人状况(patSymptom)。同时根据业务约束我们知道住院记录是由诊疗方案存档得到的,也即诊疗方案和住院记录之间存在着一对一的存档联系。住院记录((DepartRecord))弱实体集的数据字典如下表 15 所示:

表 15 住院记录弱实体集的数据字典

属性名	含义	类别	域及约束
<u>aidNo</u>	记录号	部分码	int(3), 从 001 开始, 由入院日期至当日日期差计算, 不允许取空值
patSymptom	病人状况		varchar(100)

诊疗方案存档业务的 E-R 图如下图 9 所示:



图 9 诊疗方案存档业务的建模

2.3 定义联系集及属性

基于前面涉及得到的实体集,可确定如下联系集。

(1) **具有(Have)联系集**: **用药清单**实体集和**药品**实体集之间的多对多联系集,该联系集没有描述属性。

(2) **排时间(ScheduleTime)联系集**:**医生**实体集和**值班安排**实体集之间的多对多联系集,该联系集没有描述属性。

(3) **接诊(Reception)**联系集：**医生**实体集和**病人**实体集之间的一对多联系集。

描述属性有：诊断结果(result)、诊断时间(examTime)。接诊联系集的数据字典如表 16 所示：

表 16 接诊(Receive)联系集的数据字典

属性名	含义	类别	域及约束
result	诊断结果		varchar(100)
examTime	诊断时间		date

(4) **巡检(Inspect)**联系集：**医生**实体集和**病人**实体集之间的一对多联系集，其描述属性有：病人情况(patCondition)。

(5) **记录(Record)**标识实体集：**住院档案**实体集与**住院记录**弱实体集之间的一对多联系集，没有联系属性。

(6) **存档(Save)**联系集：**治疗方案**实体集和**住院记录**弱实体集之间的一对一联系集，没有联系属性。

(7) **包含(Contain)**联系集：**处方单**实体集和**用药清单**实体集之间的一对一联系集，没有联系属性。

(8) **生成(Build)**联系集：**处方单**实体集和**缴费单**实体集之间的一对一联系集，没有联系属性。

(9) **开具处方(Prescribe)**联系集：**处方单**实体集与**接诊**联系集之间的一对一联系集，没有联系属性。该联系通过聚合表示。

(10) **缴费(Pay)**联系集：**病人**实体集和**缴费单**实体集之间的一对多联系集。其描述属性为：缴费日期(paymentDate)，已建模为缴费单实体集的属性。

(11) **建档(Document)**联系集：**住院档案**实体集与**接诊**联系集之间的一对一

联系集，没有联系属性，该联系通过聚合表示。

(12) **安排(Arrange)**联系集：**住院档案**实体集与**病床**弱实体集之间的一对一联系集，没有联系属性。

(13) **置放(Place)**标识实体集：**病房**实体集和**病床**弱实体集之间的一对多联系集，没有联系属性。

(14) **拥有(Possess)**联系集：**科室**实体集与**病房**实体集之间的一对多联系集，没有联系属性。

(15) **属于(Belong)**联系集：**科室**实体集与**医生**实体集之间的一对多联系集，没有联系属性。

(16) **根据(Base)**联系集：**医生**实体集与**挂号单**实体集之间的一对多联系集，没有联系属性。

(17) **挂号(Register)**联系集：**病人**实体集与**挂号单**弱实体集之间的一对多联系集。其描述属性有：挂号时间(RegisterTime)，已建模为挂号单实体集的属性。

(18) **就诊(Visit)**联系集：**诊室**实体集与**病人**实体集之间的一对多联系集，没有联系属性。

(19) **排诊室(ConsultateRoom)**标识实体集：**诊室**实体集和**值班安排**弱实体集之间的一对多联系集，没有联系属性。

(20) **开具(issued)**联系集：**治疗方案**实体集与**巡检**联系集之间的一对一联系集，没有联系属性，该联系通过聚合表示。

2.4 完整 E-R 模型

综上所述,包括全部实体集、联系集及其描述属性的 E-R 图如下图 10 所示。

图中省略了大部分实体集属性，实体集的属性通过数据字典精确定义，我们主要用 E-R 图来描述实体集、联系集及其联系属性。

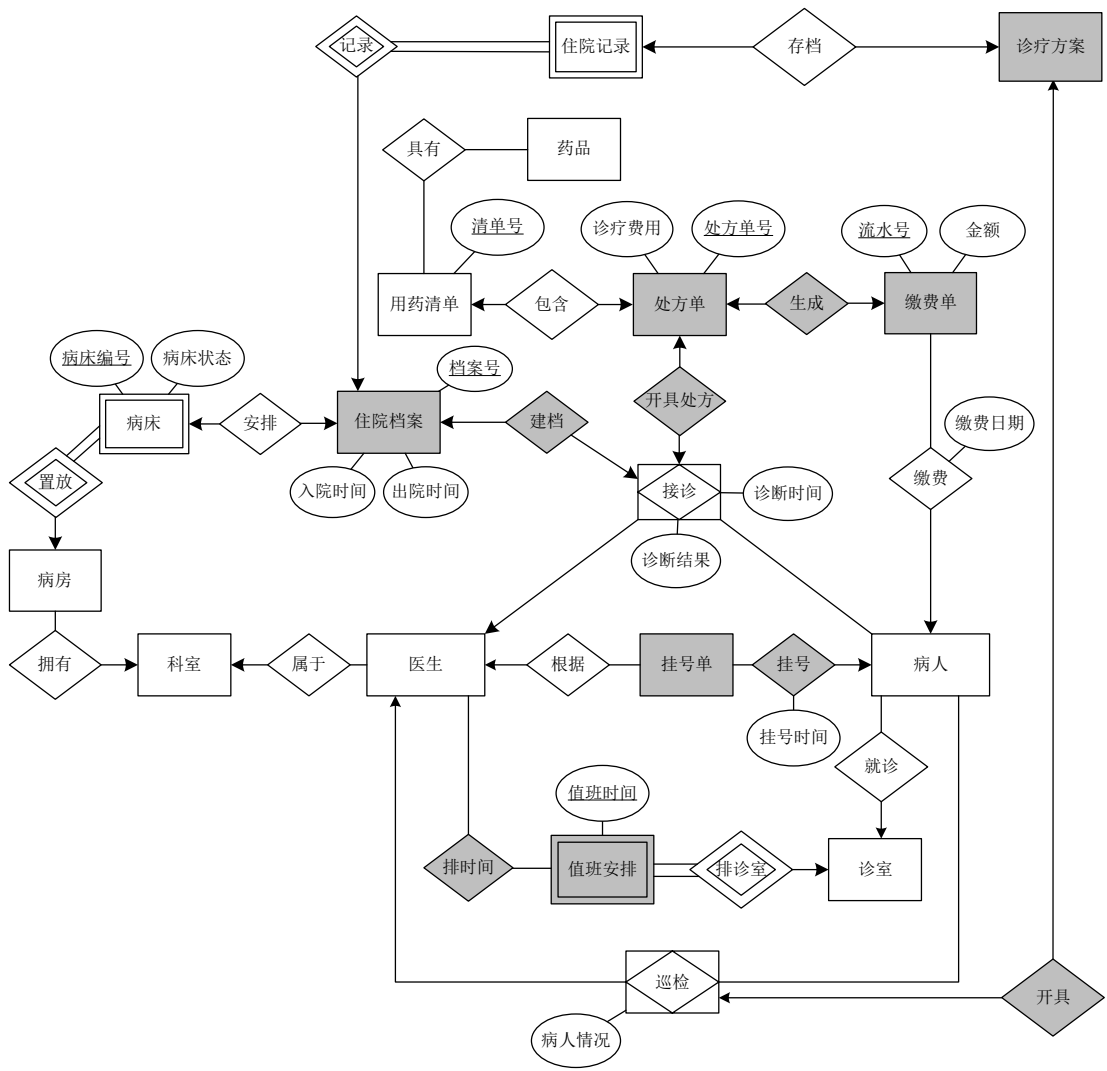


图 10 医院管理信息系统总 E-R 图

2.5 检查是否满足需求

经检查，图 10 所示的 E-R 图已基本包含了全部需求信息描述。但是，仍然发现还存在一些问题。

(1) 数据冗余。职称信息、诊疗费用在每个医生中都冗余存储，可以考虑将它独立出来，单独建立一个医生职称(Degree)实体集，属性有职称(degree)、诊疗费(fee)等，医生和医生职称实体集之间存在着多对一的引用(Citation)联系集，

如下图 11 所示：

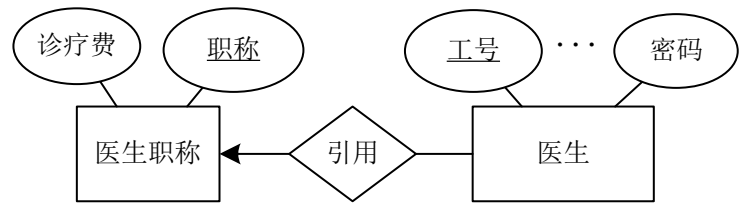


图 11 医生实体集与医生职称实体集之间的引用联系集

（2）业务规则脱离现实需求。例如，对于“病人挂号”业务，现规定的业务规则是：

- 一个病人能够挂号多次，且一次挂号只能由一个病人进行；
- 一个医生能够接受挂号单多份，而一份挂号单只能由一个医生接受；
- 医生能够自由选择挂号单来接诊对应病人。

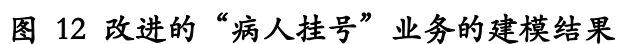
实际中，病人在进行挂号的同时，是可以自由选择医生的，所以，挂号操作能够同时确定医生，而不是医生选择对应的挂号，显然我们上面的业务规则不符合现实需求。我们将“病人挂号”业务的业务规则修改为：

- 一个病人能够挂号多次，且一次挂号只能由一个病人进行；
- 病人在挂号的同时也选择医生；
- 一个医生能够接受挂号单多份，而一份挂号单只能由一个医生接受。

分析该业务规则可以知道，病人实体集和挂号单实体集之间的一对多挂号联系集的语义没有变化，医生和挂号单之间存在着一对多的“根据”联系，且联系的语义存在着显著变化，需要达到在确定挂号单的同时便确定了挂号的医生，因此我们选择如下的“根据”业务建模方案：

- 将挂号单建模为弱实体集，并且原来的主码就诊单号变成挂号单的部分码，标识联系集是**根据**；

- 改进的“病人挂号”业务的建模结果如下图 12 所示:



```

    erDiagram
        住院记录 ||--o{ 药品 : 具有
        住院记录 ||--o{ 用药清单 : 包含
        用药清单 ||--o{ 处方单 : 包含
        处方单 ||--o{ 缴费单 : 生成
        住院档案 ||--o{ 病房 : 安排
        住院档案 ||--o{ 医生 : 建档
        医生 ||--o{ 挂号单 : 根据
        挂号单 ||--o{ 病人 : 挂号
        病人 ||--o{ 医生 : 就诊
        病人 ||--o{ 值班安排 : 排时间
        值班安排 ||--o{ 诊室 : 排诊室
        诊室 ||--o{ 诊疗方案 : 开具
        病人 ||--o{ 诊疗方案 : 开具
        病人 ||--o{ 巡检 : 巡检
        医生 ||--o{ 巡检 : 巡检
        住院记录 ||--o{ 巡检 : 巡检
        住院档案 ||--o{ 巡检 : 巡检
        用药清单 ||--o{ 巡检 : 巡检
        处方单 ||--o{ 巡检 : 巡检
        缴费单 ||--o{ 巡检 : 巡检
        病房 ||--o{ 巡检 : 巡检
        科室 ||--o{ 巡检 : 巡检
        医生 ||--o{ 巡检 : 巡检
        挂号单 ||--o{ 巡检 : 巡检
        病人 ||--o{ 巡检 : 巡检
        值班安排 ||--o{ 巡检 : 巡检
        诊室 ||--o{ 巡检 : 巡检
        诊疗方案 ||--o{ 巡检 : 巡检

        住院记录 {
            string 记录号 PK
            string 病历号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        药品 {
            string 药品名 PK
            string 清单号
            string 诊疗费用
            string 处方单号
            string 流水号
            string 金额
        }
        用药清单 {
            string 清单号 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        处方单 {
            string 处方单号 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        缴费单 {
            string 缴费单号 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        住院档案 {
            string 档案号 PK
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        病房 {
            string 病房名 PK
            string 病床编号
            string 病床状态
        }
        科室 {
            string 科室名 PK
            string 医生职称
        }
        医生 {
            string 医生名 PK
            string 医生职称
        }
        挂号单 {
            string 挂号单号 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        病人 {
            string 病人名 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        值班安排 {
            string 值班安排号 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        诊室 {
            string 诊室名 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        诊疗方案 {
            string 诊疗方案号 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
        巡检 {
            string 巡检号 PK
            string 档案号
            string 入院时间
            string 出院时间
            string 医生
            string 科室
            string 病房
            string 床位
            string 病床状态
        }
    
```

图 13 改进的医院管理信息系统总 E-R 图

3. 数据库逻辑设计

设计出 E-R 图之后，我们将 E-R 图转化为数据库模式，每个实体集都对应一个关系表，而联系集则根据映射基数确定转换方式。图 13 所示的 E-R 图可转为如下数据库关系模式，其中主码属性加粗和下划线、外码属性加粗斜体以示区分。

(1) 医生 Doctor 表：由医生(Doctor)实体集、引用(Citation)联系集和属于(Belong)联系集共同转化而来，如表 17 所示，由于联系集 Citation、Belong 都是一对多联系集，故可合并到 Doctor 表中来。

表 17 医生 Doctor 表

属性名	数据类型	含义
<u>docNo</u>	char(10)	工号
docPassword	char(20)	登录密码
docName	varchar(10)	姓名
sex	char(2)	性别
<i>degree</i>	varchar(20)	职称
telephone	char(11)	电话号码
<i>officeNo</i>	Varchar(3)	科室号

(2) 病人 Patient 表：由病人(Patient)实体集和就诊(Visit)联系集、接诊(Reception)联系集、巡检(Inspect)联系集共同转化而来，如表 18 所示。由于联系集 Visit、Reception、Inspect 是一对多联系集，故可合并到 Patient 表中来。

表 18 病人 Patient 表

属性名	数据类型	含义
patPassword	char(20)	密码
<u>patNo</u>	char(13)	病案号
patName	varchar(10)	姓名
sex	char(2)	性别
telephone	char(11)	电话号码
address	varchar(100)	地址
result	varchar(100)	诊断结果

examTime	date	诊断时间
patCondition	varchar(100)	病人情况
<i>docNo</i>	char(10)	工号
<i>roomNo</i>	varchar(4)	诊室号

挂号单 BookingForm 表:由挂号单(BookingForm)弱实体集和标识联系集

根据(Base)以及联系集(Register)挂号共同转化而来，如

(3) 表 19 所示。由于联系集 Base、Register 都是一对多联系集，故可合并到 BookingForm 表中来。

表 19 挂号单 BookingForm 表

属性名	数据类型	含义
<u>bookingNo</u>	varchar(5)	就诊单号
bookingTime	date	挂号时间
bookingStatu	varchar(10)	挂号状态
<i>patNo</i>	char(13)	病案号
<u>docNo</u>	char(10)	工号

(4) 诊室 Room 表：由诊室(Room)实体集转化而来，如表 20 所示。

表 20 诊室 Room 表

属性名	数据类型	含义
<u>roomNo</u>	varchar(4)	诊室号
roomLocation	varchar(100)	诊室地点
telephone	varchar(11)	诊室电话

值班安排 DutyArrangement 表:由值班安排(DutyArrangement)弱实体集

和标识实体集排诊室(ConsultateRoom)共同转化而来，如

(5) 表 21 所示。由于联系集排诊室 ConsultateRoom 是一对多联系集，故可

合并到 DutyArrangement 表中来。

表 21 值班安排 DutyArrangement 表

属性名	数据类型	含义
<u>dutyDate</u>	date	值班日期
<u>roomNo</u>	varchar(4)	诊室号
dutyShift	ENUM	值班班次

排时间 ScheduleTime 表：由排时间(ScheduleTime)多对多联系集转化而来，如

(6) 表 22 所示：

表 22 排时间 ScheduleTime 表

属性名	数据类型	含义
<u>dutyDate</u>	date	值班日期
<u>docNo</u>	char(10)	工号
<u>roomNo</u>	varchar(4)	诊室号

(7) 科室 Office 表：由科室(Office)实体集转化而来，如表 23 所示。

表 23 科室 Office 表

属性名	数据类型	含义
<u>officeNo</u>	int	科室号
officeName	varchar(20)	科室名称
telephone	varchar(11)	科室电话

(8) 病房 Ward 表：由病房(Ward)实体集和联系集拥有(Possess)共同转化而来，

如表 24 所示。由于联系集 Possess 是一对多联系集，故可合并到 Ward 表中来。

表 24 病房 Ward 表

属性名	数据类型	含义
<u>wardNo</u>	char(4)	病房编号
address	varchar(10)	地点
chargeStand	int	收费标准
officeNo	int	科室号

(9) 病床 Bed 表：由病床(Bed)弱实体集、标识联系集**置放**(Place)以及**安排**(Arrange)联系集共同转化而来,如表 25 所示。由于联系集 Place 是一对多联系集，联系集 Arrange 是一对一联系集，故可合并到 Bed 表中来。

表 25 病床 Bed 表

属性名	数据类型	含义
<u>bedNo</u>	char(3)	病床编号
bedState	char(4)	病床状态
<u>wardNo</u>	char(4)	病房编号
departNo	char(16)	档案号

(10) 住院档案 DepartDocument 表：由**住院档案**(DepartDocument)实体集和**建档**(Document)联系集共同转化而来，如表 26 所示。由于联系集 Document 是一对一联系集，故可合并到 DepartDocument 表中来。

表 26 住院档案 DepartDocument 表

属性名	数据类型	含义
<u>departNo</u>	char(16)	档案号
indepartDate	date	入院时间
outdepartDate	date	出院时间
patNo	char(13)	病案号

(11) 住院记录 DepartRecord 表：由弱实体集**住院记录**(DepartRecord)、标识联系集**记录**(Record)以及联系集**存档**(Save)共同转化而来，如表 27 所示。由于联系集 Record 是一对多联系集，联系集 Save 是一对一个联系集，故可合并到 DepartRecord 表中来。

表 27 住院记录 DepartRecord 表

属性名	数据类型	含义
<u>aidNo</u>	int(3)	记录号
patSymptom	varchar(100)	病人状况
<u>departNo</u>	char(16)	档案号
planNo	char(5)	方案编号

(12) 诊疗方案 Treatment 表：由**治疗方案**(Treatment)实体集以及联系集**开具**(issued)共同转化而来，如表 28 所示。由于联系集 issued 是一对一个联系集，故可合并到 Treatment 表中。

表 28 诊疗方案 Treatment 表

属性名	数据类型	含义
<u>planNo</u>	char(5)	方案编号
description	varchar(100)	方案描述
money	float	收费标准
patNo	char(13)	病案号

(13) 处方单 Recipe 表：由**处方单**(Recipe)实体集和**开具处方**(Prescribe)联系集共同转化而来，如表 29 所示。由于联系集 Prescribe 是一对一个联系集，故可合并到 Recipe 表中。

表 29 处方单 Recipe 表

属性名	数据类型	含义
<u>recipeNo</u>	char(7)	处方单号
symptom	varchar(100)	症状描述
money	numeric	诊疗费用
<i>patNo</i>	char(13)	病案号

(14) 缴费单 Payment 表：由**缴费单**(Payment)实体集和联系集**生成**(Build)共同转化而来转化而来，如表 30 所示。由于联系集 Build 是一对一的联系集，故可合并到 Payment 表中。

表 30 缴费单 Payment 表

属性名	数据类型	含义
<u>paymentNo</u>	char(16)	流水号
fee	float	金额
paymentState	char(6)	缴费状态
paymentDate	date	缴费日期
<i>recipeNo</i>	char(7)	处方单号

(15) 用药清单 MedicineList 表：由**用药清单**(MedicineList)实体集和**包含**(Contain)联系集共同转化而来，如表 31 所示。由于联系集包含 Contain 是一对一的联系集，故可以合并到表中。

表 31 用药清单 MedicineList 表

属性名	数据类型	含义
<u>listNo</u>	char(7)	清单号
medicineTotal	numeric	药品总价
<i>recipeNo</i>	char(7)	处方单号

(16) 具有 Have 表：由**具有**(Have)多对多联系集转化而来，如表 32 所示。

表 32 具有 Have 表

属性名	数据类型	含义
<u>recipeNo</u>	char(7)	清单号
<u>medicineNo</u>	char(10)	药品编号

(17) 药品 Medicine 表：由药品(Medicine)实体集转化而来，如表 33 所示。

表 33 药品 Medicine 表

属性名	数据类型	含义
<u>medicineNo</u>	char(10)	药品编号
medName	varchar(30)	药品名
price	numeric	单价
medLeft	int	库存数量
usage	varchar(100)	用法

(18) 医生职称 Degree 表：由职称(Degree)实体集转化而来。如表 34 所示：

表 34 医生职称 Degree 表

属性名	数据类型	含义
degree	varchar(20)	职称
fee	int	诊疗费用

4. 模式求精

病房 Ward 表的关系模式是：Ward(wardNo,address,chargeStand,*officeNo*)，这个关系模式不符合第二范式(2NF)，因为 chargeStand 属性依赖于 wardNo 和 officeNo 的组合，因此，我们需要将它分成更小的关系模式：

Ward (wardNo, address, *officeNo*) ;
WardCharge (wardNo, *officeNo*, chargeStand) ;

Ward 和 WardCharge 之间存在一对一的关系，因此 WardCharge 表中的 wardNo 和 officeNo 列是对 Ward 表中的 wardNo 和 officeNo 属性的外码。

另外住院记录 DepartRecord 表的关系模式是：

DepartRecord (aidNo, patSymptom, departNo, *planNo*)，也不符合第二范式 (2NF)，因为 patSymptom 属性依赖于 aidNo，而不仅仅依赖于 departNo，我们把它分解为更小的两个关系模式：

DepartRecord (aidNo, departNo, *planNo*)；

RecordSymptom (aidNo, patSymptom)；

DepartRecord 和 RecordSymptom 之间存在一对多的关系，因此 RecordSymptom 表中的 aidNo 列是对 DepartRecord 表中的 aidNo 属性的外码。

因此，根据关系模式求精，我们将病房 Ward 表与住院记录 DepartRecord 表进行分解处理后再建表。

二.系统开展展示部分

1.系统原型设计

1.1 “主页”设计

(1) 界面设计

“主页”界面设计如图 14 所示，页面头部展示本系统的基本信息。左上方展示了医院名称、医院 logo 和网址，中上方是对用户的问候语，右上方展示客服热线和登录和注册按钮。“主页”的主体内容在界面中部，展示了医院咨

询、医院通知和健康提示，右侧图片区可以放置医院的景色图。



图 14 “主页”界面设计

(2) 操作流程

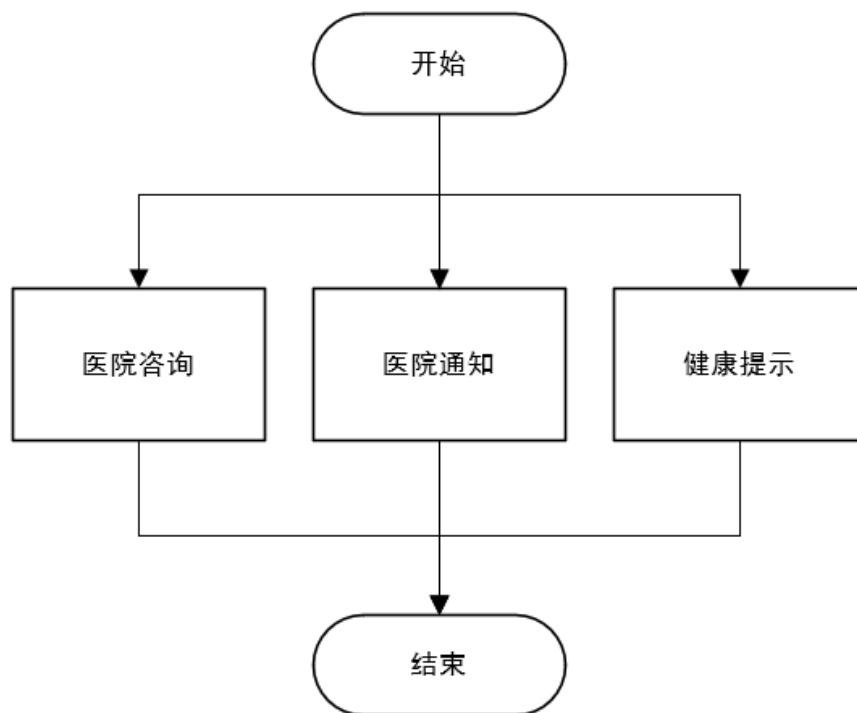


图 15 “主页” 操作流程

1.2 “登录/注册” 设计

(1) 界面设计

当用户点击系统的“登录”按钮后将出现登录界面，点击“注册”按钮后将出现注册界面，注册界面只针对“病人”角色用户。注册时系统会判断用户是否输入要求的内容且输入的电话号码是否已存在，当用户输入完要求填的内容且内容格式正确以及用户名为第一次注册时，则注册成功，系统自动跳转到登录界面。如果用户已有账号，可直接选择自己的角色（医生/病人/管理员），输入用户名与相对应的密码，若用户名和密码一致，则登录成功，跳转“主页”界面。

请登录医院管理信息系统

病案号:

密码:

身份: ☐ 医生 ☒ 病人 ☐ 管理员

图 16 “登录/注册”界面设计_“登录”

注册新用户--填写注册信息, 输入用户名和密码

电话号码:

密码:

确认密码:

图 17 “登录/注册”界面设计_“注册”

(2) 操作流程

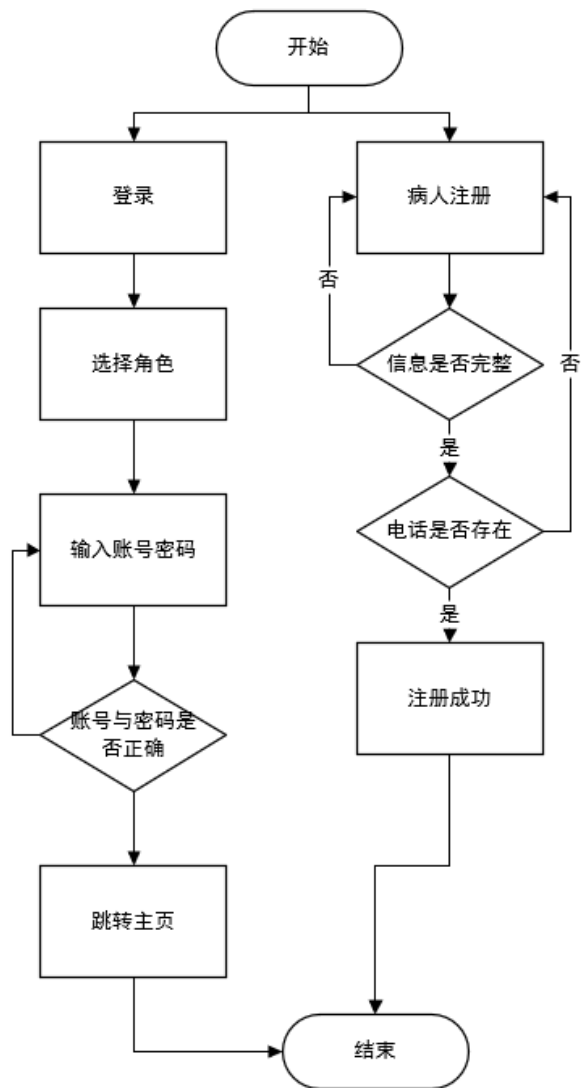


图 18 “登录/注册”操作流程

1.3 “排班”设计

(1) 界面设计

“医生”用户登录后可以进入“排班”界面，如图 19 所示。如果该名医生的职称为“主任医师”，显示排班表单，主任医师可以对本科室的所有医师进行排班，医生姓名栏通过“科室”属性查找所有医生，选择后自动填入医生工号和所属科室，接着设置日期、班次和值班诊室号，点击提交后，系统校验数据是否有空值，校验成功后写入数据库的排班(schedule)表。所有职称的医师都可以在界面下方查看排班日历，自己的班次通过灰色背景高亮显示，信息包括排班日期、

排班诊室号和排班班次。



三幺八医院管理系统
www.hospitalcn.com

D202301001欢迎使用医院管理系统

400-889-8188 客服热线(免长途费)

退出 注册 后台登录

| 主页 | 排班 | 接诊 | 住院 |

医生排班表单

医生姓名: 李梅

医生工号: D202301002

所属科室: 101

日期: yyyy/mm/日

班次: 早班

诊室号:

提交

	早班	中班	晚班
日期	诊室号	诊室号	诊室号
2023-05-24	工号: D202301001 诊室: 1		
2023-05-25	工号: D202301002 诊室: 3		
2023-05-25		工号: D202303001 诊室: 1	
2023-05-26		工号: D202303002 诊室: 3	
2023-05-24		工号: D202305001 诊室: 2	
2023-05-26	工号: D202305002 诊室: 2		

图 19 “排班” 界面设计

(2) 操作流程

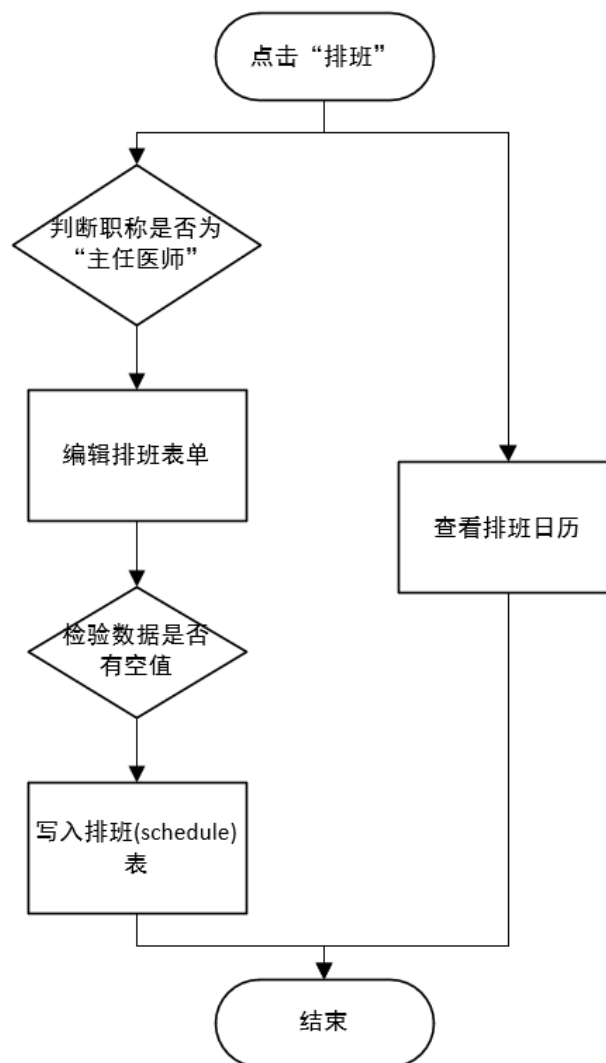


图 20 “排班” 操作流程

1.4 “接诊” 设计


(1) 界面设计

“医生” 用户登录后可以进入“接诊” 界面，医生可以看到所有挂自己号的病人，如图 21 所示。选择病人点击“接诊” 按钮后，进入“开处方” 界面，如图 22 所示。在“开处方” 界面，医生可以在页面最上方看到病人的基本信息，然后在门诊诊断过程中在“症状” 输入框输入病人的症状描述。在页面下方医生可以开具处方单，顶部显示当前的处方单号；在右侧药品列表，医生可以搜索药品名称，通过点击“添加” 按钮选择需要开具的药品；在左侧药品清单栏也可以

点击“添加”按钮快捷增加此类药品数量，点击“减少”减少此类药品数量，点击“移除”按钮移除此类药品。最后点击“保存”按钮，将此处方存入处方(recipe)表，点击“申请住院”后，为病人新建住院档案，点击“结束就诊”返回“接诊”的第一个界面。



图 21 “接诊”界面

姓名:	性别:	电话号码:
杨开放	男	13966491259
病案号:	住址:	
P202205210001	广东省佛山市南海区华南师范大学南海校园	
症状:		
<div>  感冒 </div>		

处方号:	1000004
------	---------

编号	名称	单价	用法	数量	操作
YP202303	维生素C片	1	口服, 每次1片, 每天	1	<div>添加</div> <div>减少</div> <div>移除</div>
YP202305	氯雷他定	8	口服, 每次1片, 每天	1	<div>添加</div> <div>减少</div> <div>移除</div>
YP202306	葡萄糖注射液	5	静脉注射, 每次500	1	<div>添加</div> <div>减少</div> <div>移除</div>

编号	名称	价格	库存	操作
YP202301	阿司匹林	3	200	添加
YP202302	对乙酰氨基酚	3	150	添加
YP202303	维生素C片	1	300	添加
YP202304	罗红霉素	12	50	添加
YP202305	氯雷他定	8	100	添加
YP202306	葡萄糖注射液	5	80	添加
YP202307	盐酸氨溴索胶囊	6	120	添加
YP202308	头孢克洛	15	30	添加
YP202309	氯化钠注射液	2	500	添加
YP202310	头孢呋辛酯片	10	80	添加

保存

申请住院

结束就诊

图 22 “接诊” 界面_ “开处方”

(2) 操作流程

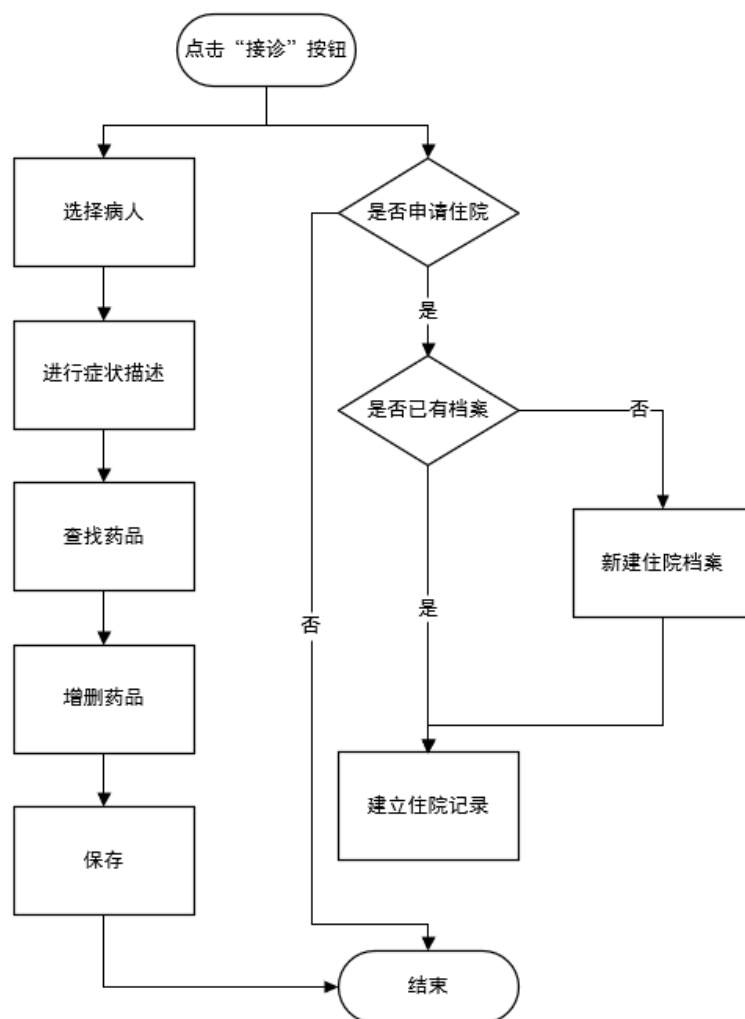


图 23 “接诊” 操作流程

1.5 “住院” 设计

(1) 界面设计

“医生”用户登录后可以进入“住院”界面，医生可以看到住院的所有病人，如图 24 所示。选择病人，点击“档案记录”后查看病人的“住院档案”。页面上方显示病人的基本信息，以及病人的主治医生、入院日期、科室、病房号和病床号。在页面下方，医生可以给当前病人添加住院记录，每条住院记录可以记录当天病人的状况，同时选择病人的治疗方案。



杨开放 男

电话: 13966491259
病案号: P202205210001
入院时间: 2022-05-12

病房号: A101
病床号: 001

档案记录

病人信息



杨开放 男

电话: 13966491259
病案号: P202205210001
入院时间: 2022-05-21

病房号: A101
病床号: 002

档案记录

病人信息



林兰 女

电话: 13570963254
病案号: P202205210002
入院时间: 2022-05-13

病房号: A101
病床号: 003

档案记录

病人信息



黄艾琳 女

电话: 15827048963
病案号: P202205210004
入院时间: 2022-05-14

病房号: B108
病床号: 001

档案记录

病人信息

图 24 “住院” 界面设计_1

姓名:	性别:	电话号码:
<input type="text" value="杨开放"/>	<input type="text" value="男"/>	<input type="text" value="13966491259"/>
病案号:	住址:	
<input type="text" value="P202205210001"/>	<input type="text" value="广东省佛山市南海区华南师范大学南海校园"/>	
主治医生:	入院日期:	科室:
<input type="text" value="李四"/>	<input type="text" value="2022-05-21"/>	<input type="text" value="101"/>
病房:	病床:	
<input type="text" value="A101"/>	<input type="text" value="002"/>	

1

病人状况:

长时间高烧42度

治疗方案:

20001

2

病人状况:

病人体温正常

治疗方案:

10001

3

病人状况:

今日状况正常

治疗方案:

10001

保存

添加记录

出院

图 25“住院”界面设计_2

(2) 操作流程



图 26 “住院” 操作流程

1.6 “挂号” 设计

(1) 界面设计

“病人” 用户登录后可以进入“挂号” 界面，界面上方显示病人基本信息，下方可以进行挂号操作，如图 27 所示。病人首先选择“挂号时间”，然后选择“科室”，确定科室后查询数据库医生(doctor)表中该科室的所有医生，显示在

“接诊医生”栏。“挂号单号”栏显示当前挂号单的单号。点击“保存”后，信息经过校验存入数据库的挂号(bookingform)表中。



三幺八医院管理系统
www.hospitalcn.com

P202205210001欢迎使用医院管理系统

400-889-8188 客服热线 (免长途费)

[退出](#) [注册](#) [后台登录](#)

[|](#) [主页](#) [|](#) [挂号](#) [|](#) [就诊](#) [|](#) [缴费](#) [|](#)

患者姓名:	病案号:	性别:
<input type="text" value="杨开放"/>	<input type="text" value="P202205210001"/>	<input type="text" value="男"/>
手机号码:	住址:	
<input type="text" value="13966491259"/>	<input type="text" value="广东省佛山市南海区华南师范大学南海校园"/>	

挂号日期:	科室:	接诊医生:
<input type="text" value="2023/05/26"/> 	<input type="text" value="神经科"/>	<div><input type="text" value="李四"/><div><div>李四</div><div>王五</div><div>李梅</div></div></div>
挂号单号:		
<input type="text" value="10000009"/>		

挂号

取消

图 27 “挂号” 界面设计

(2) 操作流程

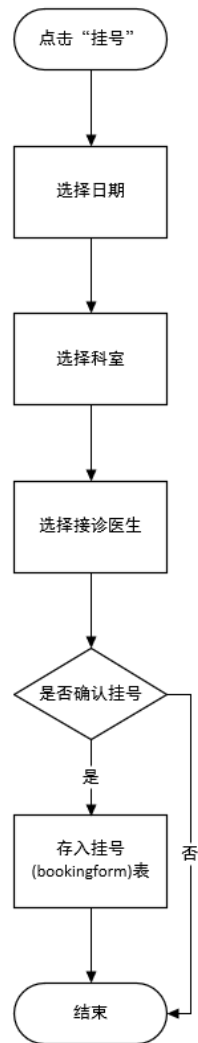


图 28 “挂号” 操作流程

1.7 “就诊” 设计

(1) 界面设计

“病人” 用户登录后可以进入“就诊” 界面，界面左上方有两个按钮，可以分别查询自己的“用药清单” 和“就诊记录”。用药清单标记了所属“处方单号”，罗列了清单的所有药品，点击某一条清单后卡片展开，显示清单详情，详细描述了“药品名称”、“药品数量” 和“用法”。“就诊记录” 则记录了病人的每一次就诊的病症描述和诊断结果，点击后显示就诊详情，包括就诊日期，接诊医生的名称，就诊诊室和诊疗费用。

用药清单

就诊记录

处方药品

输入处方单号

搜索

处方单号: 1000001

药品: 头孢克洛

药品: 头孢呋辛酯片

处方单详情1

头孢克洛 数量: 2 用法: 静脉注射, 每次1克,

头孢呋辛酯片 数量: 2 用法: 口服, 每次1片, 每天

图 29 “就诊” 界面_“用药清单”

用药清单

就诊记录

就诊记录

就诊记录编号: 1000001

病症描述: 过敏性鼻炎

诊断结果: 过敏性鼻炎

就诊记录详情

就诊日期: 2023-05-24

医生姓名: 李四

就诊科室: 101

诊疗费用: 50.00

图 30 “就诊” 界面_“就诊记录”

(2) 操作流程

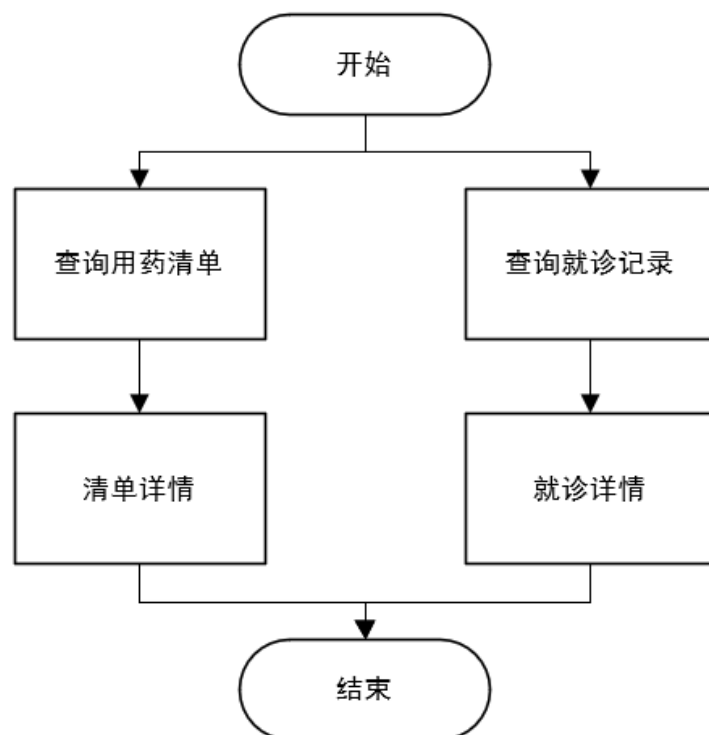


图 31 “就诊” 操作流程

1.8 “缴费” 设计

(1) 界面设计

“病人” 用户登录后可以进入“缴费” 界面，查看自己所有缴费记录，如图 32 所示。每一条缴费记录都有一个“缴费单号”，同时显示“缴费类型” 和“缴费时间”。缴费类型有两种，标识“门诊部缴费” 或“住院部缴费”。点击缴费单，显示“费用明细”，罗列缴费项目和对应的费用。每一条缴费单还会在右上角标记缴费状态和缴费总金额，缴费状态有三种：“已缴费”、“待缴费” 和“逾期未缴费”。

缴费单号: PM20220512000001	已缴费 - ¥200
缴费类型: 门诊部缴费 缴费时间: 2022-05-12	
费用明细 项目1: 诊疗费用 ¥50 项目2: 药品费用 ¥150	
缴费单号: PM20220513000001	已缴费 - ¥300
缴费类型: 门诊部缴费 缴费时间: 2022-05-13	
缴费单号: PM20220514000001	已缴费 - ¥450
缴费类型: 门诊部缴费 缴费时间: 2022-05-14	
缴费单号: PM20220515000001	已缴费 - ¥150
缴费类型: 门诊部缴费 缴费时间: 2022-05-15	
缴费单号: PAY002	待缴费 - ¥200
缴费类型: 门诊部缴费 缴费时间:	
缴费单号: PAY003	逾期未缴费 - ¥100
缴费类型: 门诊部缴费 缴费时间:	

图 32 “缴费” 界面设计

(2) 操作流程

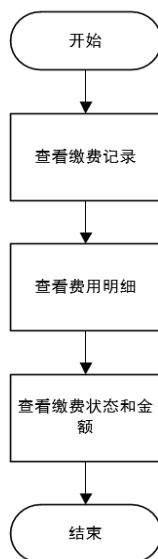


图 33 “缴费” 操作流程图

1.9 “管理” 设计

(1) 界面设计

管理员登录系统之后可以修改医院信息、医生信息、科室信息和药品信息，如。医生信息通过提交表单更改医院 logo、医院名称和医院网址。医生信息通过医生工号、密码等信息更改，如果表单中存在医生信息，则修改，若不存在则创建一条新的医生数据。科室信息可修改的包括科室名称、负责医师和科室位置。药品信息通过表单直接修改药品编号、药品名称、药品信息和用法以及库存信息。



三幺八医院管理系统
www.hospitalcn.com

M202301001欢迎使用医院管理系统

400-889-8188 客服热线 (免长途费)

退出 注册 后台登录

[主页](#) | [医院信息](#) | [医生信息](#) | [科室信息](#) | [药品信息](#) | [信息总览](#)

医院信息

医院Logo:

医院名称:

医院网址:

保存

图 34 “管理” 界面_医院信息



三幺八医院管理系统
www.hospitalcn.com

M202301001欢迎使用医院管理系统

400-889-8188 客服热线 (免长途费)

退出 注册 后台登录

[主页](#) | [医院信息](#) | [医生信息](#) | [科室信息](#) | [药品信息](#) | [信息总览](#)

医生信息

医生工号:

密码:

医生姓名:

性别:

职称:

电话号码:

所属科室号:

保存

图 35 “管理” 界面_医生信息

科室信息

科室号:
科室名称:
负责医师:
科室位置:

保存

图 36 “管理”界面_科室信息



图 37 “管理”界面_药品信息

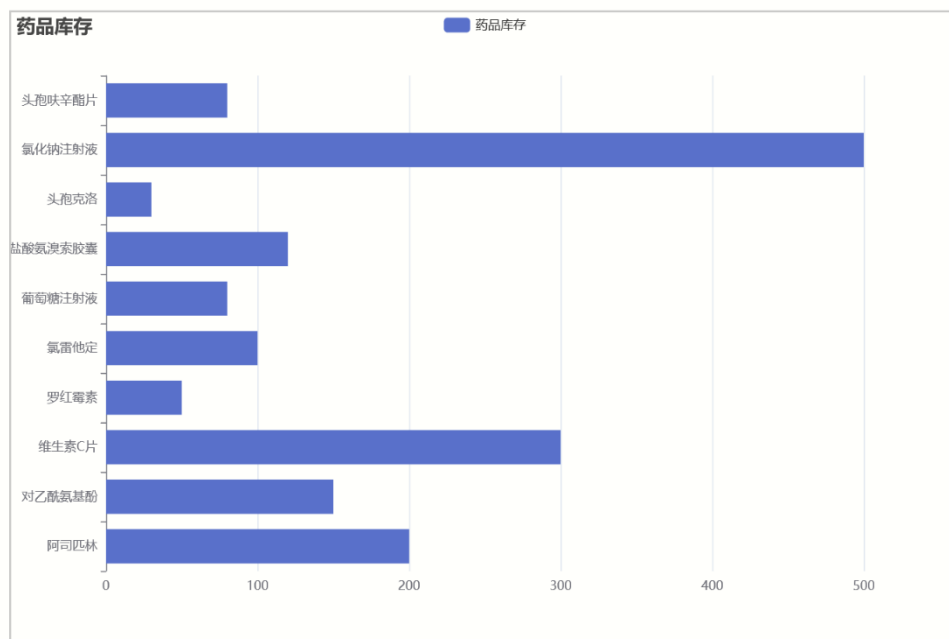


图 38 “管理”界面_药品库存展示

(2) 操作流程

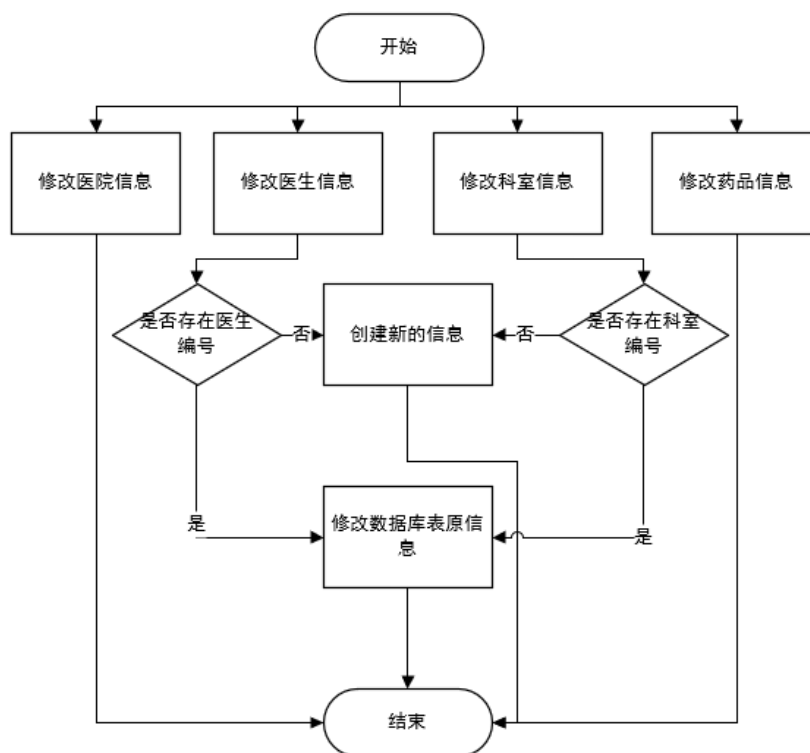


图 39 “管理” 操作流程

2 附件

2.1 源码见文件夹 DbFinal2023

2.2 源码仓库链接 <https://github.com/JJuprising/DbFinal2023.git>

2.3 所用到的技术栈

本套医院后台管理系统主要运用了以下技术栈：

前后端分离开发：本系统采用了前后端分离的开发模式，前端使用 HTML、CSS 和 JavaScript 构建，后端使用 PHP 语言进行开发。

Git 版本控制：Git 是一种分布式版本控制系统，用于管理代码库的版本。本系统使用 Git 进行版本控制，以便多人协同开发和管理代码的变更。

ThinkPHP 框架：ThinkPHP 是一个基于 PHP 语言的 Web 应用程序开发框架，它提供了一系列的开发工具和组件，可以帮助开发者快速构建高效的 Web 应用程序。本系统使用 ThinkPHP 框架作为后端开发的基础框架。

MySQL5.7 数据库：MySQL 是一种开源的关系型数据库管理系统，它支持多种操作系统，包括 Windows、Linux 和 Mac OS X 等。本系统使用 MySQL5.7 作为数据库，以存储和管理数据。MySQL5.7 提供了更高的性能和更好的安全性，可以满足本系统的数据存储和管理需求。

ZendStudio：Zend Studio 是一款基于 Eclipse 的 PHP 集成开发环境（IDE），由 Zend Technologies 开发和维护。它提供了一系列的开发工具和功能，可以帮助开发者更加高效地开发和调试 PHP 应用程序。

Ajax 技术：Ajax 是一种在 Web 应用程序中使用的 Web 开发技术，可以在不重新加载整个页面的情况下更新部分页面内容。本系统使用 Ajax 技术实现了无

刷新的数据交互，提高了用户体验。

jQuery 框架：jQuery 是一个快速、简洁的 JavaScript 框架，可用于 HTML 文档的遍历和操作、事件处理、动画效果和 AJAX 交互等。本系统使用 jQuery 框架作为前端开发的基础框架，使得前端开发更加简单、高效。

2.4 数据库文件 hospital.txt

```
Drop database hospital;
Create database hospital;
Use hospital;
drop table if exists doctor;
Create table doctor(
    docNo char(10) NOT NULL,
    docPassword char(80),
    docName varchar(10),
    sex char(2),
    degree varchar(20),
    telephone char(13),
    officeNo varchar(3),
    Primary Key(docNo)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
set names utf8;
Delete from doctor;

insert into doctor values("D202301001", md5('123456'),'李四','男','主任医师',
'13765876477','101');
insert into doctor values("D202305001", md5('123456'),'王五','男','副主任医师',
'13458493871','101');
insert into doctor values("D202301002", md5('123456'),'李梅','女','主治医师',
'13675849582','101');
insert into doctor values("D202303001", md5('123456'),'王芳','女','主治医师',
'16685768948','102');
insert into doctor values("D202305002", md5('123456'),'刘强','男','副主任医师',
'13458295069','102');
insert into doctor values("D202303002", md5('123456'),'赵娟','女','主任医师',
'13908574832','102');
```

```
select * from doctor;
```

```
drop table if exists charge;
```

```
Create table charge(  
    degree varchar(20),  
    price numeric NOT NULL,  
    Primary Key(degree)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;  
set names utf8;  
Delete from charge;
```

```
insert into charge values("主任医师",20.0);
```

```
insert into charge values("副主任医师",15.0);
```

```
insert into charge values("主治医师",10.0);
```

```
select * from charge;
```

```
drop table if exists medicine;
```

```
Create table medicine(  
    medicineNo char(10) NOT NULL,  
    medName varchar(30),  
    price numeric,  
    medLeft INT,  
    medUsage varchar(10),  
    Primary Key(medicineNo)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;  
set names utf8;  
Delete from medicine;
```

```
insert into medicine values('YP202301','阿司匹林',2.5,200,'口服，每次 1 片，每天  
3 次');
```

```
insert into medicine values('YP202302','对乙酰氨基酚',3.0,150,'口服，每次 1 片，  
每天 2 次');
```

```
insert into medicine values('YP202303','维生素 C 片',1.0,300,'口服，每次 1 片，  
每天 1 次');
```

```
insert into medicine values('YP202304','罗红霉素',12.0,50,'口服，每次 2 片，每天
```

3 次');

insert into medicine values('YP202305','氯雷他定',8.0,100,'口服，每次 1 片，每天 2 次');

insert into medicine values('YP202306','葡萄糖注射液',5.0,80,'静脉注射，每次 500 毫升，每天 1 次');

insert into medicine values('YP202307','盐酸氨溴索胶囊',6.0,120,'口服，每次 1-2 粒，每天 3 次');

insert into medicine values('YP202308','头孢克洛',15.0,30,'静脉注射，每次 1 克，每天 2 次');

insert into medicine values('YP202309','氯化钠注射液',2.0,500,'静脉注射，每次 500 毫升，根据需要使用');

insert into medicine values('YP202310','头孢呋辛酯片',10.0,80,'口服，每次 1 片，每天 2 次');

select * from medicine;

drop table if exists medicationlist;
Create table medicationlist(
 recipeNo char(7) NOT NULL,
 medicineNo char(10) NOT NULL,
 medQuantity int,
 Primary Key(recipeNo,medicineNo)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
set names utf8;
Delete from medicationlist;
insert into medicationlist values
('1000001','YP202308',2),
('1000001','YP202310',1);
('1000001','YP202310','1'),
('1000001','YP202308','2'),

```

('1000004', 'YP202302','1'),
('1000004', 'YP202303','1'),
('1000004', 'YP202304','1'),
('1000005', 'YP202303','1'),
('1000005', 'YP202305','1'),
('1000005', 'YP202307','1'),
('1000006', 'YP202306','1'),
('1000006', 'YP202307','1'),
('1000006', 'YP202308','1'),
('1000007', 'YP202301','1'),
('1000007', 'YP202302','2'),
('1000007', 'YP202308','2'),
('1000007', 'YP202307','1');
select * from medicationlist;

```

```

drop table if exists patient;
CREATE TABLE patient(
    patNo char(13) PRIMARY KEY,
    patPassword char(80) NOT NULL DEFAULT '123456',
    patName varchar(10) NOT NULL,
    sex char(2) CHECK(sex IN ('男','女')),
    telephone char(11),
    address varchar(100) NOT NULL
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
set names utf8;
Delete from patient;

insert into patient values('P202205210001',md5('123456'),'杨开放','男',
'13966491259','广东省佛山市南海区华南师范大学南海校园');

insert into patient values('P202205210002',md5('123456'),'林兰','女','13570963254','
广东省佛山市顺德区大良街道顺德大学');

insert into patient values('P202205210003',md5('123456'),'陈明','男','13876025692','
广东省佛山市高明区东平街道高明学院');

insert into patient values('P202205210004',md5('123456'),'黄艾琳','女',
'15827048963','广东省佛山市南海区桂城街道南海医学院');

```



```
insert into patient values('P202205210005',md5('123456'),'张三','男','13123869574','
```

```
广东省佛山市三水区大塘街道华南理工大学三水校区');
```

```
insert into patient values('P202205210006',md5('123456'),'李四','女','13726954123','
```

```
广东省佛山市南海区西樵街道西樵中学');
```

```
insert into patient values('P202205210007',md5('123456'),'王五','男','15674839012','
```

```
广东省佛山市顺德区培正街道培正高级中学');
```

```
insert into patient values('P202205210008',md5('123456'),'赵六','女','13589632547','
```

```
广东省佛山市高明区黄阁街道广东高明黄阁中学');
```

```
insert into patient values('P202205210009',md5('123456'),'孙七','男','13986274591','
```

```
广东省佛山市三水区梅林街道广东三水梅林街小学');
```

```
select * from patient;
```

```
drop table if exists departDocument;
```

```
CREATE TABLE departDocument(  
    departNo char(8) PRIMARY KEY,  
    patNo char(16) NOT NULL ,  
    indepartDate date NOT NULL,  
    outdepartDate date,  
    wardNo char(4) NOT NULL,  
    bedNo char(3) NOT NULL
```

```
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
set names utf8;
```

```
Delete from departDocument;
```

```
insert into departDocument values('10000001','P202205210001','2022-05-12','2022-05-15','A101','001');
```

```
insert into departDocument values('10000002','P202205210001','2022-05-21',NULL,'A101','002');
```

```
insert into departDocument values('10000003','P202205210002','2022-05-13','2022-05-14','A101','003');
```

```
insert into departDocument values('10000004','P202205210004','2022-05-14','2022-05-16','B108','001');
```

```
select * from departDocument;
```

```
drop table if exists departRecord;
```

```
CREATE TABLE departRecord(  
    departNo char(8) ,  
    aidNo int(3) NOT NULL ,  
    aidPlan char(5) ,  
    patSymptom varchar(100)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;  
set names utf8;
```

```
Delete from departRecord;
```

```
insert into departRecord values('10000001',001,'20002','长时间昏迷');
```

```
insert into departRecord values('10000001',002,'20001','意识微弱');
```

```
insert into departRecord values('10000001',003,'20001','意识微弱，有好转');
```

```
insert into departRecord values('10000001',004,'10001','病人意识正常');
```

```
insert into departRecord values('10000002',001,'20001','长时间高烧 42 度');
```

```
insert into departRecord values('10000002',002,'10001','病人体温正常');
```

```
insert into departRecord values('10000003',001,'10001','病人体温正常');
```

```
insert into departRecord values('10000003',002,'10001','病人长时间胃胀气');
```

```
insert into departRecord values('10000003',003,'10001','病人症状缓解');
```

```
select * from departRecord;
```

```
drop table if exists payment;
```

```
CREATE TABLE payment(  
    paymentNo char(16) PRIMARY KEY,  
    patNo char(13) NOT NULL,  
    fee float NOT NULL ,  
  
    paymentState char(6) CHECK(paymentState IN ('已缴费','未缴费')),  
  
    paymentDate date  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;  
set names utf8;
```

```
Delete from payment;

insert into payment values('PM20220512000001','P202205210001',200.00,'已缴费',
'2022-05-12');

insert into payment values('PM20220513000001','P202205210001',300.00,'已缴费',
'2022-05-13');

insert into payment values('PM20220514000001','P202205210001',450.00,'已缴费',
'2022-05-14');

insert into payment values('PM20220515000001','P202205210001',150.00,'已缴费',
'2022-05-15');

insert into payment values('PM20220513000002','P202205210002',350.00,'已缴费',
'2022-05-13');

insert into payment values('PM20220514000002','P202205210002',100.00,'已缴费',
'2022-05-14');

insert into payment values('PM20220514000003','P202205210004',500.00,'已缴费',
'2022-05-14');

insert into payment values('PM20220515000002','P202205210004',600.00,'已缴费',
'2022-05-15');

insert into payment values('PM20220516000001','P202205210004',550.00,'未缴费',
,NULL);
```

```
select * from payment;
```

```
drop table if exists ward;
CREATE TABLE ward(
    wardNo char(4) PRIMARY KEY,
    address varchar(10) NOT NULL,
    chargeStand INT NOT NULL,
    officeNo varchar(3) NOT NULL
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
set names utf8;
Delete from ward;

insert into ward values("A101", "一楼 A 区", 110, "101");

insert into ward values("A204", "二楼 A 区", 120, "101");
```

```
insert into ward values("B305", "三楼 B 区", 100, "102");
```

```
insert into ward values("B108", "一楼 B 区", 90, "101");
```

```
insert into ward values("C409", "四楼 C 区", 100, "102");
```

```
insert into ward values("C312", "三楼 C 区", 110, "102");
```

```
select * from ward;
```

```
drop table if exists bed;
```

```
CREATE TABLE bed(
```

```
    bedNo char(3) NOT NULL,
```

```
    wardNo char(4) NOT NULL,
```

```
    bedStand char(4) CHECK(bedStand IN ('有人','空闲'))
```

```
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
set names utf8;
```

```
Delete from bed;
```

```
insert into bed values("001", "A101", "有人");
```

```
insert into bed values("002", "A101", "有人");
```

```
insert into bed values("003", "A101", "空闲");
```

```
insert into bed values("001", "B108", "有人");
```

```
insert into bed values("003", "C409", "空闲");
```

```
insert into bed values("001", "C312", "空闲");
```

```
select * from bed;
```

```
drop table if exists schedule;
```

```
CREATE TABLE schedule (
```

```
    docNo char(10),
```

```
    officeNo varchar(3),
```

```
    dutyDate DATE,
```

```
    dutyShift ENUM('morning', 'afternoon', 'evening'),
```

```
    RoomNo varchar(4),
```

```
    PRIMARY KEY (docNo, dutyDate),
```

```
    FOREIGN KEY (docNo) REFERENCES Doctor (docNo)
```

```
);
INSERT INTO schedule (docNo, officeNo, dutyDate, dutyShift, RoomNo)
VALUES
('D202301001', 101, '2023-05-24', 'morning', 1),
('D202305001', 101, '2023-05-24', 'afternoon', 2),
('D202301002', 101, '2023-05-25', 'morning', 3),
('D202303001', 102, '2023-05-25', 'afternoon', 1),
('D202305002', 102, '2023-05-26', 'morning', 2),
('D202303002', 102, '2023-05-26', 'afternoon', 3);
```

```
select * from schedule;
```

```
drop table if exists bookingform;
CREATE TABLE bookingform (
    bookingNo INT NOT NULL auto_increment,
    bookingTime DATETIME,
    docNo CHAR(10),
    patNo CHAR(13),
    PRIMARY KEY(bookingNo)
);
```

```
INSERT INTO bookingform (bookingNo, bookingTime, docNo, patNo)
VALUES
(10000001, '2023-05-12 09:00:00', 'D202301001', 'P202205210001'),
(10000002, '2023-05-13 10:30:00', 'D202301001', 'P202205210002'),
(10000003, '2023-05-13 14:00:00', 'D202301001', 'P202205210003'),
(10000004, '2023-05-14 16:30:00', 'D202301001', 'P202205210004'),
(10000005, '2023-05-15 11:00:00', 'D202301001', 'P202205210005'),
(10000006, '2023-05-16 12:00:00', 'D202301001', 'P202205210006'),
(10000007, '2023-05-16 15:30:00', 'D202301001', 'P202205210007'),
(10000008, '2023-05-17 16:00:00', 'D202301002', 'P202205210008');
```

```
select * from bookingform;
```

```
drop table if exists office;
CREATE TABLE office (
    officeNo INT NOT NULL,
    officeName VARCHAR(50) NOT NULL,
    telephone VARCHAR(20),
    chiefDoc VARCHAR(50),
    PRIMARY KEY (officeNo)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
set names utf8;
Delete from office;
```

```
INSERT INTO office(officeNo, officeName, telephone, chiefDoc)
VALUES
```

```
(101, '耳鼻喉科', '555-1234', '李四'),
```

```
(102, '骨科', '555-5678', '赵娟');
```

```
select * from office;
```

```
drop table if exists recipe;
```

```
CREATE TABLE recipe (
    recipeNo char(7) PRIMARY KEY,
    symptom VARCHAR(100),
    money DECIMAL(10, 2),
    docNo CHAR(10),
    patNo CHAR(13)
);
```

```
INSERT INTO recipe (recipeNo, symptom, money, docNo, patNo)
VALUES
```

```
('1000001', '过敏性鼻炎', 50.00, 'D202301001', 'P202205210001'),
```

```
('1000003', '上呼吸道感染', 38.00, 'D202301001', 'P202205210002'),
```

```
('1000002', '急性肠胃炎', 72.00, 'D202301001', 'P202205210003');
```

```
select * from recipe;
```

```
drop table if exists aidplan;
```

```
CREATE TABLE aidplan (
    planNo char(5) PRIMARY KEY,
    description VARCHAR(100),
    money float
);
```

```
INSERT INTO aidplan (planNo, description, money)
VALUES
```

```
('10001', '标准护理，附生理指标监测', 50.00),
```

```
('20001', '特级护理，附生理指标监测', 100.00),
```

('20002','特准护理, 附生理指标监测、专人护理',200.00);

select * from aidplan;

drop table if exists manage;

Create table manage(

manNo char(10) NOT NULL,

manPassword char(80),

manName varchar(10),

sex char(2),

telephone char(13),

Primary Key(manNo)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

set names utf8;

Delete from manage;

insert into manage values("M202301001", md5('123456'),'赵明','男','13765876477');